Universidade do Minho
Escola de Engenharia

Paulo Sérgio Cunha Vicente

A model for the development of neural representations: from human observation to robotics

julho de 2016

UMinho | 2016

Paulo Sérgio Cunha Vicente

A model for the development of neural representations:
from human observation to robotics

Universidade do Minho
Escola de Engenharia

Paulo Sérgio Cunha Vicente

A model for the development of neural representations: from human observation to robotics

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação da
Professora Doutora Estela Guerreiro G. S. Bicho Erlhagen

julho de 2016

Acknowledgements

This Thesis would not have been possible without all the support and guidance of those who helped me during these years and who I would like to thank.

First, I would like to thank my scientific adviser, Professor Doctor Estela Bicho, for providing me with the opportunity to endeavour into this project along with all her guidance, support and expertise that were so fundamental to the success of this work.

Second, my deepest gratitude to Doctor Emanuel Sousa for taking the time to help me on this endeavour with his knowledge and guidance throughout this thesis. Helping me to focus my attention where it was most needed and providing me with his insight on the subject matter.

I would also like to thank Professor Doctor Wolfram Erlhagen for providing me with a different and critical perspective on the subject.

I wish to express my gratitude to Luís Louro and Tiago Malheiro, for helping me with their expertise and giving me access to their resources.

A special thank you to my friends, Ricardo Gomes, Ângela Loureiro and Darya Yelshyna, along with my friends and colleagues at the MARLab, at the University of Minho, namely, (alphabetically) Carlos Faria, Flora Ferreira, Gianpaolo Gulleta, Rui Silva, Toni Machado and Weronika Wojtak. Their continuous support was a great help, constantly urging me on to complete this endeavour.

Finally, I would like to thank my parents, Guilherme and Dalila, and the rest of my family for their continuous support and guidance and without whom this work would

not have been possible.

This work was partially supported by the project FP7-NETT-Neural Engineering Transformative Technologies (ref. PITN-GA-2011-289146).

Abstract

A model for the development of neural representations: from human observation to robotics

As the world population continues to rapidly age comes the need for more and improved supporting infrastructures, long term health care services and assistive devices. As a way of dealing with the growing diversity of tasks, as well as increase their acceptability by different users, these devises must be able to adapt to new situations and requirements. Such is the role of a robotic system capable of understanding and learning from the user and caregivers by means of human observation.

To that end, the work presented in this Master thesis addressed a part of the issue of Learning from Observation (LfO). The part in question is the generation of representations that a robotic system can understand for the observed actions and/or objects. This work demonstrates how a system that combines Dynamic Neural Fields (DNFs) and Self-Organizing Maps (SOMs) is capable of creating meaningful representations for actions performed by a human. In particular, this work altered a previously existing model for combining a DNF and a SOM so as to improve its stability and provide greater control over the system. It was then shown that the new model is capable of creating meaningful representations for human arm motions, using both simulated and unfiltered acquired data (demonstrating the intrinsic capabilities of the system for filtering the data).

This page has been intentionally left blank.

Resumo

Modelo para o desenvolvimento de representações neuronais: da demonstração humana à robótica

Com o rápido crescimento da população idosa a nível mundial, acentua-se a necessidade de À medida que a população mundial continua rapidamente a envelhecer vem a necessidade de mais e melhores infraestruturas de apoio, serviços de cuidado a longo prazo e dispositivos de assistência médica. Para que estes dispositivos possam lidar com a crescente diversidade de tarefas, bem como aumentar a sua aceitação por parte dos utilizadores, eles devem ser capazes de se adaptar a novas situações e necessidades do utilizador. Esta é a função de um sistema robótico capaz de compreender e aprender com o utilizador e os assistentes médicos através da observação de um humano.

Assim, o trabalho apresentado nesta dissertação aborda uma parte do problema da aprendizagem por observação. A parte em questão refere-se à geração de representações para as ações e/ou objetos observados passiveis de serem compreendidas por um sistema robótico. Este trabalho pretende demonstrar como um sistema que combina Campos Dinâmicos Neuronais (*DNFs*) e *Self-Organizing Maps* (*SOMs*) é capaz de criar representações significativas para as ações realizadas por um ser humano. Em particular, neste trabalho foi alterado um modelo anteriormente desenvolvido que combina um *DNF* e um *SOM*, de modo a melhorar a sua estabilidade e aumentar o controlo sobre o sistema. Pretende-se também demonstrar que o novo modelo é capaz de criar representações significativas para os movimentos dum braço humano, utilizando tanto dados

simulados como adquiridos, sem necessidade de filtragem (demonstrando as capacidades intrínsecas do sistema para a filtragem dos dados).

Contents

This page has been intentionally left blank.

List of Abbreviations

ANN Artificial Neural Network.

ARoS Anthropomorphic Robotic System.

DNF Dynamic Neural Field.

DNFL Dynamic Neural Field Layer.

DoF Degree of Freedom.

GNAG Grasp Nut with Above Grip.

GNSG Grasp Nut with Side Grip.

GWAG Grasp Wheel with Above Grip.

HRI Human-Robot Interaction.

IQR Interquartile range.

MARLab Mobile and Anthropomorphic Robotics Laboratory.

NN Neural Network.

OL Observational Learning.

RL Reinforcement Learning.

SDK Software Development Kit.

SL Supervised Learning.

SODNF Self-Organizing Dynamic Neural Field.

SOL Self-Organizing Layer.

SOM Self-Organizing Map.

std standard deviation.

UL Unsupervised Learning.

Nomenclature

Numbers in bold indicate the equation number.

Analysis parameters

$\mathrm{eval\,window}$ number of previous centroid positions over which to evaluate if $Q_{\mathrm{dist}}(i,p) > \eta(a,b,p)$ for any two populations. p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

$\mathrm{th}_{\mathrm{ignored}}$ fraction of the total iterations during which a population must have a dispersion inferior to the minimum dispersion at that time to not be ignored. p.26, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

p probability of the centroid being in the area of dispersion considered. p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

Q_{window} number of previous iterations of a population used for calculating its current dispersion area (Q_{dist}). p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

$\mathrm{success\,rate}$ fraction of the iterations that must have produced a stable centroid in the DNFL. p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.79, p.81–p.86, p.98–p.102, p.104–p.123

$\mathrm{th}_{\mathrm{disp}}$ fraction of the number of populations that should not be ignored, relative to the total number of populations. p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

$\mathrm{th}_{\mathrm{eval\,window}}$ fraction of the previous $\mathrm{eval\,window}$ iterations during which two populations must have a distance above $\eta(a,b,p)$ to be considered as separated. p.25, p.57, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123

$\arctan2(y, x)$ four quadrant inverse tangent. p.41, p.46, **4.28**, **4.37**, **4.43**, **4.52**, **4.68**, **4.69**, **4.74**, **4.76**, **4.77**, **4.79–4.81**, **4.94**, **4.95**, **4.98**, **4.110**, **4.113**

Arm and hand joint angles

θ_{elbow} elbow rotation. p.37

$\theta_{\text{shoulder}\,x}$ shoulder joint yaw. p.37

$\theta_{\text{shoulder}\,y}$ shoulder joint pitch. p.37

$\theta_{\text{shoulder}\,z}$ shoulder joint roll. p.37

$\theta_{\text{wrist}\,x}$ wrist yaw. p.37

$\theta_{\text{wrist}\,y}$ wrist pitch. p.37

$\theta_{\text{wrist}\,z}$ wrist roll. p.37

Arm and hand joint positions

${}^{A}P_{\text{joint}} = (x, y, z)^T$ position of joint on the axes of joint A. p.37

${}^{A}P_{\text{shoulder}} = (x, y, z)^T$ position of the shoulder joint on the axes of joint A. **4.88**

Arm and hand link vectors

${}^{A}\vec{v}_{\text{link}} = {}^{A}\overrightarrow{P_{\text{prev}}P_{\text{joint}}}$ vector of the link named link that ends in joint and starts in joint prev, the previous joint in the hierarchy relative to joint, on the axes of joint A. p.37, p.38, p.43, p.44, p.48, **4.17–4.21**, **4.24–4.28**, **4.35**, **4.37–4.44**, **4.48**, **4.60**, **4.68**, **4.69**, **4.74–4.76**, **4.79–4.81**

${}^{A}\vec{v}_{\text{fore-arm}} = {}^{A}\overrightarrow{P_{\text{elbow}}P_{\text{wrist}}}$ vector of the fore-arm on the axes of joint A. p.55, **4.98**, **4.99**, **4.103**, **4.104**, **4.110**, **4.111**

${}^{A}\vec{v}_{\text{hand}} = {}^{A}\overrightarrow{P_{\text{wrist}}P_{\text{hand}}}$ vector of the hand on the axes of joint A. **4.106**

${}^{A}\vec{v}_{\text{index}} = {}^{A}\overrightarrow{P_{\text{hand}}P_{\text{index}}}$ vector of the index finger on the axes of joint A. **4.106**

${}^{A}\vec{v}_{\text{shoulder}} = {}^{A}\overrightarrow{P_{\text{neck base}}P_{\text{shoulder}}}$ vector of the shoulder on the axes of joint A. p.51, **4.85**, **4.86**, **4.90**

${}^{A}\vec{v}_{\text{upper arm}} = {}^{A}\overrightarrow{P_{\text{shoulder}}P_{\text{elbow}}}$ vector of the upper arm on the axes of joint A. p.52, **4.94**, **4.95**

${}^{A}\vec{v}_{\text{upper spine}} = {}^{A}\overrightarrow{P_{\text{middle spine}}P_{\text{neck base}}}$ vector of the upper spine on the axes of joint A. p.51, **4.90**

$\text{cdist}_{\text{points}}(x, y)$ circular distance between two points. p.23, **4.7**

Movement planing

$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{11})$ joint expense factors. p.28, p.29, **4.14**, **4.15**

$\boldsymbol{\theta}_0 = (\theta_{01}, \theta_{02}, \ldots, \theta_{011})$ initial posture. p.28, **4.14**, **4.15**

Θ_b set of possible bounce postures. **4.15**

$\boldsymbol{\theta}_b = (\theta_{b1}, \theta_{b2}, \ldots, \theta_{b11})$ bounce posture. p.28, p.29, **4.15**

Θ_f set of possible final postures. **4.14**

$\boldsymbol{\theta}_f = (\theta_{f1}, \theta_{f2}, \ldots, \theta_{f11})$ final posture. p.28, **4.14**

Quantiles

$Q_{a \to b}(a, b, p)$ minimum distance between populations a and b so their lower and upper quantiles, for percentage p, do not overlap. p.24, **4.10**, **4.11**

$Q_{\text{dist}}(i, p)$ distance between the quantiles of a population i for the probabilities $\frac{p-1}{2}$ and $\frac{p+1}{2}$. p.xiii, p.24, p.25, **4.11–4.13**

$Q_f(i, p)$ quantile of a population i for probability p. p.22, **4.8**, **4.9**, **4.12**

σ standard deviation. p.22

SODNF

$\underset{obs \to som}{a}(x^{obs}, x^{som}, t)$ adaptable synaptic connections from x^{obs} to $u^{som}(x^{som}, t)$. p.12, p.16, p.17, p.20, p.22, **2.13–2.15**, **2.20**, **3.7**

$c(x^{obs}, t)$ back propagation. p.15, p.20, **2.20**

$\hat{c}(x^{obs}, t)$ normalized back propagation. p.13, p.17, **2.19**, **3.1**

c_T proportionality constant of the iteration time. p.21, p.59–p.63, p.67–p.70, p.74–p.78, p.80–p.86, p.98–p.102, p.104–p.123, **4.2**, **B.2**

$\text{cdist}_{\text{pop}}(a, b, p)$ circular distance between the centre of the dispersion areas of populations a and b, which are the areas delimited by the quantiles of percentiles $\frac{p-1}{2}$ and $\frac{p+1}{2}$. p.23, **4.4**, **4.7**

$\text{centre}(a, p)$ centre of a population based on its quantiles for the $\frac{p-1}{2}$ and $\frac{p+1}{2}$ percentiles. p.23, **4.7**, **4.8**

DNFL a layer of the SODNF composed by a DNF.

A_w amplitude of the interaction kernel. p.11, p.95, **2.11**

C^{som} input gain for the DNFL. p.xix, p.xxi, p.xxiv, p.13, p.20, p.21, p.31–p.33, p.59–p.63, p.67–p.86, p.88, p.98–p.102, p.104–p.123, **2.18**, **3.7**

k_{σ_w} inverse proportionality constant of the standard deviation of the interaction kernel. p.21, **4.1**

l_{stable} length of the stable single peak solution. p.16, p.17, **3.5**, **3.6**

n_{som} number of neurons in the DNFL. p.21, p.95, **4.1**

r resting level. p.xxi, p.10, p.20, p.21, p.59–p.63, p.67–p.70, p.74–p.86, p.88, p.98–p.102, p.104–p.123, **2.9**

$s^{som}(x^{som}, t)$ input to the DNFL. p.13, p.15, p.17, **2.18**, **3.8**

σ_w standard deviation of the interaction kernel. p.11, p.20, p.21, p.59–p.63, p.67–p.70, p.74–p.78, p.80–p.86, p.98–p.102, p.104–p.123, **2.11**, **4.1**

$u^{som}(x^{som}, t)$ neural activation of the DNFL. p.xv, p.12, p.13, p.15, **2.13**, **2.17**, **2.20**

$w(\Delta x)$ interaction kernel. p.10, p.22, **2.9**, **2.11**, **2.12**

w_{inhib} global inhibition. p.11, p.95, **2.11**

ζ^k_{stoch} stochastic noise. p.22, p.95

$\eta(a, b, p)$ minimum circular distance between the centre of the dispersion area of two stable populations, considering the dispersion area with probability p. p.xiii, p.24, p.25, **4.4**, **4.10**

k_{bp} back propagation gain. p.xx, p.xxi, p.xxiii, p.xxiv, p.15, p.20, p.21, p.59–p.70, p.74–p.86, p.98–p.102, p.104–p.123, **3.1**

$\mathrm{k}_{\mathrm{disp}}$ proportionality factor for the maximum dispersion of a population. p.24, **4.13**

k_{η} proportionality factor for the minimum distance between two populations. p.24, **4.10**

l_{min} minimum activation length to consider a peak as valid. p.22

$\hat{Q}^{obs}\left(x^{obs}\right)$ input to the SODNF with normalized energy. p.13, p.17, **2.13**, **2.16**, **2.19**, **3.1**

$Q^{obs}\left(x^{obs}\right)$ input to the SODNF. p.13, **2.16**

SOL a layer of the SODNF composed by a SOM.

$\tau^{a^{som}}$ time constant for the SOL. p.20, p.21, p.59–p.63, p.67–p.70, p.74–p.78, p.81–p.86, p.98–p.102, p.104–p.123, **2.13**

$z^{obs}\big(x^{obs}, t\big)$ input to the SOL. p.13, p.15, **2.18**, **2.19**, **3.1**, **3.7**

T_p iteration time. p.18, p.20, p.21, **4.2**, **B.2**

Transformation matrixes

$\mathrm{Rot}(\mathrm{axis}, \theta)$ 3×3 rotation matrix for a rotation around axis of an angle θ. p.38, p.44–p.46, **4.16**, **4.19**, **4.21–4.25**, **4.30–4.32**, **4.34**, **4.35**, **4.39–4.41**, **4.45–4.47**, **4.51**, **4.57**, **4.60–4.63**, **4.78**, **4.87**, **4.93**, **4.96**, **4.97**, **4.105**, **4.109**, **4.112**, **A.1**, **A.5**, **A.9**, **A.10**, **A.16–A.20**

$\mathrm{Rot_h}(\mathrm{axis}, \theta)$ 4×4 homogeneous rotation matrix for a rotation around axis of an angle θ. p.38, **4.16**, **4.88**

${}^{A}\mathrm{R}_{B}$ total 3×3 rotation matrix from axes A to B. p.38, **4.22**, **4.24**, **4.25**, **4.29**, **4.30**, **4.33**, **4.34**, **4.39**, **4.40**, **4.45**, **4.50**, **4.78**, **A.20**

${}^{A}\mathrm{T}_{B}$ total transformation matrix from axes A to B. p.39, **4.88**, **4.89**

$\mathrm{Trans_h}(p)$ 4×4 homogeneous translation matrix of value p. p.39, **4.88**

Unit vectors of the x, y, z axis

${}^{A}\hat{x}_{B}$ directional unit vector of axis x of referential B relative to axes A. p.39, p.51, **4.22**, **4.90–4.92**

${}^{A}\hat{y}_{B}$ directional unit vector of axis y of referential B relative to axes A. p.39, p.43, p.51, p.53, p.55, **4.22**, **4.34**, **4.35**, **4.48–4.52**, **4.106–4.108**, **4.113**

${}^{A}\hat{z}_{B}$ directional unit vector of axis z of referential B relative to axes A. p.39, p.44, **4.17**, **4.19–4.22**, **4.24**, **4.31**, **4.34**, **4.35**, **4.39**, **4.40**, **4.46**, **4.51**, **4.78**, **4.85**, **4.86**, **4.103**, **4.104**

This page has been intentionally left blank.

List of Figures

This page has been intentionally left blank.

List of Tables

This page has been intentionally left blank.

Chapter 1

Introduction

1.1 Human centred robotics: what happens when new actions are observed?

As prices for robotic systems keep decreasing the technology is quickly disseminating into different market segments. This dissemination can be seen in the appearance of robotic systems in areas such as

- industry [2],
- entertainment [30, 36],
- cleaning [11, 24],
- search and rescue operations [18, 23, 25],
- scientific exploration of hazardous environments [8, 16],
- healthcare [5, 9, 14, 22, 27, 34, 37]
- etc.

In some cases, where a large amounts of Human-Robot Interaction (HRI) may be required, such as healthcare, the acceptance by the user can become a problem. This is due to the disparity between what current robots are capable of and what the user expects from them [4]. Given that the percentage of people aged 60 years or over is expected to surpass 30% in many countries by 2050, increasing at a far greater rate than what has happened in the past [35], this area of application is of particular importance. As such, a system capable of adapting so as to meet the expectations of the user and provide him/her with specialized assistance, allowing for greater independence of the user and reducing the strain on healthcare providers, could be greatly advantageous for society in general. Such a robot must be capable of learning from observing the user and the healthcare providers, so as to adapt to the needs and preferences of the user as they change over time and meet his/her expectations.

1.2 Creation of new representations through Observational Learning

As stated in section 1.1, given the unforeseeable nature of the environment and tasks the robot will encounter, it is impossible to pre-program a general purpose robot for all the tasks and objects that it will encounter. Even if continuous updates were to be programmed, it would be infeasible to program all the possible situations particular to each user and task.

A solution to this problem is endowing the robot with the ability to learn. This can be performed in several ways, one of which is, for instance, the creation of a global database, these database could then be continuously updated with new objects and actions, which a central program could use to generate the new learnt behaviour. However, this constantly updated database would requiring massive amounts of storage, as it would need to store the preferences of all the users and all possible object interactions, as well as an immense processing power and time required to calculate the virtually infinite and constantly increasing possibilities. The robot itself, due to the large amount of possible situations that need to be analysed, would probably also require a very fast processing system, as it would need to determine the correct behaviour in real time. Another approach is that each robot learns individually and creates it's own representations, this approach may still require a robot with powerful processing capabilities but, given that it would only need to account for the specific users case and conditions, the number of possibilities that it would need to known would be far more manageable. As objects are introduced, others are removed from the environment and, therefore, can be forgotten if the memory is insufficient (while the centralised approach would need to maintain the object for longer, as some users would require the same object/task for longer than others).

Since the robot has to be able to learn on its own, it must be able to acquire the data needed for each new task/object independently, through Observational Learning (OL). This means that the robot must be able to analyse its environment and observe, so as to determine what the purpose of each unknown object is, how to interact with it and create its own representations for these tasks/objects. Another important aspect is the need to learn from as few demonstrations as possible – ideally from a single observation as some tasks may require a long time to perform and because the end user will not want to demonstrate the similar tasks repeatedly [7].

There exist three main types of machine learning paradigms – Supervised Learning (SL), Reinforcement Learning (RL), Unsupervised Learning (UL) – each with its own strengths and weaknesses and with various alterations on the base model.

1.2.1 Supervised Learning

SL is based on finding the function that, for a given input set, i.e. training set, minimizes the difference between the obtained outputs and the desired ones, for each input, so that when similar inputs are presented, the correct output will be produced [10, Introduction]. The problem is that this desired outputs have to be known and, therefore, someone must tell the robot what the correct output is. This would mean that each time the user wants the robot to learn something new – which was not contemplated in nor could be deduced from the previous training set – he/she would have to tell the robot, in a way that it could understand, what is expected of it. The system would then need to be retrained so as to contemplate this new situations. As such, this is not the best approach for a system that must be continuously learning new things.

1.2.2 Reinforcement Learning

Unlike in SL that uses a provided mapping between input and expected output, in RL the learning agent must learn through the use of rewards. These rewards are a numeric value received from the environment and the goal of the learning agents is to maximize the total reward. Since each action the agent performs has an impact on the environment, whose state/situation changes due to those actions, the reward received changes depending on this state and the sequence of actions. Because the agent does not know beforehand what the reward is for each action and/or sequences, it must experiment with different actions and their sequence so as to learn the one that will produce the maximum total reward for each state, based on its memory. [33]

Although some rewards could come from other parts of the robot, external to the learning agent – such as the battery if controlling the battery level was one of the goals [33] –, because some goals would only be known to the user, he/she would still need to attribute some rewards. Another problem is that the robot may need to perform many experiments by interacting with the environment which may take a long time, or in a simulator but the new objects and intended goals would still need to be programmed. As such, even though this approach is more suited for incremental learning, a system that could learn simply from observing the user or other people, would be a better match for cases where HRI is intended.

1.2.3 Unsupervised Learning

The UL focusses on analysing the input data and finding similarities in the input, rather than a supplied mapping like in SL or external rewards like in RL. In this way, an internal classification of the inputs is formed based on their intrinsics characteristics. This method on its own does not determine what the right output for each situation is, it only focuses on the creation of internal classification of the input data based on its features that are generic to several cases. This internal classification can then be used to determine the correct output – possibly by using other learning methods – in a simpler way than using the input data directly [3] – and in so doing alleviating some of the problems with SL and RL if used in conjunction. Given that many objects and actions share many characteristics in different tasks, these classifications can be reused in several tasks while adding new and/or more specific classifications.

Because UL is focused on the creation of internal classifications of the input, it does not directly solve the problem of how to decide the correct actions. Despite this, since the user is only required to provide the robot with examples of what is intended, which can be obtained through observation of the actions performed and interactions with objects, this is likely the best method for the first layer of a system capable of learning through HRI using OL. As such, this will be the focus of this dissertation.

1.3 Contributions of this thesis

As stated in subsection 1.2.3, a first step towards endowing a robotic system with the capability of OL is the use of UL methods to generate generic internal representations for the various new objects and actions the robot encounters. These can later be used to determine the correct behaviour or course of action of the robot. One model directed at this task was proposed by Sousa [31], which was based on the combination of the UL method of the Self-Organizing Maps (SOMs) combined with the Dynamic Neural Fields (DNFs). Although in [31] it was shown that the model could be used to classify high-dimensional inputs, only a small data sample of simulated data with no noise was used. This did not demonstrate how the system would cope with real acquired data (specifically movement data), plagued by noise, nor did it allow for a minimal parameter test in order to test the stability of the system when faced with small parameter changes nor the selection of the optimal parameters. To that end, this dissertation re-evaluates the aforementioned model using both a large set of simulated motions as well as motion data acquired using a 3D vision system. These tests were performed and analysed over several parameter combinations, which allowed for the detection and correction of instabilities in the system (that could manifest when certain parameter combinations were used), further improving the performance of the system.

1.4 Thesis outline

The remainder of this thesis is structured as follows:

In chapter 2, an overview of the DNFs and the SOMs will be given along with the describe the base model (which uses these the Neural Networks (NNs)) on which this work is based.

The following chapters are dedicated to the focus of this thesis starting of with the extensions to the base model presented in chapter 3 and goes on to describe the tests performed as well as display and analyse the obtained results in chapter 4.

Finally, chapter 5 will address the conclusions of this work and the future work.

Chapter 2

A biologically inspired model for the creation of representations

As stated in Section 1.3, this work will be based on the model proposed by Sousa [31, Part V] for the development of neural representation for high dimensional data. This model was based on two other neurologically inspired models – the DNF [1] and the SOM [15] – and represents a Self-Organizing Dynamic Neural Field (SODNF).

2.1 The Self-Organizing Map

The SOM is a class of Artificial Neural Networks (ANNs) whose goal is to transform a pattern of input signals of arbitrary dimension into an n-dimensional topographically ordered map (typically of one or two dimensions) [10]. Meaning that the more similar the input patterns encoded by a set of neurons, the closer the neurons are in the field. As such, different regions of neural field will respond to different patterns in the data. This learning process, where the input patters are topographically ordered, is achieved via three principles [10]:

1. **Competition** amongst neurons where the neuron that display the greatest re-

sponse to the input pattern is selected, becoming the winning neuron;

2. **Cooperation** in which the winning neuron defines a topological neighbourhood where the neurons will become excited.

3. **Synaptic Adaptation**, where the connection weights between the input and the excited neurons are updated. Being that the responsiveness of these active neurons to the input pattern increases.

One type of SOM that has seen great use in diverse areas and applications is Kohonen's Self-Organizing Map [12, 26, 29]. From this point forth this will be the model considered for the SOM. Next, the iterative process for the formation of the topographic mapping will be explained for this model.

First the neuron, j, with the highest response to the current input vector, $\vec{q} = [q_1, q_2, ..., q_m]^T$, is chosen as the winning neuron, $j^{win}(\vec{q})$. This is calculated according to the desired metric such as the Euclidean distance or the inner-product. In the case of the Euclidean distance, the objective is to find the neuron with the minimum distance between its connection weights, $\vec{a}_j = [a_{j_1}, a_{j_2}, ..., a_{j_m}]^T$, and $\vec{q}$ [10]:

$$j^{win}(\vec{q}) = \arg\min_j \|\vec{q} - \vec{a}_j\| \ , \ j = 1, 2, ..., n \ , \tag{2.1}$$

where n is the number of neurons. In the case of the inner-product, the objective is to find the neuron with the maximum inner-product between $\vec{a}_j$ and $\vec{q}$ [15, 31]:

$$j^{win}(\vec{q}) = \arg\max_j \left\{ \vec{q}^{\mathrm{T}} \vec{a}_j \right\} \ , \ j = 1, 2, ..., n \ . \tag{2.2}$$

Second, the activity of each neuron, j, is calculated based on their distance to the winning neuron, $j^{win}(\vec{q})$, in accordance to an interaction kernel, $w(j, j^{win}(\vec{q}), k)$. This kernel is a function such that [10]:

1. the maximum excitation is obtained when $j = j^{win}(\vec{q})$ and the excitation of the topological neighbourhood, $w(j, j^{win}(\vec{q}), k)$, is symmetric in relation to the

winning neuron;

2. the level of excitation decreases monotonically with the distance between the neuron j and the wining neuron, being that it decays towards 0 as $\|x_j - x_{j^{win}}\| \rightarrow \infty$, where $x_j \in \mathbb{R}^{\dim}$ is the position of neuron j in a SOM with $\dim$ dimensions. (Required condition for convergence)

One common approach is the use the Gaussian kernel

$$w(j, j^{win}(\vec{q}), k) = e^{-\frac{\|j-j^{win}(\vec{q})\|}{2\sigma^2(k)}} , \tag{2.3}$$

where k is the current iteration. The width of the kernel and, therefore, of the topological neighbourhood is controlled by $\sigma(k)$. This width can decrease with time and is normally described as the exponential decay

$$\sigma(k) = \sigma_0 e^{-\frac{k}{\tau_\sigma}}, \tag{2.4}$$

where σ_0 is the initial value and τ_σ is a time constant [10].

Third, the variation of the weight vector for neuron j, denoted as $\Delta\vec{a}_j$, is calculated using

$$\Delta\vec{a}_j = \eta(k) w\big(j, j^{win}(\vec{q}), k\big)\big(\vec{q} - \vec{a}_j\big) \ , \ j = 1, 2, ..., n \ , \tag{2.5}$$

where $\eta(k)$ is the variation rate for iteration k, and

$$\vec{a}_j(k) = \vec{a}_j(k-1) + \Delta\vec{a}_j \ , \tag{2.6}$$

if the Euclidean distance (and therefore (2.1)) is used. If the inner-product (and therefore (2.2)) is used than

$$\Delta\vec{a}_j = \eta(k) w\big(j, j^{win}(\vec{q}), k\big)\vec{q} \ , \ j = 1, 2, ..., n \ , \tag{2.7}$$

In this case an additional normalization is required to maintain the stability of the weights and (2.6) becomes [15, 31]:

$$\vec{a}_j(k) = \frac{\vec{a}_j(k-1) + \Delta\vec{a}_j}{\|\vec{a}_j(k-1) + \Delta\vec{a}_j\|} \ , \ j = 1, 2, ..., n \ . \tag{2.8}$$

Although the decrease of the width of the neighbourhood allows for a fine tuning of the topological mapping of the input vectors, it also decreases the adaptability to patterns that were not present in the original input data set. When a new input pattern is presented much latter in the learning process, even if the probability of occurrence is high, the area occupied by the pattern will be small due to the decreased kernel width. Given that in some cases it may not be possible to have a representative set of inputs for the situations that will be encountered, this can be a problem for classifying new and important patterns.

2.2 The Dynamic Neural Field

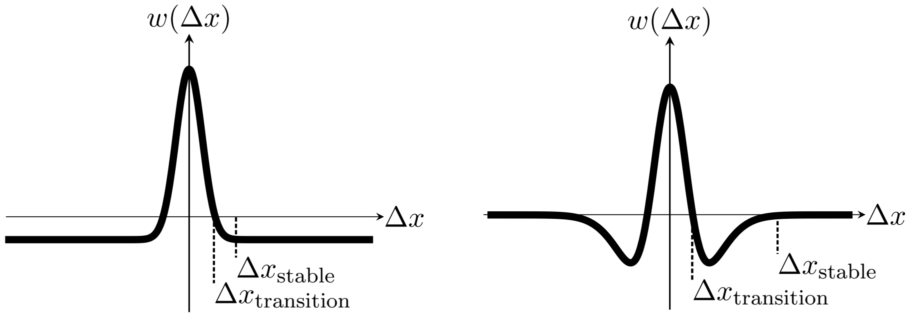
In 1977, Amari [1] proposed a model for the formation and interaction of patterns in the cortical neural tissues

$$\tau \frac{\partial u(x,t)}{\partial t} = -u(x,t) + \int w\big(x - x'\big) f\big(u(x',t)\big)\,\mathrm{d}x' + r + s(x,t) \tag{2.9}$$

where $u(x,t)$ represents the activation of the neural field, τ is the time constant of the field, r is the resting level and $s(x,t)$ corresponds to the field's input. The linear output function, $f(u)$, is a monotonically increasing function with saturation. For mathematical convenience it was defined as the step-function

$$f(u) = \begin{cases} 0 & \text{if } u \leq 0 \\ 1 & \text{if } u > 1 \end{cases} . \tag{2.10}$$

The remaining function is the interaction kernel, $w(\Delta x)$, which controls the interaction between neurons in the same filed. In [1] only the case of fields of lateral inhibition type were analysed. In such fields, neurons in close proximity ($\Delta x < \Delta x_{\text{transition}}$) excite one another and inhibit those at greater distances ($\Delta x > \Delta x_{\text{transition}}$). Two possible kernels of this type can be see in Fig. 2.1. The Gaussian kernel seen in Fig. 2.1(a) is



(a) Gaussian kernel

(b) Mexican hat kernel

Figure 2.1: Lateral inhibition kernels ($\forall \Delta x \geq \Delta x_{\text{stable}} : w(\Delta x) \approx \text{const}$)

characterized by the equation

$$w(\Delta x) = A_w e^{\frac{-(\Delta x)^2}{2\sigma_w^2}} - w_{\text{inhib}}, \tag{2.11}$$

where A_w is the amplitude, σ_w is the standard deviation of the interaction kernel and w_{inhib} is the global inhibition. The mexican-hat kernel seen in Fig. 2.1(b) is characterized by the equation

$$w(\Delta x) = Ae^{\frac{-(\Delta x)^2}{2\sigma_A^2}} - Be^{\frac{-(\Delta x)^2}{2\sigma_B^2}}, \tag{2.12}$$

where $A > B$ and $\sigma_A < \sigma_B$. Several other types of kernels exist, each with a different function. For instance, non symmetric kernels that can help track object movements [32] or oscillating kernels for guaranteeing the existence and stability of multiple active peaks [17] while exiting neurons at specific distances.

Depending on the type of kernel and the parametrization of the field, several cognitive processes can be implemented using a DNF. Some of these processes are [19]:

- **detection**, where only sufficiently strong inputs can form a stable activation peak;
- **decision making**, being that for a sufficiently strong inhibition between the neurons, only one pattern will remain active. All the other inputs and active peaks will

be suppressed, if sufficiently distant from one another, or, if in close proximity, the active peak will take a pondered intermediate positioned, until a strong enough input to suppress the active peak appears;

- **memory**, being that once a stable activation peak is formed it will remain active, even without input, until a new input is sufficiently strong to suppress it or something destabilizes the field;

- and **forgetting** capabilities, by altering the stability of the field so that no peak can be stable.

It is also possible to create a complex architecture for decision making and error detection using interconnected DNFs to perform complex tasks [28]. This system possesses an interaction kernel that shares many similarities with the SOM as it entails competition amongst neurons that are far from one another while exciting those in close proximity, for a Gaussian kernel.

2.3 The Self-Organizing Dynamic Neural Field

By exploiting the similarity between the kernels of the SOM and the DNF, Sousa [31] developed a SODNF composed of two layers. A Self-Organizing Layer (SOL), based on the SOM, that controls the connection weights from the input to a second layer, and the Dynamic Neural Field Layer (DNFL), that contains a neural field whose activation is used to generate the output of the system. The DNFL is formed by a DNF that must be tuned so that only the null solution and a single peak solution are stable. This limitation derives from the fact that the neural activation of the DNFL, $u^{som}(x^{som}, t)$, is used to update the synaptic connections, $\underset{obs \to som}{a}\big(x^{obs}, x^{som}, t\big)$, in the SOL:

$$\tau^{a^{som}} \frac{\partial}{\partial t} \underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) = \Big[u^{som}(x^{som}, t)\Big]_0^+ \hat{Q}^{obs}\big(x^{obs}\big) \tag{2.13}$$

$$\underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) = \frac{\underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big)}{\Big\| \underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) \Big\|_{x^{obs}}} \tag{2.14}$$

$$\Big\| \underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) \Big\|_{x^{obs}} = \sqrt{\int\limits_{x^{obs}} \Big[\underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) \Big]^2 \mathrm{d}x^{obs}} \ , \tag{2.15}$$

where $\hat{Q}^{obs}\big(x^{obs}\big)$ is the input to the SODNF with normalized energy and is defined as

$$\hat{Q}^{obs}\big(x^{obs}\big) = \frac{Q^{obs}\big(x^{obs}\big)}{\sqrt{\int \Big[Q^{obs}\big(x^{obs}\big)\Big]^2 \mathrm{d}x^{obs}}} \ , \tag{2.16}$$

where $Q^{obs}\big(x^{obs}\big)$ is the input to the SODNF. The value of $\Big[u^{som}(x^{som}, t)\Big]_0^+$ is given by

$$\Big[u^{som}(x^{som}, t)\Big]_0^+ = \begin{cases} u^{som}(x^{som}, t) & \text{if } u^{som}(x^{som}, t) > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.17}$$

The input to the DNFL, $s^{som}(x^{som}, t)$, is given by

$$s^{som}(x^{som}, t) = C^{som} \int z^{obs}\big(x^{obs}, t\big) \underset{obs \to som}{a} \big(x^{obs}, x^{som}, t\big) \, \mathrm{d}x^{obs}, \tag{2.18}$$

where C^{som} is the input gain for the DNFL and $z^{obs}\big(x^{obs}, t\big)$ is the input to the SOL. The input to the SOL can be defined as

$$z^{obs}\big(x^{obs}, t\big) = \hat{Q}^{obs}\big(x^{obs}\big) - \hat{c}\big(x^{obs}, t\big) \ , \tag{2.19}$$

where $\hat{c}\big(x^{obs}, t\big)$ is the normalized back propagation. This back propagation is defined as

$$c\big(x^{obs},t\big) = \int \Big[u^{som}(x^{som},t)\Big]_0^+ \underset{obs\rightarrow som}{a}\big(x^{obs},x^{som},t\big)\,\mathrm{d}x^{som} \tag{2.20}$$

and serves as a way of controlling the separation between the formed populations. It works by increasing the divergence velocity in an initial stage when the weights have yet to adapt to the input to the SODNF, but progressively reduces the input to the field as the weights adapt.

Although this model proposed by Sousa [31] was capable of correctly separating a few trajectories in an organized manner, this was only tested with a small data sample of simulated motions. As such it is still crucial to verify the performance of the system with a higher amount of inputs and using acquired data to see if the system can cope with noisy and imprecise data.

Chapter 3

The extended model for the creation of representations

The model proposed in [31] utilized the back propagation, $c\big(x^{obs}, t\big)$, of u^{som} to increase the separation of the populations in the layer (2.19). Despite this, no method for controlling the influence of c was provided. In order to provide this feature a back propagation gain, $\mathrm{k_{bp}}$, was added to (2.19) and thus the input to the SOL, $z^{obs}\big(x^{obs}, t\big)$, is now given by

$$z^{obs}\big(x^{obs}, t\big) = \hat{Q}^{obs}\big(x^{obs}\big) - \mathrm{k_{bp}}\hat{c}\big(x^{obs}, t\big). \tag{3.1}$$

Although the original model was able to create distinct logical representations for certain inputs, in some cases an unstable behaviour could be observed (see Figure 3.1). This behaviour is due to the influence of $\overline{s^{som}}$ on the stability of the DNFL. Amari [1] determined that for an input

$$s(x, t) = \overline{s} + \mathrm{var}(s(x, t)), \tag{3.2}$$

such that $\mathrm{var}(s(x, t))$ represents the variable component of $s(x, t)$, and

$$h = \overline{s} - r, \tag{3.3}$$

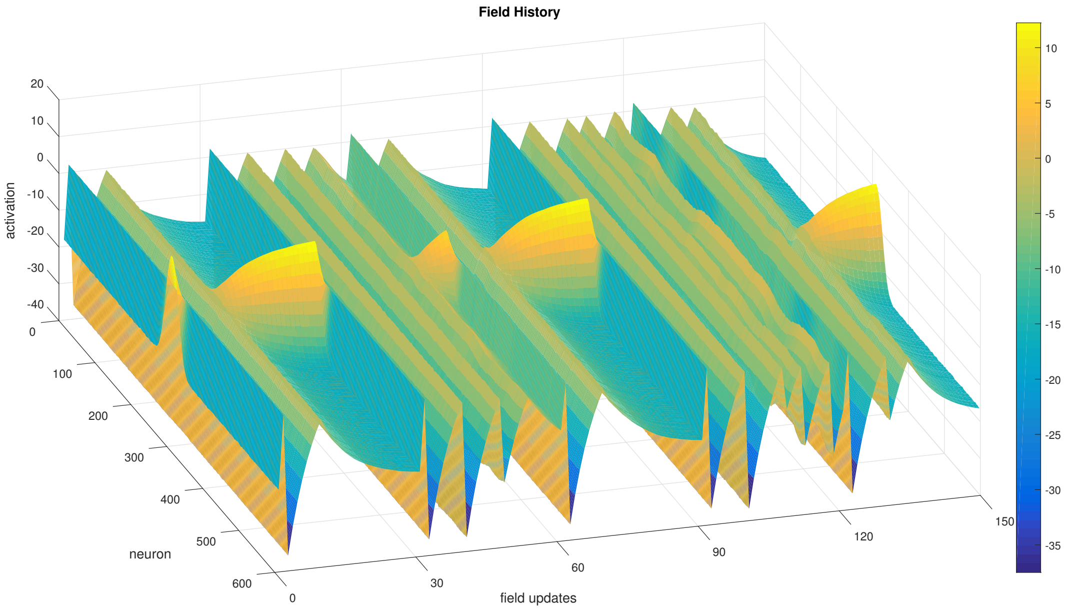


Figure 3.1: Example of an unstable SODNF. Every 30 field updates the DNFL is forcefully reset and the input is altered. It can be seen that for the 2^{nd} and 4^{th} inputs the field is unstable and keeps resetting itself as the whole field tries to become active at the same time

where r is the resting potential, that a DNF with the equation

$$\tau \frac{\partial u(x,t)}{\partial t} = -u + \int w(x-y) f[u(y)] \, \mathrm{d}y + h + \mathrm{var}(s(x,t)), \tag{3.4}$$

with an interaction kernel $w(x)$, has only one stable single peak solution [1, theorem 3] and it can only exist for $h < 0$ [1, theorem 1]. As such, the system must guaranty that $\overline{s} < r$ so that there may be a stable single peak solution, which is needed for updating $\underset{obs \to som}{a}$ (2.13) and obtaining a stable output. Since the length of the stable single peak solution, l_{stable}, is such that

$$W(l_{\text{stable}}) + h = 0 \wedge \frac{\mathrm{d}W(l_{\text{stable}})}{\mathrm{d}l_{\text{stable}}} < 0, \tag{3.5}$$

where $W(x) = \displaystyle\int_0^x w(y) \, \mathrm{d}y$, the length of the stable solution will also vary with $\overline{s}$ (see

Figure 3.2) being that

$$\forall l_{\text{stable}1}, l_{\text{stable}2}, \exists \overline{s_1}, \overline{s_2} : \overline{s_1} < \overline{s_2} \Leftrightarrow h_1 < h_2 \;\Rightarrow\; l_{\text{stable}1} > l_{\text{stable}2}. \tag{3.6}$$

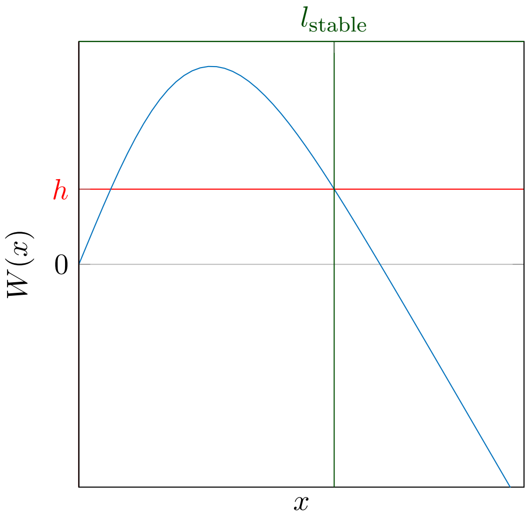


Figure 3.2: Integral of the DNF's kernel $W(x)$ and the length of the stable single peak solution, l_{stable}.

This means that $\overline{s^{som}}$ influences not only the stability of the system but also the importance of each input, as it will activate more neurons, increasing the output to the connected fields as well as the region of $\underset{obs\to som}{a}$ that responds to that input.

Given that $\hat{Q}^{obs}$, $\hat{c}$ and $\underset{obs\to som}{a}$ are all normalized, it is possible to select a value of r such that $\forall s^{som} : h < 0$. However, the importance of each input would still vary with its mean. Since this is an undesirable behaviour, it was opted to add an additional step to the connection between the SOL and the DNFL so that (2.18) becomes

$$s'(x^{som}, t) = C^{som} \int z^{obs}(x^{obs}, t) \underset{obs\to som}{a} (x^{obs}, x^{som}, t)\, \mathrm{d}x^{obs} \tag{3.7}$$

$$s^{som}(x^{som}, t) = s'(x^{som}, t) - \overline{s'} \tag{3.8}$$

such that $\overline{s^{som}} = 0$. Another factor that was altering the importance of each input was the varying iteration times, which were dependent on the duration of the movement. Although this duration can have meaning (from a psychological standpoint for instance), it does not alter the type of movement and therefore should not alter its importance when classifying the motion. To this end it was decided that the iteration time, T_p, should be constant rather than a function of the duration of the movements, see (B.2).

Chapter 4

Model validation

Although in [31] it was shown that the model could be used to classify high-dimensional inputs, only a small data sample – 18 goal directed actions that were decomposed into 2 parts each – of simulated data with no noise was used. This did not demonstrate how the system would cope with real acquired data (specifically movement data), plagued by noise, nor did it allow for a minimal parameter test in order to test the stability of the system when faced with small parameter changes nor the selection of the optimal parameters. To that end, two parameter tests were performed:

- One using a larger set of simulated data
- Another using acquired data with large amounts of noise

These two tests are not exactly the same since they use different objects and conditions as will be seen in the following sections. Despite this, in both cases 5 trials will be performed for each parameter combination, each of the trials using a different subset of the data (in the case of the trials using simulated data) or a different order in which the data is feed into the system (in the case of the trials using acquired data), that is maintained across all the parametrizations.

In [31] compound movements – formed by a reaching and grasping motion (first component) followed by a handover or insertion motion (second component)– were used. The input to the the SODNF was then composed by the colour of the grasped object and the evolution of the joint angles over a normalized time. In the present work this was altered being that only the first component of the movement (reaching and grasping motion) is used, and the input to the field was reduced to the movement data given that it already represents a high-dimensional input, sufficient per se to imply the creation of new neural representations.

4.1 Parameter selection

Testing the influence of each parameter in the system is an impractical process. Thus, it was necessary to limit the number of parameters to analyse in order to keep with time constraints for the project. To that end 6 main relevant parameters were selected:

1. T_p which must guarantee that the DNFL has enough time to form a self-sustained peak and a reasonable amount of time for updating $\underset{obs \to som}{a}$ once the peak has stabilized. This value should be as low as possible given that it can greatly influence the amount of time required for creating the representations;

2. the time constant for the SOL, $\tau^{a^{som}}$, to control the rate at which the connections $\underset{obs \to som}{a}$ are updated, controlling how quickly populations are formed, converge, diverge and are overwritten;

3. r for maintaining the stability of the field;

4. C^{som} which governs the long term effect of an input on the DNFL while ensuring that the input has sufficient amplitude to generate a self-sustainable peak;

5. σ_w to change the neural distance with positive reinforcement and test its influence on the separation distance between the formed populations;

6. $\mathrm{k_{bp}}$ to regulate the influence of c and test its effects.

To ensure that the percentage of a field occupied by a population is not dependent on the number of neurons in the DNFL, n_{som}, the parameter σ_w is set as:

$$\sigma_w = \frac{\mathrm{n}_{som}}{\mathrm{k}_{\sigma_w}}, \tag{4.1}$$

where k_{σ_w} is the inverse proportionality constant of the standard deviation of the interaction kernel. This means that the value of σ_w (parameter of item 5) is not altered directly but by altering k_{σ_w}. Given that the system will be iterated with a fixed time step, Δt, it is possible to define T_p as

$$T_p = c_T \Delta t, \tag{4.2}$$

where c_T is the proportionality constant of the iteration time.

For the purposes of the following tests, the default values for these parameters were the ones presented on Table 4.1.

Table 4.1: Default parameters for the SODNF

Parameter	Value
σ_w	50
$\tau^{a^{som}}$	600
k_{bp}	1
C^{som}	15
c_T	30
r	-0.4

4.2 Criteria for performance evaluation

In order to measure the separation between neural populations it is necessary to track their centroids and, as such, their positions need to be recorded at the moment before a new input is presented. This moment was chosen as its position varies in 2 instances:

1. during the iteration of an input which belongs to that population, when its position is directly altered, and;
2. during the iteration of inputs from other populations, whose position is close enough so that the activated neural region overlaps with the region of the population being analysed.

In both cases, as the system is iterated with a constant input, the position of the centroid will shift until a stable position is reached. From this moment on, it remains in the same approximate position until a different input is demonstrated. Meaning that only the position of the centroids on the moments when the input changes need to be analysed. Given item 2, due to the normalization of $\underset{obs \to som}{a}$ (2.14),the population can be overwritten by other populations as if forgotten. This means that, only the position after an input of that population was demonstrated – case described in item 1 – can be determined with certainty and thus only this position will be analysed.

Although the centroids are a more accurate descriptor for the position of a neural population than the centre of the activated region, they take longer to calculate. Given a symmetric w and an input with a small amplitude, so as to only cause minimal distortion of the self-sustained peak, the central position may be a sufficient approximation for the centroid. In such cases we can use the former rather than the latter. As such, **all the statistics will be based on the position of the centre of a neural population, if it is adequate replacement for the centroid**. Being that centre of the neural population shall only be considered as valid if the activation length is greater than the minimum activation length, $l_{\min}$, so as to not consider peaks due to the stochastic noise, ζ^k_{stoch}, or other small noises.

Once the positions of the populations have been determined it is possible to calculate several statistics about them such as:

- mean and standard deviation, σ, of the position;
- median and quantiles, Q_f, of the position;

- minimum, average and maximum distance between the current/mean/median position of the populations.

From these statistics it is then possible to determine the success rate of the algorithm in creating logical neural representations from the demonstrated data. Ideally, the formed representation would group all the demonstrations for each scenario (c.f. Figure 4.2 and section 4.4) as a separate population with a stable position and small dispersion – small variations of the median, relative to the size of the field, and small Interquartile range (IQR), relative to the size of the field. In order to test this, it is necessary to specify when two populations are considered as separate or grouped together. Given the set of all populations, P, the set of all stable populations, $P_s \subseteq P$, and the set of populations grouped with stable population i, $G(i)$, such that

$$\forall i \in P_s : i \in G(i) \wedge G(i) \subseteq P_s, \tag{4.3}$$

then

$$\begin{aligned} \forall a, b, c \in P_s :\ \mathrm{cdist}_{\mathrm{pop}}(a, b, p) < \eta(a, b, p) \ &\Rightarrow\ b \in G(a) && (4.4)\\ b \in G(a) \ &\Rightarrow\ G(a) = G(b), && (4.5)\\ b \in G(a) \wedge b \in G(c) \ &\Rightarrow\ G(a) = G(c), && (4.6)\end{aligned}$$

where $\mathrm{cdist}_{\mathrm{pop}}(a, b, p)$ is the circular distance between the centre of the dispersion areas of populations a and b, which are the areas delimited by the quantiles of percentiles $\frac{p-1}{2}$ and $\frac{p+1}{2}$. As such,

$$\mathrm{cdist}_{\mathrm{pop}}(a, b, p) = \ \mathrm{cdist}_{\mathrm{points}}(\,\mathrm{centre}(a, p),\ \mathrm{centre}(b, p)) \tag{4.7}$$

where $\mathrm{cdist}_{\mathrm{points}}(x, y)$ is the circular distance between two points and $\mathrm{centre}(a, p)$ is the centre of a population based on its quantiles for the $\frac{p-1}{2}$ and $\frac{p+1}{2}$ percentiles, defined as

$$\mathrm{centre}(a,p) = \frac{1}{2}\left[Q_f\left(a, \frac{p+1}{2}\right) - Q_f\left(a, \frac{p-1}{2}\right)\right] + Q_f\left(a, \frac{p-1}{2}\right) \tag{4.8}$$

$$= Q_f\left(a, \frac{p+1}{2}\right) + \frac{1}{2}Q_f\left(a, \frac{p-1}{2}\right). \tag{4.9}$$

We define $\eta(a,b,p)$ in (4.4) as the minimum circular distance between the centre of the dispersion area of two stable populations, considering the dispersion area with probability p:

$$\eta(a,b,p) = \mathrm{k}_\eta Q_{a\to b}(a,b,p) \tag{4.10}$$

where k_η is the proportionality factor for the minimum distance between two populations and $Q_{a\to b}(a,b,p)$ is the minimum distance between populations a and b so their lower and upper quantiles, for percentage p, do not overlap. Meaning that

$$Q_{a\to b}(a,b,p) = \frac{1}{2}\left[Q_{\mathrm{dist}}(a,p) + Q_{\mathrm{dist}}(b,p)\right]. \tag{4.11}$$

where $Q_{\mathrm{dist}}(i,p)$ is the distance between the quantiles of a population i for the probabilities $\frac{p-1}{2}$ and $\frac{p+1}{2}$, defined as

$$\forall i \in P : Q_{\mathrm{dist}}(i,p) = \frac{1}{2}\left[Q_f\left(i, \frac{p+1}{2}\right) - Q_f\left(i, \frac{p-1}{2}\right)\right] \tag{4.12}$$

(see Figure 4.1). Therefore, so long as $\mathrm{k}_\eta \geq 1$, no two stable primitives in the same group have overlapping areas of probability p.

A stable population is defined as any population i such that

$$\forall i \in P : Q_{\mathrm{dist}}(i,p) \leq \underset{x\in P}{\mathrm{mean}}\big(Q_{\mathrm{dist}}(x,p)\big) + \mathrm{k}_{\mathrm{disp}}\,\underset{x\in P}{\mathrm{std}}\big(Q_{\mathrm{dist}}(x,p)\big) \;\Rightarrow\; i \in P_s \tag{4.13}$$

where $\mathrm{mean}_{x\in P}$ and $\mathrm{std}_{x\in P}$ are the mean and standard deviation $\forall x \in P$, respectively. The parameter $\mathrm{k}_{\mathrm{disp}}$ is the proportionality factor for the maximum dispersion of a population and is defined as The unstable populations are ignored for the formation of groups as, due to their abnormal dispersion relative to the rest of the data, they may cause well

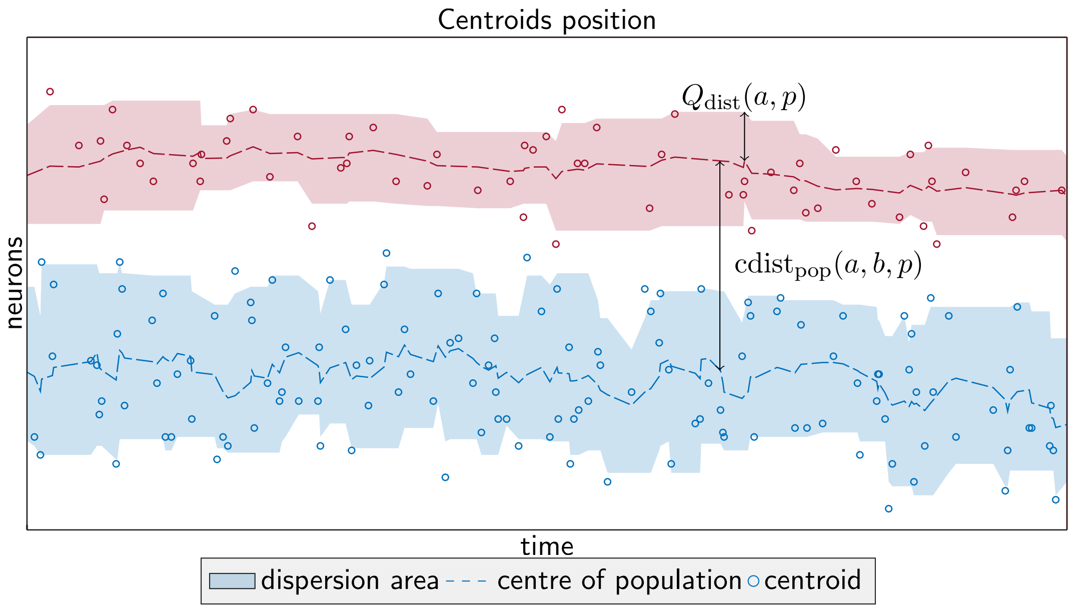


Figure 4.1: Example of the distance between two populations

defined and separated groups to be considered as a single group due to the presence of outliers.

As such, the parameters for analysing the data are:

- the fraction of the iterations that must have produced a stable centroid in the DNFL, $\mathrm{success\,rate}$;
- the number of previous centroid positions over which to evaluate if $Q_{\mathrm{dist}}(i,p) > \eta(a,b,p)$ for any two populations, $\mathrm{eval\,window}$;
- the probability of the centroid being in the area of dispersion considered, p;
- the fraction of the previous $\mathrm{eval\,window}$ iterations during which two populations must have a distance above $\eta(a,b,p)$ to be considered as separated, $\mathrm{th}_{\mathrm{eval\,window}}$;
- the number of previous iterations of a population used for calculating its current dispersion area (Q_{dist}), Q_{window};
- the fraction of the number of populations that should not be ignored, relative to the total number of populations, $\mathrm{th}_{\mathrm{disp}}$;

- and the fraction of the total iterations during which a population must have a dispersion inferior to the minimum dispersion at that time to not be ignored, $\text{th}_{\text{ignored}}$.

4.3 Tests with Simulated movements

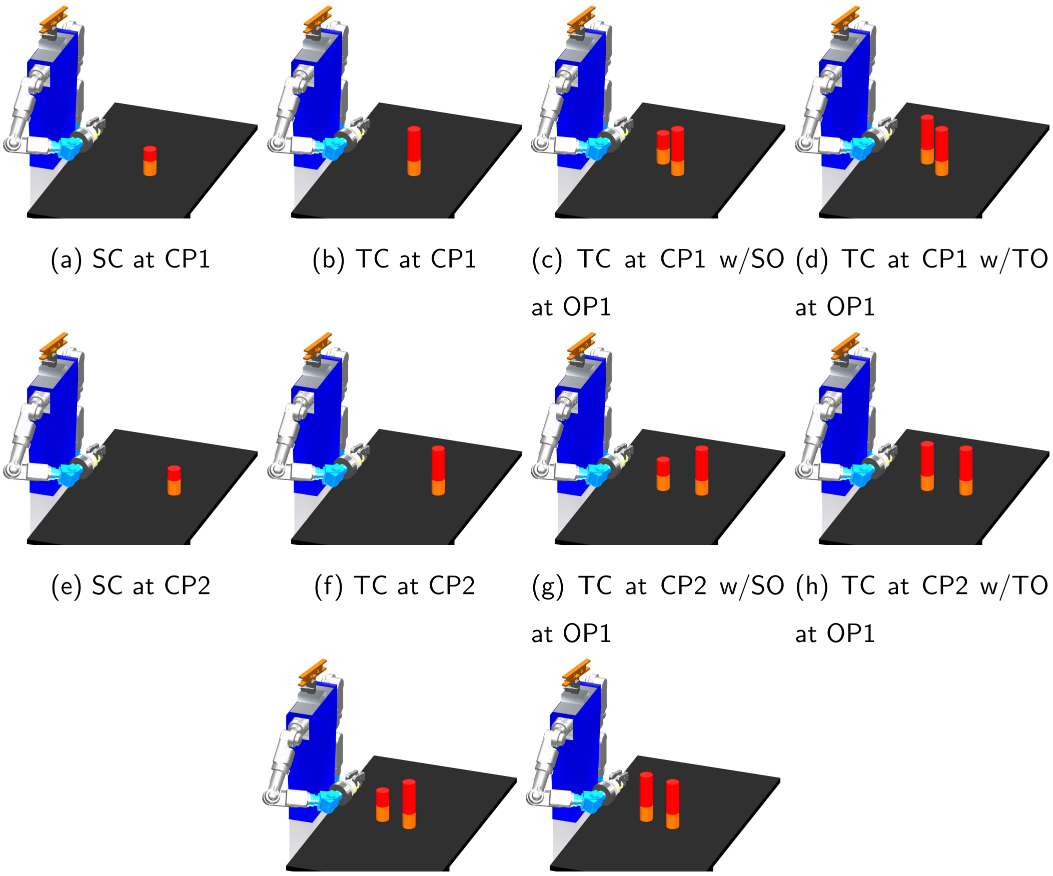
In order to obtain a considerably large data sample of trajectories with controlled variability it was not feasible to use real data. It would be nearly impossible to perform the same movement hundreds of times with controlled variation and, for small variations, the noise would likely represent the majority of the difference. As such, it was decided to use a simulator for generating human-like trajectories developed in the Mobile and Anthropomorphic Robotics Laboratory (MARLab) [6].

4.3.1 Experimental set-up

Using the aforementioned simulation software, 10 scenarios were created. In these the Anthropomorphic Robotic System (ARoS) must reach to grasp a cylinder, with varying hight and/or position, while possibly having to avoid a short or tall obstacle in different positions (see Figure 4.2). These scenarios were chosen because they allow for testing:

- changes in the target position along the 3 spacial axis, given the different (x, y) position of the target object and the variation of the grasp position in the z axis between the shorter and taller cylinders (see Figure 4.3);
- the presence of an obstacle of varying size and/or in a different position.

After generating the dataset of simulated trajectories, the same tests were run 5 times for each of the analysed combinations of the test parameters (see section 4.1), to see if the results were consistent. Because of the time required for testing with a large amount of data, the dataset used for each trial is composed of a unique subset of the complete database (each trajectory can appear in 2 or more trials but not all



(a) SC at CP1

(b) TC at CP1

(c) TC at CP1 w/SO at OP1

(d) TC at CP1 w/TO at OP1

(e) SC at CP2

(f) TC at CP2

(g) TC at CP2 w/SO at OP1

(h) TC at CP2 w/TO at OP1

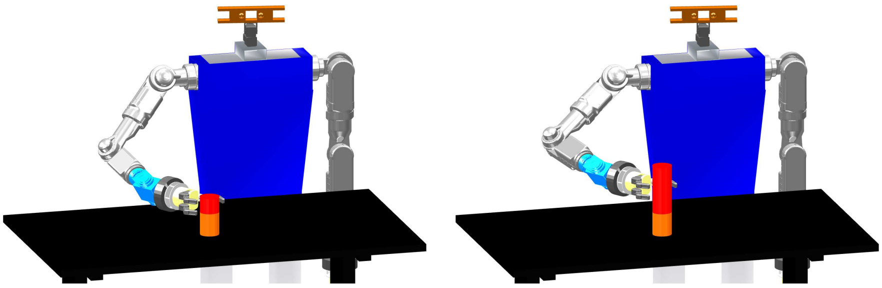
(i) TC at CP1 w/SO at OP2

(j) TC at CP1 w/TO at OP2

Figure 4.2: Simulated scenarios where: SC – Short Cylinder, TC – Tall Cylinder, CP1 – Cylinder Position 1 (close and to the right), CP2 – Cylinder Position 2 (far and to the centre), SO – Short Obstacle, TO – Tall Obstacle, OP1 – Obstacle Position 1 (further from the robot) and OP2 – Obstacle Position 2 (closer to the robot)

the trajectories can be the same). Each subset is formed by randomly selecting unique trajectories. The order in which the trajectories are used is selected randomly and each trajectory is shown only once.

Ideally, the robot would be able to create a neural representation for each scenario and arrange them so that the populations will be organized based on their similarity (for instance, the populations representing the target cylinder being in position 1 should be



(a) End posture for scenario (a)

(b) End posture for scenario (b)

Figure 4.3: End postures for scenarios (a) and (b)

closer to each other than to those referring to position 2 or, the populations representing the short cylinder being closer to each other than to the ones representing the tall cylinder).

4.3.2 Generating movements

As mentioned in the introduction of section 4.3, the simulation software selected for this work was designed to generate human-like trajectories [6]. This is an important feature as one of the intended purposes of the system presented in this work is to autonomously generate neural representations of observed human actions. Another important feature of this software is the ability to define the joint expense factors, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{11})$. These are used for determining the arm (first seven joints) and hand (remaining joints) joint angles for the final posture, $\boldsymbol{\theta}_f = (\theta_{f_1}, \theta_{f_2}, \ldots, \theta_{f_{11}})$, and the bounce posture, $\boldsymbol{\theta}_b = (\theta_{b1}, \theta_{b2}, \ldots, \theta_{b11})$, based on

$$\boldsymbol{\theta}_f = \min_{\boldsymbol{\theta}_f \in \Theta_f \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \big(\theta_{f_k} - \theta_{0k}\big)^2, \lambda_k \geq 0 \tag{4.14}$$

$$\boldsymbol{\theta}_b = \min_{\boldsymbol{\theta}_b \in \Theta_b \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \big(\theta_{bk} - \theta_{0k}\big)^2, \lambda_k \geq 0 \tag{4.15}$$

where $\boldsymbol{\theta}_0 = (\theta_{01}, \theta_{02}, \ldots, \theta_{011})$ is the initial posture [6, section 3.1]. As such, by altering these weights, it is possible to generate a large set of trajectories for the arm, with a controlled degree of variation, in a relatively small amount of time (something that would not be possible from observing a human).

In order to obtain several trajectories for the same end posture it was decided that, for each trajectory for which a collision was detected, 10 different bounce postures would be used instead of just the optimum posture, as was the case in [6]. As in this case the trajectories do not have to be calculated in real time as the robot is not executing a task, more complex and time consuming search grids for the bounce points (from which the bounce postures are calculated using inverse kinematics) than those used in [6] are possible. As such, a tighter spherical grid, originating in the mid point of the base trajectory (see Figure 4.4) is used, so as to minimise the displacement of the arm. The selected bounce positions are the first 10 that appear in the search order (which starts with the points closest to the mid point of the base trajectory and progressively increases the search radius), for which a collision is not detected [6, section 3.4].

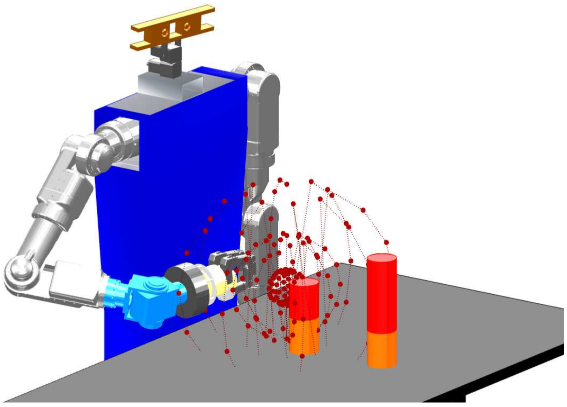


Figure 4.4: Possible bounce positions for scenario (g)

By giving each of the expanse factors for the arm joints, $(\lambda_1, \ldots, \lambda_7)$, a combination of 3 values – 1, 10 and 100 – and repeating the process several times, around 7800 trajectories were generated for each scenario (as can be seen in Table 4.2).

Scenario	Trajectories	%
1	7457	9.57
2	8675	11.14
3	7667	9.84
4	8351	10.72
5	7947	10.20
6	6291	8.08
7	7978	10.24
8	8503	10.92
9	7568	9.72
10	7462	9.58
Total	77899	100

Table 4.2: Number of trajectories per simulated scenario and overall percentage

4.3.3 Results and analysis

Given the time constraints for this work and the amount of time required for calculating and analysing each trial of each parametrization (as mentioned in subsection 4.3.1), the number of inputs for this simulation was set at 15000 for each trial, corresponding to less than 20% of the total trajectories. This number was chosen so as to give the field sufficient time to stabilize. Due to the variability of the simulated motions, caused by the wide range of expense factors, coupled with the small number of trajectories for each expanse factor and the small percentage of used trajectories for each trial, it becomes difficult to apply the criteria described in section 4.2 to the scenarios described in Figure 4.2. Despite this, the separation by scenario is the most logical representation of the data from the perspective of a human user, either performing a collaborative task or teaching the robot, even though it may not the most adequate criteria for considering two trajectories as belonging to the same basic type from the perspective of the robot, it is the one that the user will most likely expect and desire, and therefore the criteria

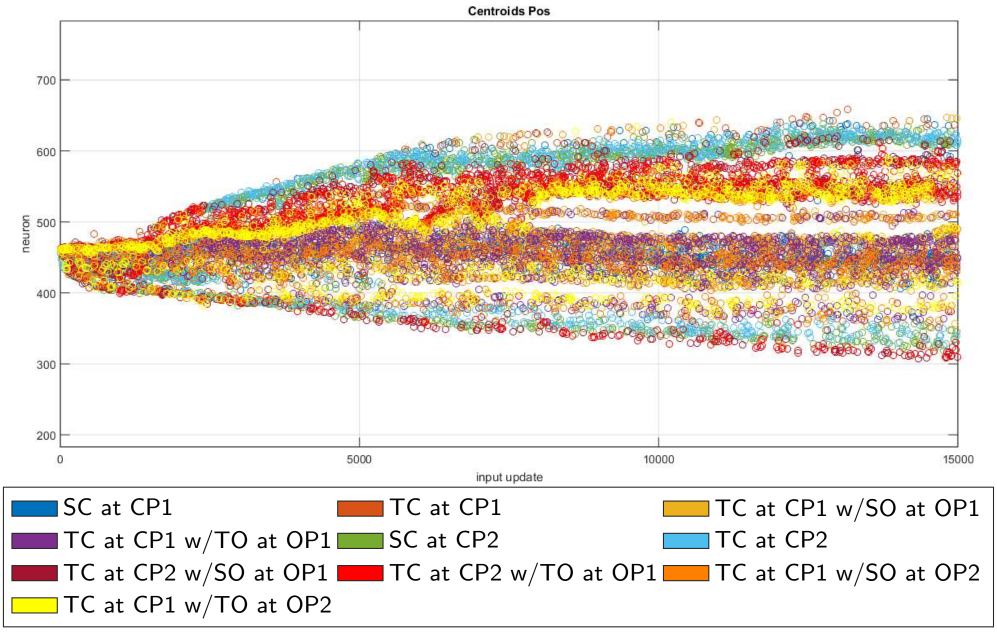


Figure 4.5: Results from 2$^{\text{nd}}$ trial for $C^{som} = 20$

used. Being that the case the system will produce results such as the ones displayed in Fig. 4.5. From this figure alone it is hard to see the overlapping areas for the various centroid positions from the different base populations. To that end Figs. 4.6 to 4.8 were generated, in which only part of the populations are shown. Using these figures, it is possible to see that some of the trajectories were placed in a different position than the rest of the trajectories in the same population, this was most likely due to a jump in the value of the joint expense factors, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{11})$, producing a trajectory through a considerably different path. Another observation that can be made is the grouping of the majority of the trajectories with the same target position. In the case of the target cylinder in position 1 (CP1), which is closer to the robot and to the side of the obstacle in position 1 (OP1) making the motion required for grasping the target easier and more direct, there was no obvious distinction between the presence or absence of the obstacle as seen in Fig. 4.7 (see Figure 4.2). In contrast, in the case of the target cylinder in position 2, there was a slight difference in the position of the centroids though it remained far closer to the rest of the trajectorises in the same position than those for CP1. This separation is most likely due to this position being more obstructed by an

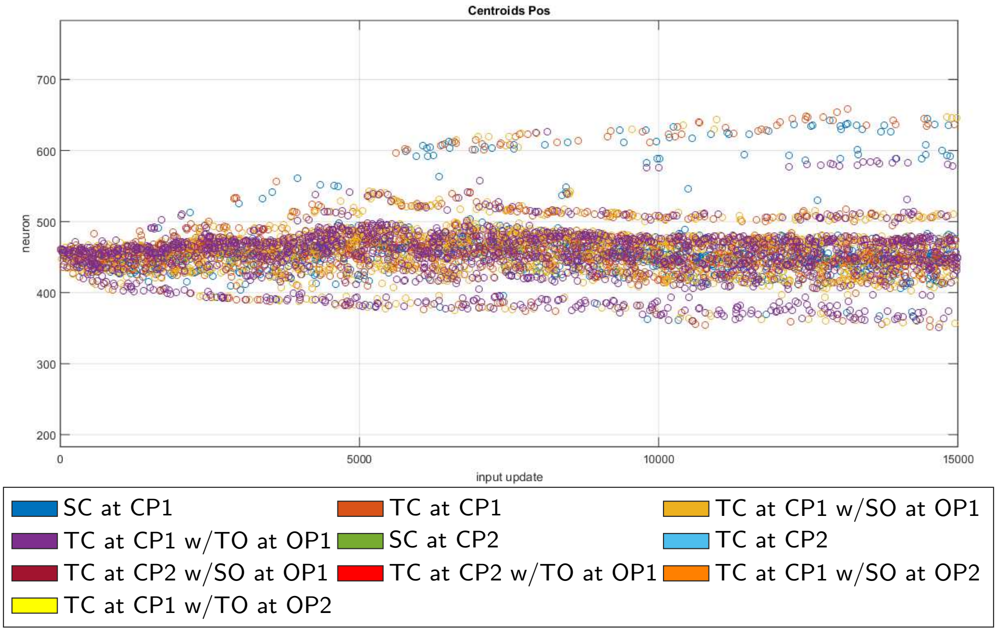


Figure 4.6: Results from 2nd trial for $C^{som} = 20$, for only the objects with target in CP1 and no obstacle in OP2

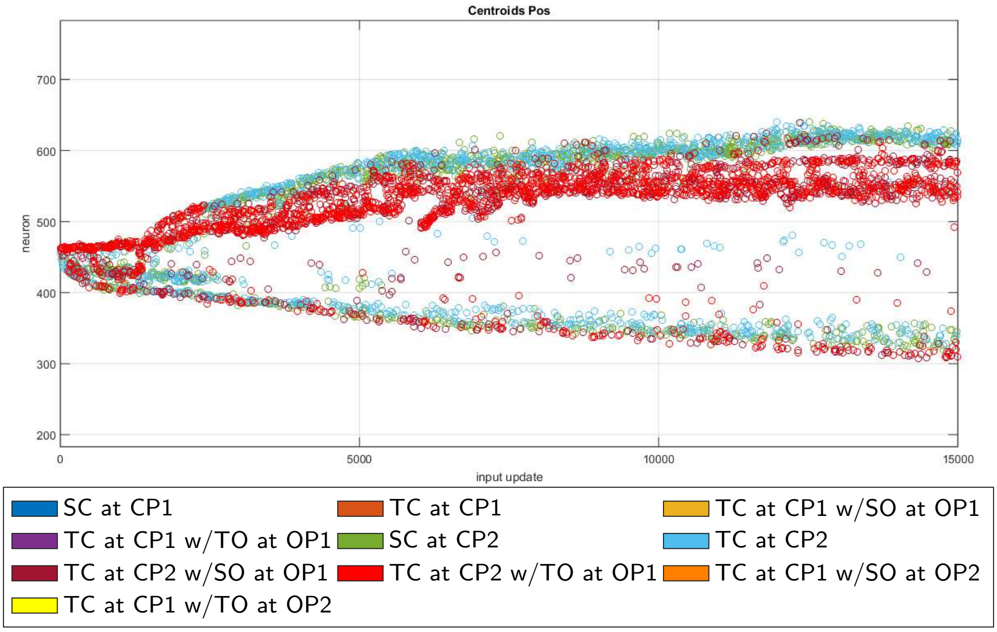


Figure 4.7: Results from 2nd trial for $C^{som} = 20$, for only the objects with target in CP2

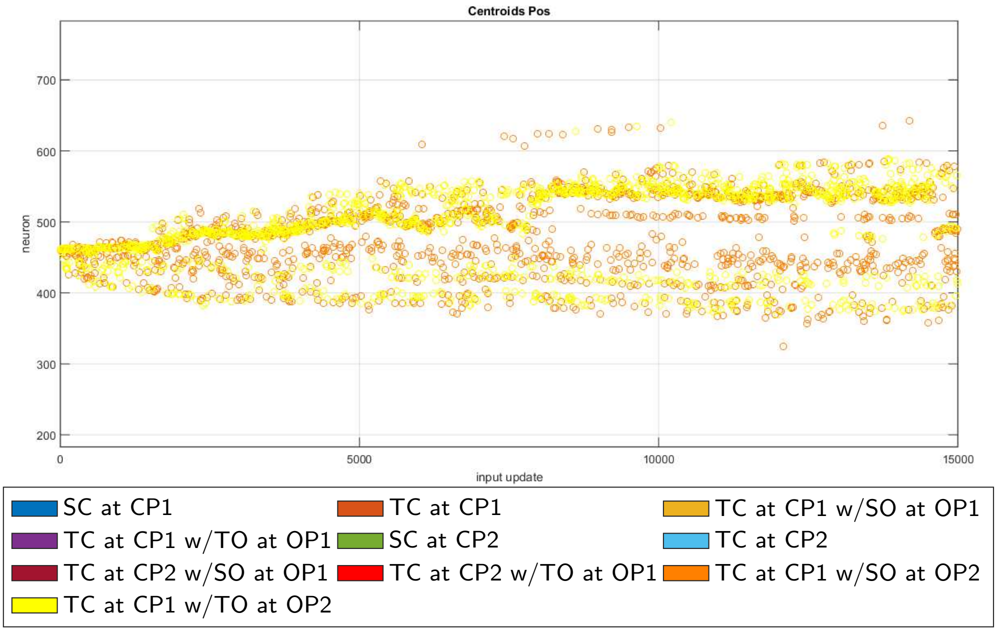


Figure 4.8: Results from 2$^{\text{nd}}$ trial for $C^{som} = 20$, for only the objects with target in CP1 and no obstacle in OP2

obstacle in OP1 than CP1 (see Figure 4.2). In the case of the remaining 2 situations the object is causing the most obstruction so it stands to reason that since the trajectories diverge more frequently than in the other case, so does the centroid position. From this example it was possible to see that even though not all the trajectories were groupped together there was still a logical relation between the positions of the centroids and the similarity of the situations.

4.4 Tests with Observed movements

4.4.1 Experimental set-up

To test the real world application of the system 2 target objects of similar shape are used – a nut and a wheel (see Figure 4.9). These objects were chosen because of their small height which reduced the chances of occluding the arm of the human demonstrator. Unlike the cylinders used for the simulated scenarios, these objects have virtually the

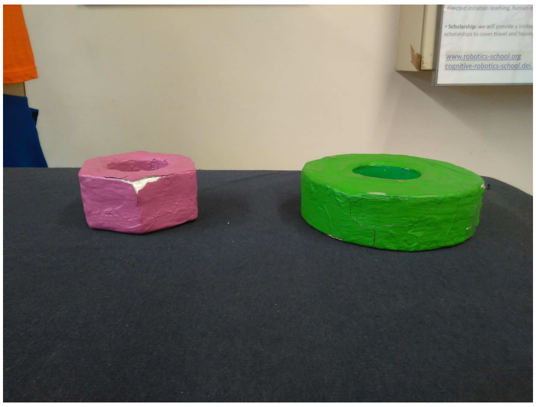
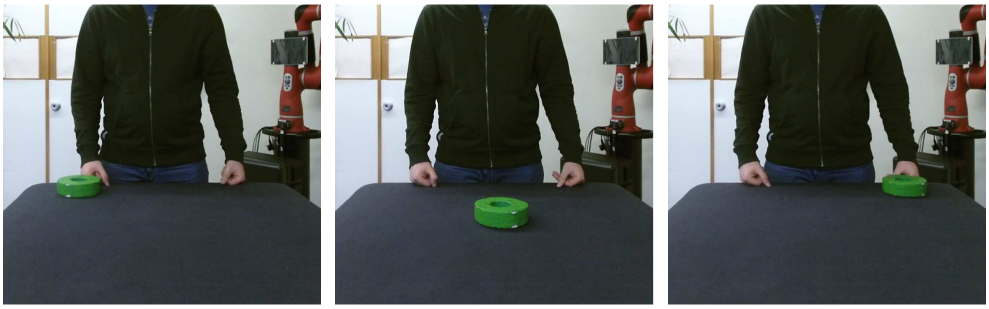


Figure 4.9: Objects used for acquiring data – nut on the left and wheel on the right

same height so the grasping position should not be significantly different in terms of z axis position. Despite this, the (x, y) position of the objects will still vary considerably as the objects will be placed around 3 reference positions (see Figure 4.10), not straying further than their own radius from each of the corresponding reference positions.



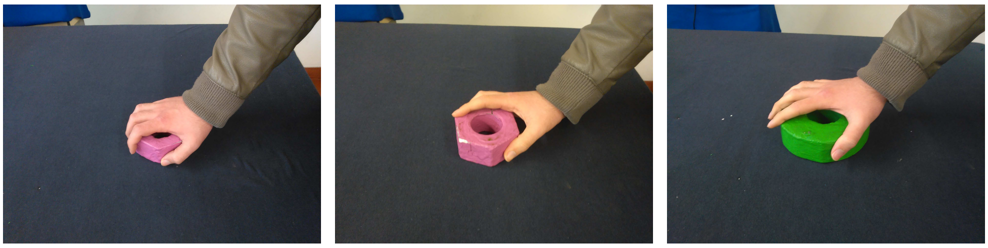
(a) To the right of the demonstrator

(b) At the centre and far away from the demonstrator

(c) To the left of the demonstrator

Figure 4.10: Approximate real world positioning of the objects (position is viewed from the perspective of the observer)

Another difference from the simulated scenario is the different grasping possibilities for the nut (see Figure 4.11(a) and (b)) being that it can be grasped in 2 ways – above grip and side grip. Although the wheel is also grasped using an above grip (see Figure 4.11(c)), because of the dimensions of the object, the posture tends to be slightly different from the same grasp type for the nut. This adds variability that would be desirable for the system to pick up on. From these differences in objects, position and grasp type 9 scenarios can be formed, encompassing all possible combinations.



(a) Grasp Nut with Above Grip (GNAG)

(b) Grasp Nut with Side Grip (GNSG)

(c) Grasp Wheel with Above Grip (GWAG)

Figure 4.11: Possible grasping types for each object

Given that the objective of this work is not the accurate tracking of a human demonstrator, these scenarios do not feature an obstacle, contrary to what was done in the simulated scenarios, as it could compromise the tracking of the arm by occluding it.

Since the intended application scenario for this system is learning from user demonstration, it must be able to learn from only a few demonstrations. Seeing how the users would not want to repeat the same task multiple times and because certain demonstration may require large amounts of time to perform [7]. As such, only a small number of demonstration should be performed for each of the 9 scenarios, meaning that there will be a small dataset with which to work. Because of this, contrary to the simulated scenario, each trajectory will have to be used several times. To that end, each of the 5 trials performed in this stage has a fixed randomly selected order in which the inputs are demonstrated, being that each trajectory is use approximately the same number of times.

The desirable outcome of these tests is that the SODNF will be able to distinguish between the motions for an above grip and a side grip, between grasping from close to each of the reference positions and, as mentioned above, between grasping a nut or a wheel, while organizing them in a logical way – placing the more similar motions closer to each other in the DNFL.

4.4.2 Kinect for movement data acquisition

In order to obtain the joint angles from observation it was decided to use a Kinect Sensor for Xbox One and an Adapter for Windows along with the Kinect for Windows SDK 2.0 [1] so as to use its body tracking functionality, which tracks 25 skeletal joints per person [21] (see Figure 4.12).

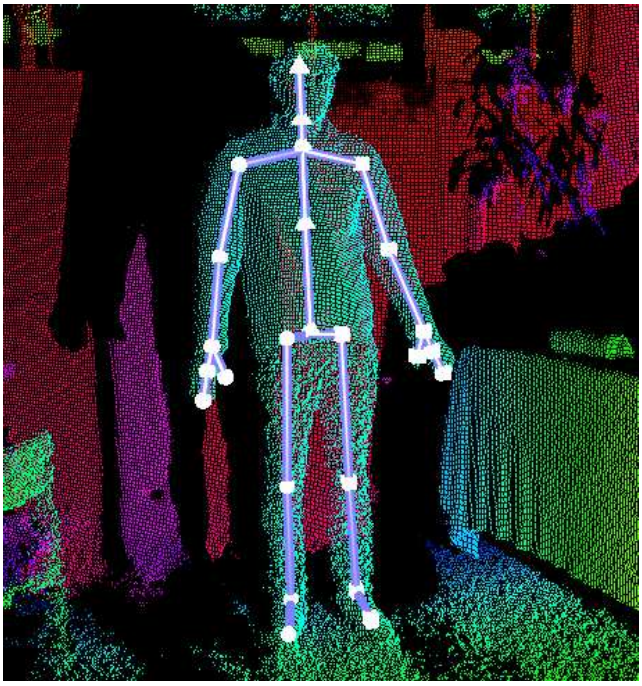


Figure 4.12: Overlay of a body frame acquired using Kinect Studio on the depth frame

Although the Kinetic SDK 2.0 is capable of determining the rotation quaternions for each joint, from which it is possible to calculate the angles for each of the arm joints, how this is done is not described. As such, in order to have a better understanding of the process for calculating the joint angles and to avoid dependence on the specific hardware and/or software, the decision was made to implement the code for determining the joint angles based on the joint positions. These points are still calculated using the Software Development Kit (SDK) but could be obtained through other image processing techniques or using visual markers.

[1] The specific version was 2.0.1410.19000, publised 10/21/2014[20]

4.4.3 Determining the joint angles

For the purposes of this thesis only the arm joints were considered. Although the arm only has three joints (shoulder, elbow and wrist), unlike the robotic arm, each joint can have multiple Degrees of Freedom (DoFs) and therefore multiple angles per joint. These angles are distributed in the following way:

- the shoulder joint has three DoFs associated with 3 angles ($\theta_{\text{shoulder}\,z}$, $\theta_{\text{shoulder}\,x}$ and $\theta_{\text{shoulder}\,y}$);
- the elbow joint has one DoF associated with the elbow rotation, θ_{elbow};
- the shoulder joint has three DoFs associated with 3 angles ($\theta_{\text{wrist}\,z}$, $\theta_{\text{wrist}\,x}$ and $\theta_{\text{wrist}\,y}$).

In order to determine the rotation angles, one must first define the reference axes for the rotation. In this work it was decided that the reference axes would be defined such that the z axis is oriented in the direction of the supporting link – the link that connects the previous joint in the hierarchy to the desired joint – the y axis is defined so that it corresponds to the axis of rotation for the joints that present only one possible axis of rotation, and the x axis corresponds to the normal of the rotation that transforms the y axis into the x axis. In the case of the joints that present more than a single axis of rotation, the axes is chosen so as to minimize the rotation from one joint to the next in the hierarchy (from the hip centre to the extremities). Given the reference position shown in Figure 4.13, the axes of rotation are the ones presented in Figure 4.14.

Let us consider a nomenclature where:

- ${}^{A}P_{\text{joint}} = (x, y, z)^T$ is the position of joint on the axes of joint A,
- ${}^{A}\vec{v}_{\text{link}} = {}^{A}\overrightarrow{P_{\text{prev}}P_{\text{joint}}}$ is the vector of the link named link that ends in joint and starts in joint prev, the previous joint in the hierarchy relative to joint, on the axes of joint A,

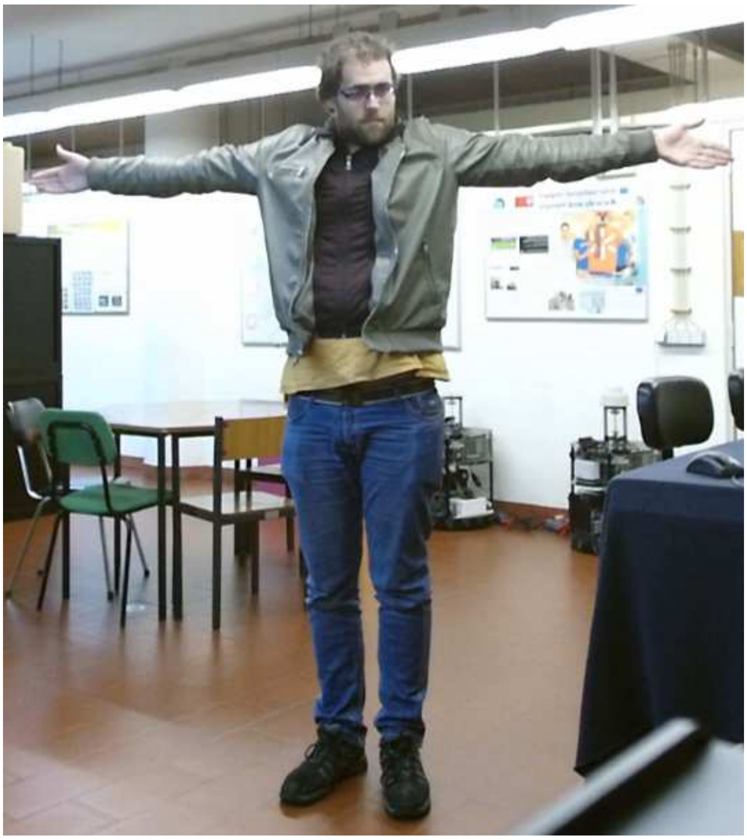


Figure 4.13: Reference position – facing towards the camera with both arms stretched out to the sides and the palm of the hands turned towards the camera

- $v_{\mathrm{link}i}$ is the i^{th} element of ${}^{A}\vec{v}_{\mathrm{link}}$,
- ${}^{A}\hat{v}_{\mathrm{link}}$ is the unit vector of ${}^{A}\vec{v}_{\mathrm{link}}$,
- $\mathrm{Rot}(\mathrm{axis}, \theta)$ is the 3×3 rotation matrix for a rotation around axis of an angle θ,
- ${}^{A}\mathrm{R}_{B}$ is the total 3×3 rotation matrix from axes A to B,
- $\mathrm{Rot_h}(\mathrm{axis}, \theta)$ is the 4×4 homogeneous rotation matrix for a rotation around axis of an angle θ defined as

$$\mathrm{Rot_h}(\mathrm{axis}, \theta) = \begin{pmatrix} & & & 0 \\ & \mathrm{Rot}(\mathrm{axis}, \theta) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4.16}$$

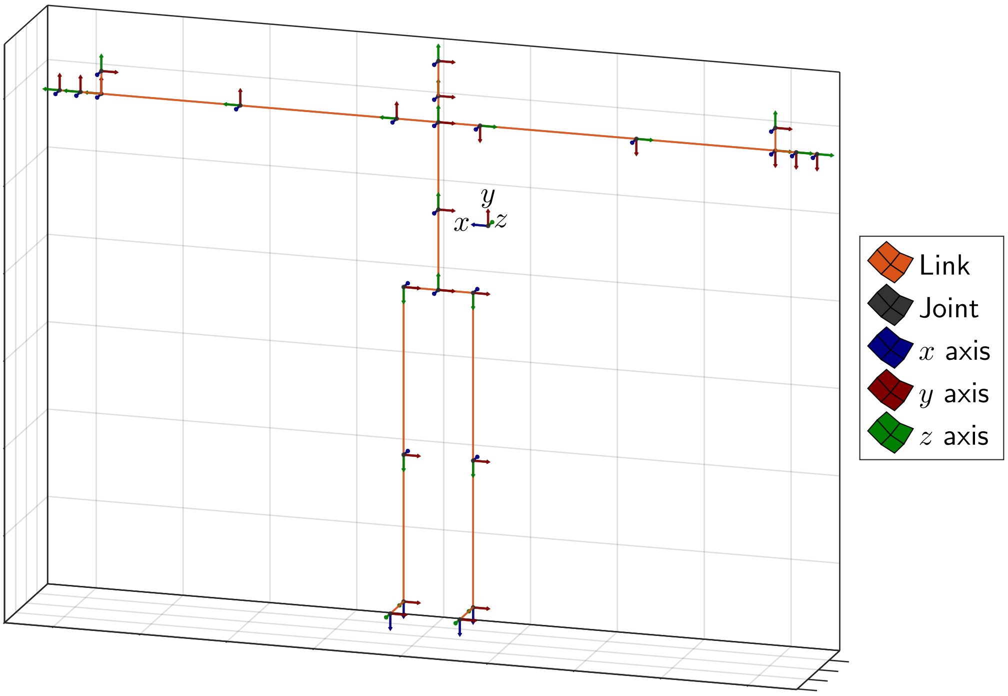


Figure 4.14: Reference position and axes of rotation for calculating the joint rotations, as seen from the perspective of the camera (labelled axes), when the person is facing towards the camera (see Figure 4.13, note: not to scale)

- $\mathrm{Trans_h}(p)$ is the 4×4 homogeneous translation matrix of value p,
- ${}^{A}\mathrm{T}_{B}$ is the total transformation matrix from axes A to B, and
- ${}^{A}\hat{x}_B$, ${}^{A}\hat{y}_B$, ${}^{A}\hat{z}_B$ are the directional unit vectors of axis x, y, z, respectively, of referential B relative to axes A.

Given the reference axes chosen,

$$ {}^{A}\hat{v}_{\mathrm{link}} = {}^{A}\hat{z}_{\mathrm{joint}}, \tag{4.17} $$

which means that

$$ {}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} = (0, 0, 1)^T \tag{4.18} $$

and

$$
\begin{aligned}
{}^{B}\vec{v}_{\text{link}} = \text{Rot}\Big({}^{A}\hat{z}_{\text{joint}}, \theta_z\Big)\,{}^{A}\vec{v}_{\text{link}} \;&\Rightarrow\; {}^{B}\vec{v}_{\text{link}} = {}^{A}\vec{v}_{\text{link}} && (4.19)\\
&\Rightarrow\; {}^{B}\hat{v}_{\text{link}} = {}^{A}\hat{z}_{\text{joint}}, && (4.20)
\end{aligned}
$$

where link is the link that ends in joint and starts on the previous joint in the hierarchy, prev. As such

$$
\begin{aligned}
{}^{B}\vec{v}_{\text{link}} &= \text{Rot}\Big({}^{\text{joint}}\hat{z}_{\text{joint}}, \theta_z\Big)\,{}^{\text{joint}}\vec{v}_{\text{link}}\\
&= \text{Rot}\Big((0,0,1)^T, \theta_z\Big)\,{}^{\text{joint}}\hat{v}_{\text{link}} \qquad \Rightarrow\; {}^{B}\vec{v}_{\text{link}} = {}^{\text{joint}}\vec{v}_{\text{link}}. && (4.21)
\end{aligned}
$$

When a joint presents a rotation order Y, X, Z relative to the new axes, i.e.

$$
\begin{aligned}
{}^{A}\text{R}_C &= \text{Rot}\Big({}^{A}\hat{y}_A, \theta_y\Big)\,\text{Rot}\Big({}^{B}\hat{x}_B, \theta_x\Big)\,\text{Rot}\Big({}^{C}\hat{z}_C, \theta_z\Big) && (4.22)\\
&= \text{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\text{Rot}\Big((1,0,0)^T, \theta_x\Big)\,\text{Rot}\Big((0,0,1)^T, \theta_z\Big), && (4.23)
\end{aligned}
$$

– as is the case of the shoulder joint – then

$$
\begin{aligned}
{}^{\text{joint}}\hat{v}_{\text{next link}} &= {}^{\text{joint}}\text{R}_{\text{pre rot z}}\,{}^{\text{pre rot z}}\text{R}_{\text{next}}\,{}^{\text{next}}\hat{v}_{\text{next link}}\\
&= {}^{\text{joint}}\text{R}_{\text{pre rot z}}\,\text{Rot}\Big({}^{\text{pre rot z}}\hat{z}_{\text{pre rot z}}, \theta_z\Big)\,{}^{\text{next}}\hat{v}_{\text{next link}} && (4.24)\\
&= {}^{\text{joint}}\text{R}_{\text{pre rot z}}\,{}^{\text{next}}\hat{v}_{\text{next link}}\\
&= \text{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\text{Rot}\Big((1,0,0)^T, \theta_x\Big)\,{}^{\text{next}}\hat{v}_{\text{next link}} && (4.25)
\end{aligned}
$$

where pre rot z is the axes before the rotation around the z axis is performed. Based on (4.18), it is possible to determine that

$$
{}^{\text{joint}}\hat{v}_{\text{next link}} = \begin{pmatrix} \cos(\theta_x)\sin(\theta_y) \\ -\sin(\theta_x) \\ \cos(\theta_x)\cos(\theta_y) \end{pmatrix} \tag{4.26}
$$

(see section A.1 for prof). As such, the angles for the rotation around the x and y axes can be determined as follows

$$
\begin{aligned}
\theta_x &= \arcsin(-v_{\text{next link}2}), && \forall\theta_x \in [-\pi/2, \pi/2[&& (4.27)\\
\theta_y &= \text{arctan2}(v_{\text{next link}1}, v_{\text{next link}3}), && \forall\theta_x \in]-\pi/2, \pi/2[&& (4.28)
\end{aligned}
$$

(c.f. subsubsection 4.4.3.1 for limitations), where $\arctan2(y, x)$ is the four quadrant inverse tangent. Once these angles are known, it is possible to apply the inverse rotations, leaving only the rotation around the z axis, for which more data (another point or vector) is needed

$$\begin{aligned}
{}^{\text{pre rot z}}P &= {}^{\text{pre rot z}}\mathrm{R}_{\text{next}}\,{}^{\text{next}}P && (4.29)\\
&= {}^{\text{pre rot z}}\mathrm{R}_{\text{joint}}\,{}^{\text{joint}}\mathrm{R}_{\text{next}}\,{}^{\text{next}}P \\
&= \mathrm{Rot}\Big((1,0,0)^T, -\theta_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\theta_y\Big)\,{}^{\text{joint}}\mathrm{R}_{\text{next}}\,{}^{\text{next}}P && (4.30)\\
&= \mathrm{Rot}\Big({}^{\text{pre rot z}}\hat{z}_{\text{pre rot z}}, \theta_z\Big)\,{}^{\text{next}}P && (4.31)
\end{aligned}$$

meaning that

$${}^{\text{pre rot z}}P = \mathrm{Rot}\Big((1,0,0)^T, -\theta_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\theta_y\Big)\,{}^{\text{joint}}P. \qquad (4.32)$$

If the next joint, after next, presents only one DoF then, given the way in which the reference axes were defined,

$$\begin{aligned}
{}^{\text{next}}P &= {}^{\text{next}}\mathrm{R}_{\text{after next}}\,{}^{\text{after next}}P && (4.33)\\
\Leftrightarrow\ {}^{\text{pre rot z}}P &= {}^{\text{pre rot z}}\mathrm{R}_{\text{next}}\,{}^{\text{next}}\mathrm{R}_{\text{after next}}\,{}^{\text{after next}}P \\
&= \mathrm{Rot}\Big({}^{\text{pre rot z}}\hat{z}_{\text{pre rot z}}, \theta_z\Big)\,\mathrm{Rot}\Big({}^{\text{next}}\hat{y}_{\text{next}}, \theta_s\Big)\,{}^{\text{after next}}P. && (4.34)
\end{aligned}$$

This means that

$${}^{\text{pre rot z}}\hat{v}_{\text{after next l}} = \mathrm{Rot}\Big({}^{\text{pre rot z}}\hat{z}_{\text{pre rot z}}, \theta_z\Big)\,\mathrm{Rot}\Big({}^{\text{next}}\hat{y}_{\text{next}}, \theta_s\Big)\,{}^{\text{after next}}\hat{v}_{\text{after next l}} \qquad (4.35)$$

$$= \begin{pmatrix} \cos(\theta_z)\sin(\theta_s) \\ \sin(\theta_z)\sin(\theta_s) \\ \cos(\theta_s) \end{pmatrix} \qquad (4.36)$$

where after next l is the link that starts in next and ends in after next (see section A.2 for prof). As such

$$\begin{aligned}
\theta_z &= \arctan2(v_{\text{after next l2}}, v_{\text{after next l1}}), && \forall\theta_s \in\,]0,\pi[&& (4.37)\\
\theta_s &= \arccos(v_{\text{after next l3}}), && \forall\theta_s \in [0,\pi[&& (4.38)
\end{aligned}$$

(c.f. subsubsection 4.4.3.1 for limitations).

When the rotation order is Z, Y, X relative to the new axes – as is the case of the wrist joint – the system is characterized by

$$\begin{aligned}
{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} &= {}^{\mathrm{joint}}\mathrm{R}_{\mathrm{pos\,rot\,z}}\,{}^{\mathrm{pos\,rot\,z}}\mathrm{R}_{\mathrm{next}}\,{}^{\mathrm{next}}\hat{v}_{\mathrm{link}} \\
&= \mathrm{Rot}\Big({}^{\mathrm{joint}}\hat{z}_{\mathrm{joint}}, \theta_z\Big)\,{}^{\mathrm{pos\,rot\,z}}\mathrm{R}_{\mathrm{next}}\,{}^{\mathrm{next}}\hat{v}_{\mathrm{link}} && (4.39) \\
\Leftrightarrow \quad {}^{\mathrm{next}}\hat{v}_{\mathrm{link}} &= {}^{\mathrm{next}}\mathrm{R}_{\mathrm{pos\,rot\,z}}\,\mathrm{Rot}\Big({}^{\mathrm{pos\,rot\,z}}\hat{z}_{\mathrm{pos\,rot\,z}}, -\theta_z\Big)\,{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} \\
&= {}^{\mathrm{next}}\mathrm{R}_{\mathrm{pos\,rot\,z}}\,{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} && (4.40) \\
&= \mathrm{Rot}\Big((1,0,0)^T, -\theta_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\theta_y\Big)\,{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}}, && (4.41)
\end{aligned}$$

where $\mathrm{pos\,rot\,z}$ is the axes obtained after the rotation around the z axis of joint is performed. Using (4.18) (as in the case of the Y, X, Z rotation order),

$$ {}^{\mathrm{next}}\hat{v}_{\mathrm{link}} = \begin{pmatrix} -\sin(\theta_y) \\ \sin(\theta_x)\cos(\theta_y) \\ \cos(\theta_x)\cos(\theta_y) \end{pmatrix} \tag{4.42}$$

(see section A.3 for prof). Meaning that the angles for the x and y axis can be determined based on

$$\begin{aligned}
\theta_x &= \arctan2(v_{\mathrm{link}2}, v_{\mathrm{link}3}), && \forall\theta_x \in [-\pi/2, \pi/2[&& (4.43) \\
\theta_y &= \arcsin(-v_{\mathrm{link}1}), && \forall\theta_x \in]-\pi/2, \pi/2[&& (4.44)
\end{aligned}$$

(c.f. subsubsection 4.4.3.1 for limitations). Applying the inverse rotations for the calculated angles using

$$\begin{aligned}
{}^{\mathrm{pos\,rot\,z}}P &= {}^{\mathrm{pos\,rot\,z}}\mathrm{R}_{\mathrm{joint}}\,{}^{\mathrm{joint}}P \\
&= {}^{\mathrm{pos\,rot\,z}}\mathrm{R}_{\mathrm{next}}\,{}^{\mathrm{next}}\mathrm{R}_{\mathrm{joint}}\,{}^{\mathrm{joint}}P \\
&= \mathrm{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\mathrm{Rot}\Big((1,0,0)^T, \theta_x\Big)\,{}^{\mathrm{next}}\mathrm{R}_{\mathrm{joint}}\,{}^{\mathrm{joint}}P && (4.45) \\
&= \mathrm{Rot}\Big({}^{\mathrm{pos\,rot\,z}}\hat{z}_{\mathrm{pos\,rot\,z}}, -\theta_z\Big)\,{}^{\mathrm{joint}}P, && (4.46)
\end{aligned}$$

so that

$$ {}^{\mathrm{pos\,rot\,z}}P = \mathrm{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\mathrm{Rot}\Big((1,0,0)^T, \theta_x\Big)\,{}^{\mathrm{next}}P, \tag{4.47}$$

which means that the z axis of $\mathrm{pos\,rot\,z}$ and joint are already aligned, leaves only the angle that aligns the x or y axis of the two axes (given that it is only the angle of rotation around the z axis) to be calculated. If the previous joint, prev, possesses only one DoF which, given the way in which the reference axes were defined, this corresponds to a rotation around the y axis of prev. This axis can be calculated based on ${}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{prev\,link}} \times {}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{link}}$ as

$$ {}^{\mathrm{pos\,rot\,z}}\hat{y}_{\mathrm{prev\,link}} = \frac{{}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{prev\,link}} \times {}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{link}}}{\| {}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{prev\,link}} \times {}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{link}} \|} \tag{4.48} $$

(c.f. subsubsection 4.4.3.1 for limitations). It also means that

$$ {}^{A}\hat{y}_{\mathrm{prev\,link}} = {}^{A}\hat{y}_{\mathrm{link}}. \tag{4.49} $$

From this it is now possible to calculate θ_z as follows

$$ {}^{\mathrm{pos\,rot\,z}}\hat{y}_{\mathrm{link}} = {}^{\mathrm{pos\,rot\,z}}\mathrm{R}_{\mathrm{joint}} \, {}^{\mathrm{joint}}\hat{y}_{\mathrm{link}} \tag{4.50} $$

$$ = \mathrm{Rot}\left({}^{\mathrm{pos\,rot\,z}}\hat{z}_{\mathrm{pos\,rot\,z}}, -\theta_z \right) {}^{\mathrm{joint}}\hat{y}_{\mathrm{prev\,link}} $$

$$ = \begin{pmatrix} \sin(\theta_z) \\ \cos(\theta_z) \\ 0 \end{pmatrix} \tag{4.51} $$

$$ \therefore \theta_z = \mathrm{arctan2}(y_1, y_2), \tag{4.52} $$

where y_1 and y_2 are the 1st and 2nd elements (x and y coordinates), respectively, of ${}^{\mathrm{pos\,rot\,z}}\hat{y}_{\mathrm{link}}$.

As can be seen in (4.26) and (4.42) – the calculations required for obtaining the first angles for each order, Y, X, Z and Z, Y, X respectively – in order to calculate the joint angles, the coordinates of the points need to be calculated in relation to two axes – joint for the Y, X, Z order and next for the Z, Y, X order – meaning that the first step is to calculate the coordinates in respect to these two axes. To do this, two steps are necessary:

1. align the z axis of the desired joint, $\mathrm{desired}$, with that of the current axes, $\mathrm{cur\,axes}$ obtaining the axes $\mathrm{z\,aligned}$, for which the z axis of $\mathrm{desired}$ and $\mathrm{z\,aligned}$ are the

same, meaning that only a rotation around the z axis is required to transform $\mathrm{z\,aligned}$ into $\mathrm{desired}$;

2. align the remaining axis by aligning either the x or y axis of $\mathrm{desired}$ with the corresponding axis of $\mathrm{z\,aligned}$.

To perform item 1, since both the original axis, z axis of $\mathrm{z\,aligned}$, and the desired axis, z axis of $\mathrm{desired}$ are known, we can use their cross and dot products

$$(\vec{u} \times \vec{v}) \perp \vec{u} \wedge (\vec{u} \times \vec{v}) \perp \vec{v} \tag{4.53}$$

$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos(\theta) \tag{4.54}$$

to calculate a normal vector to their plane of rotation – a possible rotation axis, $\hat{n}$ – as well as the angle of rotation around this axis, θ, defined as

$$\hat{n} = \frac{\vec{u} \times \vec{v}}{\|\vec{u} \times \vec{v}\|} \tag{4.55}$$

$$\theta = \arccos\left(\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|}\right) \tag{4.56}$$

(c.f. subsubsection 4.4.3.1 for the limitations of this formula). It is now possible to determine a rotation matrix that aligns $\vec{u}$ with $\vec{v}$, from $\hat{n}$ and θ, using Rodrigues' rotation formula

$$\mathrm{Rot}(\hat{n}, \theta) = \mathrm{I} + s\tilde{n} + (1 - c)\tilde{n}^2 \tag{4.57}$$

$$= \begin{pmatrix} c + n_1^2(1-c) & n_1 n_2(1-c) - n_3 s & n_1 n_3(1-c) + n_2 s \\ n_1 n_2(1-c) + n_3 s & c + n_2^2(1-c) & n_2 n_3(1-c) - u_1 s \\ n_1 n_3(1-c) - n_2 s & n_2 n_3(1-c) + n_1 s & c + n_3^2(1-c) \end{pmatrix}, \tag{4.58}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$ and $\tilde{n}$ is the antisymmetric matrix of $\hat{n}$ (4.55), defined as

$$\tilde{n} = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}. \tag{4.59}$$

As such and given (4.17), by replacing $\vec{u}$ with ${}^{\mathrm{cur\,axes}}\hat{v}_{\mathrm{desired\,link}}$ and $\vec{v}$ with ${}^{\mathrm{cur\,axes}}\hat{z}_{\mathrm{cur\,axes}}$ in (4.55) and (4.56), $\mathrm{Rot}(\hat{n}_{\mathrm{z\,aligned}}, \theta_{\mathrm{z\,aligned}})$, where $\hat{n}_{\mathrm{z\,aligned}}$ and $\theta_{\mathrm{z\,aligned}}$ correspond to $\hat{n}$ and θ after substituting $\vec{u}$ and $\vec{v}$, becomes a rotation matrix that transforms $\mathrm{cur\,axes}$ into $\mathrm{z\,aligned}$, being that

$$^{\mathrm{z\,aligned}}\hat{v}_{\mathrm{desired\,link}} = \mathrm{Rot}\big(\hat{n}_{\mathrm{z\,aligned}}, \theta_{\mathrm{z\,aligned}}\big)\ ^{\mathrm{cur\,axes}}\hat{v}_{\mathrm{desired\,link}} \tag{4.60}$$

$$^{\mathrm{z\,aligned}}P = \mathrm{Rot}\big(\hat{n}_{\mathrm{z\,aligned}}, \theta_{\mathrm{z\,aligned}}\big)\ ^{\mathrm{cur\,axes}}P, \tag{4.61}$$

In order to finish aligning the $\mathrm{desired}$ axes and the reference axes, as stated in item 2, the current reference axes, $\mathrm{z\,aligned}$, must be rotated so that the x and y axis of both the reference axes and the shoulder axes align. Let $^{\mathrm{z\,aligned}}\hat{a}_{\mathrm{desired}}$ be the unit vector of either the x or y axis of the $\mathrm{desired}$ axes whose is known relative to the $\mathrm{z\,aligned}$ axes and $^{\mathrm{z\,aligned}}\hat{a}_{\mathrm{z\,aligned}}$ the corresponding axes of $\mathrm{z\,aligned}$. Than, by replacing $\vec{u}$ and $\vec{v}$ in (4.55) and (4.56) by $^{\mathrm{z\,aligned}}\hat{a}_{\mathrm{desired}}$ and $^{\mathrm{z\,aligned}}\hat{a}_{\mathrm{z\,aligned}}$, respectively, $\mathrm{Rot}(\hat{n}_{\mathrm{desired}}, \theta_{\mathrm{desired}})$ now becomes the rotation matrix that transforms the $\mathrm{z\,aligned}$ into the $\mathrm{desired}$ axes

$$^{\mathrm{desired}}\hat{a}_{\mathrm{desired}} = \mathrm{Rot}(\hat{n}_{\mathrm{desired}}, \theta_{\mathrm{desired}})\ ^{\mathrm{z\,aligned}}\hat{a}_{\mathrm{desired}} \tag{4.62}$$

$$^{\mathrm{desired}}P = \mathrm{Rot}(\hat{n}_{\mathrm{desired}}, \theta_{\mathrm{desired}})\ ^{\mathrm{z\,aligned}}P. \tag{4.63}$$

4.4.3.1 Correcting the issues

The solution presented in subsection 4.4.3 has a few issues that will be seen next. Some of these issues may cause problems when calculating the joint angles for the arm and must therefore be addressed. The issues that do not affect the calculations of the arm angles will still be explained, although not corrected, given that the same formulas that present said issues may be required for calculating other angles.

One issue with the solution presented in subsection 4.4.3 is the duality of the axis of rotation given by (4.55), which can be derived from

$$\vec{u} \times \vec{v} = \|u\|\|v\|\sin(\theta)\hat{n} \tag{4.64}$$

where $\hat{n}$ is the unit vector perpendicular to $\vec{u}$ and $\vec{v}$ for which the angle of rotation around $\hat{n}$ that aligns $\vec{u}$ with $\vec{v}$, θ, is such that $\theta \in \,]0, \pi[$ – for $\theta = 0 \wedge \theta = \pi$ the cross product is null and any unit vector perpendicular to $\vec{u}$ can serve as the axis of rotation, such as the normalized vector $\left(u_2/\sqrt{u_2^2+u_1^2}, -u_1/\sqrt{u_2^2+u_1^2}, 0\right)^T$. This angle is the same as the one calculated in (4.56). When the angle of rotation, θ, for a given $\hat{n}$, would be a value in the interval $]-\pi, 0[$, the axis becomes $-\hat{n}$ and the accompanying angle turns into $2\pi - \theta$. Although this does not cause an error when using the rotation matrix defined as $\mathrm{Rot}(\hat{n}, \theta)$ in (4.58), as both axis and angle are used, it may however cause an error when only the axis is used, as in the case of (4.48) and (4.90) (c.f. subsubsection 4.4.3.2), or when only the angle is used. Given that this error only occurs when

$$\angle(\vec{u}, \vec{v}) \notin \,]0, \pi[\,, \tag{4.65}$$

that all joints with a single DoF, for the reference system used, all have an angle of rotation in the interval $]0, \pi[$ and, since when no rotation is applied to the $\mathrm{shoulder}$ link the angle corresponds to $\pi/2$ radians, the $\mathrm{shoulder}$ link can not rotate enough so that (4.65) becomes true, as it would need to rotate at least $\pi/2$ radians so that (4.90) would be incorrect, this issue will not affect the calculations.

Another issue are the singularities of (4.26), (4.36) and (4.42) that occur when $\theta_x = \pm\pi/2$, $\theta_s = 0 \vee \theta_s = \pi$ and $\theta_y = \pm\pi/2$, respectively. In all these situations we end up with $\arctan2(0, 0)$ and the values of θ_y, θ_z and θ_x, for $\theta_x = \pm\pi/2$, $\theta_s = 0 \vee \theta_s = \pi$ and $\theta_y = \pm\pi/2$ respectively, have to be calculated based on previous information – either maintaining the previous value or estimating the angle based on its variation, if it was the first value then there is no way of knowing until a different value appears.

The final issue occurs when $\cos(\theta_x) < 0$, for the angles determined from (4.26), $\sin(\theta_s) < 0$, for angles determined from (4.36), and $\cos(\theta_y) < 0$, for angles determined from (4.42). Given that (4.26) and (4.42) are very similar – only changing θ_x for θ_y and the order of the terms of the vector – the limitations for θ_x in (4.26) are the same as the ones for θ_y in (4.42) and, therefore, only (4.26) will be analysed. Let

$$a = |\cos(\theta_x)| \tag{4.66}$$

$$b = |\sin(\theta_s)| \tag{4.67}$$

than, for $\cos(\theta_x) < 0$ and $\sin(\theta_s) < 0$,

$$\begin{aligned}\theta_y &= \arctan2(v_{\text{next link1}}, v_{\text{next link3}}) \\ &= \arctan2(-a\sin(\theta_y), -a\cos(\theta_y)) \\ &= \arctan2(a\sin(\theta_y+\pi), a\cos(\theta_y+\pi)) \\ &= \arctan2(\sin(\theta_y+\pi), \cos(\theta_y+\pi)) \\ &= \theta_y + \pi \qquad (4.68)\end{aligned}$$

$$\begin{aligned}\theta_z &= \arctan2(v_{\text{after next l2}}, v_{\text{after next l1}}) \\ &= \arctan2(-b\sin(\theta_z), -b\cos(\theta_z)) \\ &= \arctan2(b\sin(\theta_z+\pi), b\cos(\theta_z+\pi)) \\ &= \arctan2(\sin(\theta_z+\pi), \cos(\theta_z+\pi)) \\ &= \theta_z + \pi \qquad (4.69)\end{aligned}$$

$$\therefore \cos(\theta_x) > 0 \qquad (4.70)$$

$$\Rightarrow\ \theta_x \in \left]-\pi/2, \pi/2\right[\qquad (4.71)$$

$$\therefore \sin(\theta_s) > 0 \qquad (4.72)$$

$$\Rightarrow\ \theta_s \in \left]0, \pi\right[\,. \qquad (4.73)$$

Since θ_s in (4.36) is only applied to joints with a single DoF, which all present an angle of rotation in the the interval $]0,\pi[$, (4.73) is always uphold. Therefore, no alterations to (4.36) are needed. Given that, in this work, these formulas will only be used to calculate the arm angles and that only the wrist presents a rotation order of type Z, Y, X, the rotation order for which (4.36) is needed, no alterations to (4.42) are required given that the angles of rotation around the x and y axis for the wrist are both within the limit allowed by (4.71). In the case of (4.26), which applies to the shoulder joint, (4.71) is not true given that the rotations around both the x and y axis have a range of motion greater than π radians. This means that a more complex system must be employed to determine the correct rotations, rather than just using (4.27) and (4.28). To that end (4.27) can be altered to

$$\theta_x = \arctan2\left(-v_{\text{next link2}}, \frac{v_{\text{next link1}}}{\sin(\theta_y)}\right). \qquad (4.74)$$

Because the restriction on θ_x was removed, the system can now return two different angle combinations – the desired one, presented in (4.74) and (4.28), and an erroneous solution – given that the same vector can be obtained through two distinct rotations

$$
\begin{aligned}
{}^{\text{joint}}\hat{v}_{\text{next link}} &= \begin{pmatrix} \cos(\theta_x)\sin(\theta_y) \\ -\sin(\theta_x) \\ \cos(\theta_x)\cos(\theta_y) \end{pmatrix} \\
&= \begin{pmatrix} (-\cos(\theta_x))(-\sin(\theta_y)) \\ -\sin(\theta_x) \\ (-\cos(\theta_x))(-\cos(\theta_y)) \end{pmatrix} \\
&= \begin{pmatrix} \cos(\pi-\theta_x)\sin(\theta_y+\pi) \\ -\sin(\pi-\theta_x) \\ \cos(\pi-\theta_x)\cos(\theta_y+\pi) \end{pmatrix}. \qquad (4.75)
\end{aligned}
$$

As seen in (4.68), if $cos(\theta_x) < 0$ the value obtained from (4.28) will be the incorrect value of $\theta_y + \pi$ meaning that (4.74) will give the value of $\pi - \theta_x$ since the second rotation was considered as the correct one

$$
\begin{aligned}
\theta_x &= \operatorname{arctan2}\left(-v_{\text{next link2}}, \frac{v_{\text{next link1}}}{\sin(\theta_y+\pi)}\right) \qquad (4.76) \\
&= \operatorname{arctan2}\left(\sin(\theta_x), \frac{\cos(\theta_x)\sin(\theta_y)}{-\sin(\theta_y)}\right) \\
&= \operatorname{arctan2}(\sin(\theta_x), -\cos(\theta_x)) \\
&= \operatorname{arctan2}(\sin(\pi-\theta_x), \cos(\pi-\theta_x)) \\
&= \pi - \theta_x, \qquad (4.77)
\end{aligned}
$$

the same value would have been obtained even if $v_{\text{next link3}}/\cos(\theta_y+\pi)$ had been used instead of $v_{\text{next link1}}/\sin(\theta_y+\pi)$ (although the use of one or the other can have an impact on the precision of the calculation due to floating point error when $\sin(\theta_y) \approx 0$ or $\cos(\theta_y) \approx 0$). Because the calculation of the pos rot z axes depends on θ_x and θ_y (see (4.32)), the value of θ_z will also be affected as the axis obtained will be rotated an additional π radians

$$
\begin{aligned}
{}^{\text{pre rot z}}P &= \operatorname{Rot}\Big((1,0,0)^T, -(\pi-\theta_x)\Big)\operatorname{Rot}\Big((0,1,0)^T, -(\theta_y+\pi)\Big)\,{}^{\text{joint}}\mathrm{R}_{\text{next}}\,{}^{\text{next}}P \\
&= \operatorname{Rot}\Big({}^{\text{pre rot z}}\hat{z}_{\text{pre rot z}}, \theta_z+\pi\Big)\,{}^{\text{next}}P \qquad (4.78)
\end{aligned}
$$

(see section A.4 for prof), meaning that (4.37) will give the value $\theta_z + \pi$. If the values given by (4.74), (4.28) and (4.37) are within the angular limits of the joint then they are correct. If not, the alternative value is selected

$$\alpha_y = \arctan2(v_{\text{next link1}}, v_{\text{next link3}}) \tag{4.79}$$

$$\alpha_x = \arctan2\left(-v_{\text{next link2}}, \frac{v_{\text{next link1}}}{\sin(\alpha_y)}\right) \tag{4.80}$$

$$\alpha_z = \arctan2(v_{\text{after next l2}}, v_{\text{after next l1}}) \tag{4.81}$$

$$\theta_y = \begin{cases} \alpha_y & \text{if } \alpha_x \in S_{\text{joint}_x} \wedge \alpha_y \in S_{\text{joint}_y} \wedge \alpha_z \in S_{\text{joint}_z} \\ \alpha_y - \pi & \text{if otherwise} \end{cases} \tag{4.82}$$

$$\theta_x = \begin{cases} \alpha_x & \text{if } \alpha_x \in S_{\text{joint}_x} \wedge \alpha_y \in S_{\text{joint}_y} \wedge \alpha_z \in S_{\text{joint}_z} \\ \pi - \alpha_x & \text{if otherwise} \end{cases} \tag{4.83}$$

$$\theta_z = \begin{cases} \alpha_z & \text{if } \alpha_x \in S_{\text{joint}_x} \wedge \alpha_y \in S_{\text{joint}_y} \wedge \alpha_z \in S_{\text{joint}_z} \\ \alpha_z - \pi & \text{if otherwise} \end{cases}, \tag{4.84}$$

where S_{joint_x}, S_{joint_y} and S_{joint_z} are the sets of all possible values for rotation angles around the x, y and z axis respectively.

4.4.3.2 Application to the arm

Given the desired angles – shoulder, elbow, and wrist angles – it is possible to start with either the shoulder or wrist angles. Starting with the shoulder angles, the first step, as seen in subsection 4.4.3, is to align the shoulder axes and the reference axes, camera. Using (4.55), (4.56) and (4.61)

$$\hat{n}_{\text{z aligned}} = \frac{{}^{\text{camera}}\vec{v}_{\text{shoulder}} \times {}^{\text{camera}}\hat{z}_{\text{camera}}}{\|{}^{\text{camera}}\vec{v}_{\text{shoulder}} \times {}^{\text{camera}}\hat{z}_{\text{camera}}\|} \tag{4.85}$$

$$\theta_{\text{z aligned}} = \arccos\left(\frac{{}^{\text{camera}}\vec{v}_{\text{shoulder}} \cdot {}^{\text{camera}}\hat{z}_{\text{camera}}}{\|{}^{\text{camera}}\vec{v}_{\text{shoulder}}\|\|{}^{\text{camera}}\hat{z}_{\text{camera}}\|}\right) \tag{4.86}$$

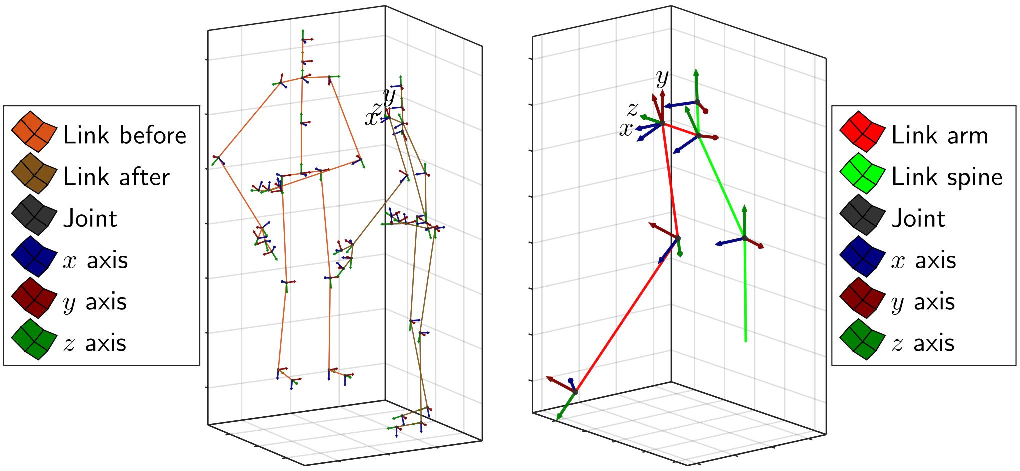
$${}^{\text{z aligned}}P = \text{Rot}\big(\hat{n}_{\text{z aligned}}, \theta_{\text{z aligned}}\big)\, {}^{\text{camera}}P, \tag{4.87}$$

to align the the z axis we obtain something similar to Figure 4.15. To obtain those results, it is necessary to also apply a translation to the joint positions

$$ {}^{\mathrm{z\,aligned}}\mathrm{T}_{\mathrm{z\,aligned}} = \mathrm{Rot}_{\mathrm{h}}\left(\hat{n}_{\mathrm{z\,aligned}}, \theta_{\mathrm{z\,aligned}}\right) \mathrm{Trans}_{\mathrm{h}}(-\,{}^{\mathrm{camera}}P_{\mathrm{shoulder}}) \tag{4.88} $$

$$ \begin{pmatrix} {}^{\mathrm{z\,aligned}}P \\ 1 \end{pmatrix} = {}^{\mathrm{z\,aligned}}\mathrm{T}_{\mathrm{z\,aligned}} \begin{pmatrix} {}^{\mathrm{camera}}P \\ 1 \end{pmatrix}. \tag{4.89} $$

Given that the calculations to determine the joint angles are based on the vectors of the links and,therefore, not on the joint positions but on their difference, the translation is unnecessary in order to obtain the desired data. As such, although the figures presented here will have the translation applied to the joints, this will be merely for presentation purposes in order to make the visualization of the alignment of the various axes simpler. Being that the case, no further translation will be described (although the process is identical to the one performed in (4.88) and (4.89)). The next step is performing the alignment of the x and y axis of the shoulder with the corresponding axis in the



(a) Position and orientation of all joints before and after transformation

(b) Position and orientation of the arm and spine joints after transformation

Figure 4.15: Transformation of the joint positions in order to align the z axis of the shoulder with that of the reference axes. (The axis with a label on the figures, in this and in the following figures, correspond to axis of the reference axes for the joint positions)

$\mathrm{z\,aligned}$ using (4.63). To do this either ${}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}}$ or ${}^{\mathrm{z\,aligned}}\hat{y}_{\mathrm{shoulder}}$ must be known. Since, for the reference axes chosen, ${}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}}$ corresponds to the axis of rotation between ${}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{upper\,spine}}$ and ${}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{shoulder}}$ then

$$ {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}} = \frac{{}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{upper\,spine}} \times {}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{shoulder}}}{\| {}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{upper\,spine}} \times {}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{shoulder}} \|}. \tag{4.90} $$

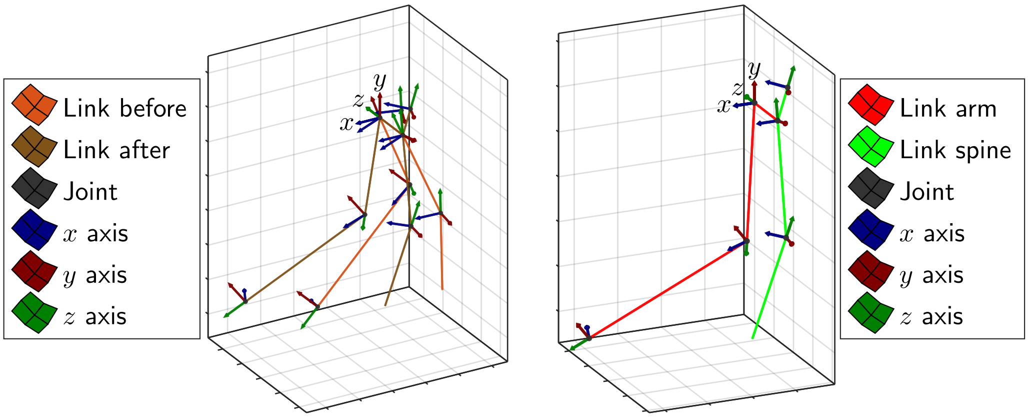
Replacing in (4.55), (4.56) and (4.63)

$$ \hat{n}_{\mathrm{z\,aligned}} = \frac{{}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}} \times {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{z\,aligned}}}{\| {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}} \times {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{z\,aligned}} \|} \tag{4.91} $$

$$ \theta_{\mathrm{z\,aligned}} = \arccos\left(\frac{{}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}} \cdot {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{z\,aligned}}}{\| {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{shoulder}} \| \| {}^{\mathrm{z\,aligned}}\hat{x}_{\mathrm{z\,aligned}} \|} \right) \tag{4.92} $$

$$ {}^{\mathrm{shoulder}}P = \mathrm{Rot}(\hat{n}_{\mathrm{shoulder}}, \theta_{\mathrm{shoulder}})\, {}^{\mathrm{z\,aligned}}P, \tag{4.93} $$

the $\mathrm{shoulder}$ becomes the current axes (see Figure 4.16), as required for the calculation of the $\mathrm{shoulder}$ and elbow angles.



(a) Position and orientation of the arm and spine joints before and after transformation

(b) Position and orientation of the arm and spine joints after transformation

Figure 4.16: Transformation of the joint positions in order to align the x axis of the $\mathrm{shoulder}$ axes with the corresponding axis of $\mathrm{z\,aligned}$

Using (4.80) and (4.79) to calculate the temporary values for θ_x and θ_y

$$ \alpha_y = \operatorname{arctan2}(v_{\mathrm{upper\,arm}\,1}, v_{\mathrm{upper\,arm}\,3}) \tag{4.94} $$

$$ \alpha_x = \operatorname{arctan2}\left(-v_{\mathrm{upper\,arm}\,2}, \frac{v_{\mathrm{upper\,arm}\,1}}{\sin(\alpha_y)} \right), \tag{4.95} $$

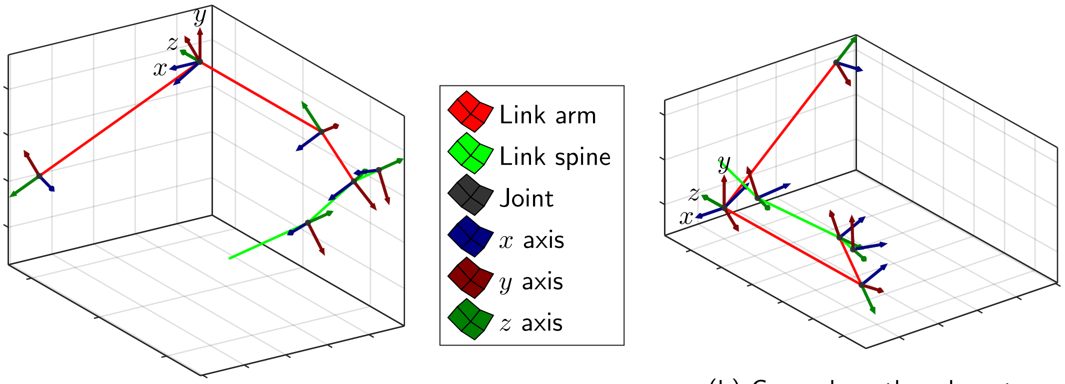
where $v_{\mathrm{upper\,arm}1}$, $v_{\mathrm{upper\,arm}2}$ and $v_{\mathrm{upper\,arm}3}$ are the 1st, 2nd and 3rd elements, respectively, of ${}^{\mathrm{shoulder}}\vec{v}_{\mathrm{upper\,arm}}$. Removing the calculated rotations, as in (4.32), we obtain

$$ {}^{\mathrm{pre\,rot\,z}}P = \mathrm{Rot}\Big((1,0,0)^T, -\alpha_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\alpha_y\Big)\,{}^{\mathrm{shoulder}}P, \tag{4.96} $$

meaning that

$$ {}^{\mathrm{pre\,rot\,z}}P = \mathrm{Rot}\Big((0,0,1)^T, \theta_z\Big)\,{}^{\mathrm{elbow}}P \;\vee\; {}^{\mathrm{pre\,rot\,z}}P = \mathrm{Rot}\Big((0,0,1)^T, \theta_z + \pi\Big)\,{}^{\mathrm{elbow}}P \tag{4.97} $$

(see Figure 4.17). It is now possible to calculate the remaining temporary angle for the shoulder and the angle of the elbow using (4.81) and (4.38), respectively,



(a) Case where α_x and α_y correspond to θ_x and θ_y

(b) Case where they do not correspond

Figure 4.17: Remove rotations around x axis and y axis of the shoulder

$$ \alpha_z = \arctan2(v_{\text{fore-arm}2}, v_{\text{fore-arm}1}) \tag{4.98} $$

$$ \theta_s = \arccos(v_{\text{fore-arm}3}). \tag{4.99} $$

Since all the temporary values for the $\mathrm{shoulder}$ angles are known, it is now possible to apply (4.83), (4.82) and (4.84) to determine the correct angles

$$\theta_y = \begin{cases} \alpha_y & \text{if } \alpha_x \in S_{\mathrm{shoulder}x} \wedge \alpha_y \in S_{\mathrm{shoulder}y} \wedge \alpha_z \in S_{\mathrm{shoulder}z} \\ \alpha_y - \pi & \text{if otherwise} \end{cases} \tag{4.100}$$

$$\theta_x = \begin{cases} \alpha_x & \text{if } \alpha_x \in S_{\mathrm{shoulder}x} \wedge \alpha_y \in S_{\mathrm{shoulder}y} \wedge \alpha_z \in S_{\mathrm{shoulder}z} \\ \pi - \alpha_x & \text{if otherwise} \end{cases} \tag{4.101}$$

$$\theta_z = \begin{cases} \alpha_z & \text{if } \alpha_x \in S_{\mathrm{shoulder}x} \wedge \alpha_y \in S_{\mathrm{shoulder}y} \wedge \alpha_z \in S_{\mathrm{shoulder}z} \\ \alpha_z - \pi & \text{if otherwise} \end{cases} . \tag{4.102}$$

In order to determine the wrist angles, all coordinates must be in relation to the hand axes, as seen in subsection 4.4.3, given that the wrist presents an order of rotation of type Z, Y, X. To that end, the z axis of hand must first be aligned with the current axes, $\mathrm{cur\,axes}$, using (4.55), (4.56) and (4.61)

$$\hat{n}_{\mathrm{z\,aligned}} = \frac{{}^{\mathrm{cur\,axes}}\vec{v}_{\text{fore-arm}} \times {}^{\mathrm{cur\,axes}}\hat{z}_{\mathrm{cur\,axes}}}{\|{}^{\mathrm{cur\,axes}}\vec{v}_{\text{fore-arm}} \times {}^{\mathrm{cur\,axes}}\hat{z}_{\mathrm{cur\,axes}}\|} \tag{4.103}$$

$$\theta_{\mathrm{z\,aligned}} = \arccos\left(\frac{{}^{\mathrm{cur\,axes}}\vec{v}_{\text{fore-arm}} \cdot {}^{\mathrm{cur\,axes}}\hat{z}_{\mathrm{cur\,axes}}}{\|{}^{\mathrm{cur\,axes}}\vec{v}_{\text{fore-arm}}\|\|{}^{\mathrm{cur\,axes}}\hat{z}_{\mathrm{cur\,axes}}\|}\right) \tag{4.104}$$

$${}^{\mathrm{z\,aligned}}P = \mathrm{Rot}\big(\hat{n}_{\mathrm{z\,aligned}}, \theta_{\mathrm{z\,aligned}}\big)\ {}^{\mathrm{cur\,axes}}P \tag{4.105}$$

(see Figure 4.18). For the next step either the x or y axis are need but, given that the hand axes is a joint with two DoFs, neither of these are known. Despite this fact, since the rotation around the x axis is small it will be disregarded and the joint will be treated as one with a single DoF. As such, the y axis of the hand, ${}^{\mathrm{z\,aligned}}\hat{y}_{\mathrm{hand}}$, corresponds to its axis of rotation and

$${}^{\mathrm{z\,aligned}}\hat{y}_{\mathrm{hand}} = \frac{{}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{hand}} \times {}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{index}}}{\|{}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{hand}} \times {}^{\mathrm{z\,aligned}}\vec{v}_{\mathrm{index}}\|}. \tag{4.106}$$

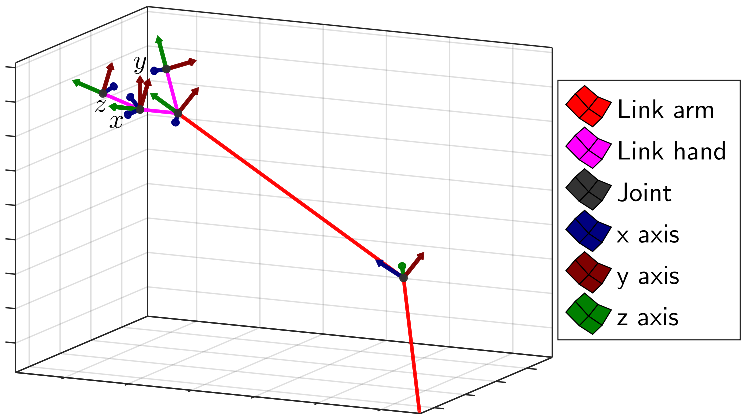


Figure 4.18: Transformation of the joint positions in order to align the z axis of the hand with that of the current axes

Replacing in (4.55), (4.56) and (4.63)

$$\hat{n}_{\text{hand}} = \frac{{}^{\text{z aligned}}\hat{y}_{\text{hand}} \times {}^{\text{z aligned}}\hat{y}_{\text{z aligned}}}{\| {}^{\text{z aligned}}\hat{y}_{\text{hand}} \times {}^{\text{z aligned}}\hat{y}_{\text{z aligned}} \|} \tag{4.107}$$

$$\theta_{\text{hand}} = \arccos\left(\frac{{}^{\text{z aligned}}\hat{y}_{\text{hand}} \cdot {}^{\text{z aligned}}\hat{y}_{\text{z aligned}}}{\| {}^{\text{z aligned}}\hat{y}_{\text{hand}} \| \| {}^{\text{z aligned}}\hat{y}_{\text{z aligned}} \|} \right) \tag{4.108}$$

$${}^{\text{hand}}P = \text{Rot}(\hat{n}_{\text{hand}}, \theta_{\text{hand}})\, {}^{\text{z aligned}}P \tag{4.109}$$

(see Figure 4.19), it is now possible to determine the angles of rotation around the x

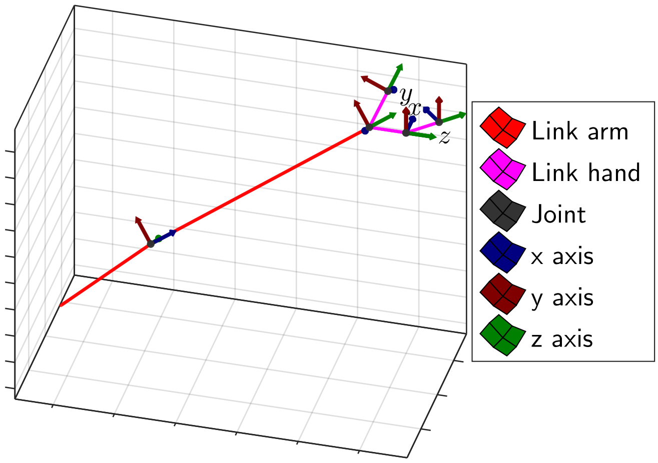


Figure 4.19: Transformation of the joint positions in order to align the y axis of the hand axes with the corresponding axis of $\mathrm{z\,aligned}$

and y axis thanks to (4.43) and (4.44)

$$\theta_x = \operatorname{arctan2}(v_{\text{fore-arm}2}, v_{\text{fore-arm}3}) \tag{4.110}$$

$$\theta_y = \arcsin(-v_{\text{fore-arm}1}), \tag{4.111}$$

where $v_{\text{fore-arm}1}$, $v_{\text{fore-arm}2}$ and $v_{\text{fore-arm}3}$ are the 1st, 2nd and 3rd elements, respectively, of ${}^{\text{hand}}\vec{v}_{\text{fore-arm}}$. Now only the angle of rotation around the z axis of the wrist needs to be calculated. Removing the rotations around the x and y axis, based on (4.47),

$$^{\text{pos rot z}}P = \text{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\text{Rot}\Big((1,0,0)^T, \theta_x\Big)\ {}^{\text{hand}}P \tag{4.112}$$

(see Figure 4.20), it becomes possible to calculate the final angle using (4.52)

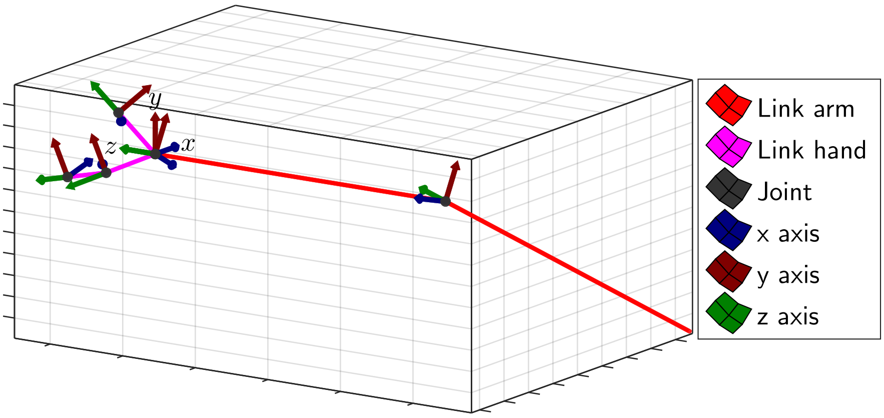


Figure 4.20: Remove rotations around x axis and y axis of the hand

$$\theta_z = \operatorname{arctan2}(y_1, y_2)., \tag{4.113}$$

where y_1 and y_2 are the 1st and 2nd elements, respectively, of ${}^{\text{pos rot z}}\hat{y}_{\text{link}}$.

4.4.4 Experimental results

Applying the solution presented in subsection 4.4.3 to the data acquired from the Kinect and its SDK (see subsection 4.4.2), it was possible to obtain joint movement data similar to that presented in Figure 4.21. As can be seen in Figure 4.21, the data possesses a considerable amount of noise (see curve for wrist x). This noise is due to the imprecisions

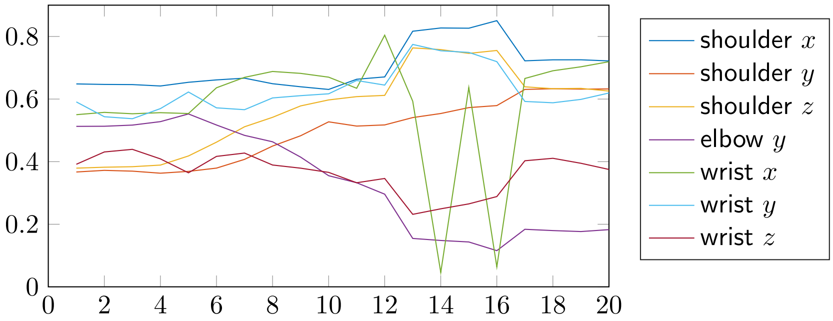


Figure 4.21: Aquired joint data, scaled to a $[0, 1]$ interval according to (B.1)

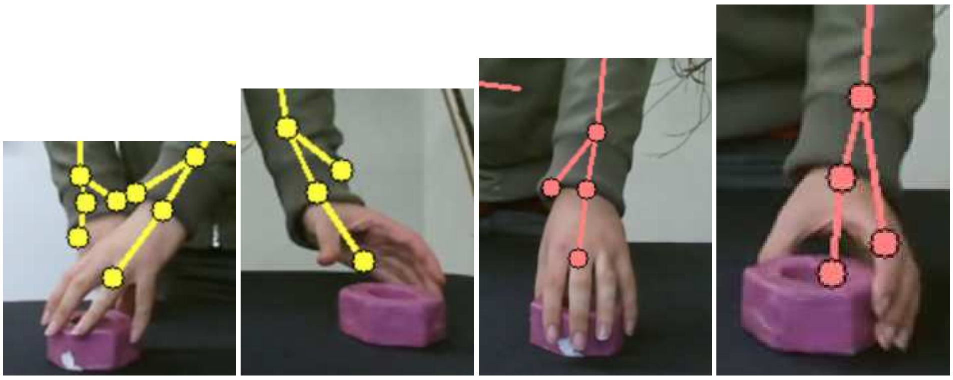


Figure 4.22: Examples of erroneous acquisitions of the positions for the hand and the tip of the index finger

in the acquisition of the position of the hand and tip of the index finger as seen in Figure 4.22, which can cause the system to return erroneous data. Despite this the data used for the following tests was unfiltered, to see if the system itself was capable of handling considerable amounts of noise, given that a DNF already possesses some noise immunity.

In order to test the efficiency of the the SODNF when handling acquired data, the method described in section 4.2 was used to determine which trajectories were grouped together. The parameters used for this evaluation can be seen in Table 4.3, And, in order to test and compare the performance of each parametrization, the data was processed in the following way:

1. Determine the groups formed and trajectories to ignore using the method described

Table 4.3: Analysis parameters

Parameter	Value
$\mathrm{success\,rate}$	0.95
$\mathrm{eval\,window}$	10
p	0.7
$\mathrm{th}_{\mathrm{eval\,window}}$	0.7
Q_{window}	10
$\mathrm{th}_{\mathrm{disp}}$	0.7
$\mathrm{th}_{\mathrm{ignored}}$	0.8

in section 4.2;

2. Based on obtained groups, the trajectories in each group are separated according to the characteristics being analysed
   (a) position (left, centre or right, see Figure 4.10);
   (b) object type (nut or wheel, see Figure 4.9);
   (c) grasp type (above grip or side grip, see Figure 4.11);
   (d) object and grasp types (the Grasp Nut with Above Grip (GNAG), the Grasp Nut with Side Grip (GNSG) or the Grasp Wheel with Above Grip (GWAG), see Figure 4.11);
   (e) or according to the trajectory type, i.e. the combination of the above criteria (the GNAG Left/Centre/Right, the GNSG Left/Centre/Right, the GWAG Left/Centre or Right)
3. Calculate how many trajectories that fit into each criterion are present in each of the groups. Being that, if a group has trajectories from more than one criterion, only those from the criterion with the most elements in the group will be counted as correctly grouped trajectories.
4. After the above steps have been applied to each of the 5 trials (see introduction to chapter 4), the value of

   (a) the median number of correctly grouped trajectories for each criterion (the greater the better);
   (b) the the IQR of the number of correctly grouped trajectories for each criterion (the smaller the better so as to decrease the uncertainty of the results);
   (c) and the number of groups into which the trajectories were divided (the smaller the better to not over separate the trajectories)

   can be calculated. These values can them be represented a percentage relative to the total number of possible trajectories (so as to balance the criteria since each criterion can have a different number of trajectories).

5. Once these percentages are known their mean across the various criteria can be calculated and used as a way of comparing the results for each parametrization.

Using this method, each of the criteria types described in item 2 were analysed and the top 20 parametrizations are displayed in Tables 4.4 to 4.8.

Table 4.4: The 20 best results from the tests with acquired data, ordered according to performance in grouping by position

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
1	$\mathrm{k_{bp}}$	0.8	100.0	8.3	1.3
2	c_T	25	100.0	74.7	1.0
3	C^{som}	20	99.0	19.9	2.3
4	$\tau^{a^{som}}$	600	77.8	9.3	1.0
5	c_T	30	77.8	16.2	1.0
6	$\mathrm{k_{bp}}$	0.95	77.8	22.2	1.0
7	$\tau^{a^{som}}$	900	77.8	64.9	1.0
8	$\tau^{a^{som}}$	750	77.8	65.2	1.0
9	$\mathrm{k_{bp}}$	0.75	77.8	66.7	1.0
10	c_T	15	70.7	97.5	1.0
11	r	-0.1	68.7	10.1	1.3
12	r	-0.2	68.7	37.6	1.7
13	C^{som}	16	67.7	17.2	1.3
14	r	-0.3	67.7	26.3	1.7
15	C^{som}	18	66.7	3.5	1.3
16	σ_w	100	66.7	7.6	1.0
17	$\mathrm{k_{bp}}$	0.45	66.7	8.3	0.7
18	C^{som}	22	66.7	21.5	1.0
19	σ_w	110	66.7	29.3	1.0
20	$\mathrm{k_{bp}}$	0.6	66.7	33.3	0.7

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.5: The 20 best results from the tests with acquired data, ordered according to performance in grouping by object type

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
1	C^{som}	20	60.6	18.0	4.0
2	r	-0.2	53.0	14.4	2.5
3	C^{som}	16	51.5	0.8	2.0
4	σ_w	40	51.5	1.9	1.0
5	r	-0.3	51.5	3.0	2.5
6	$\tau^{a^{som}}$	600	51.5	4.4	2.0
7	C^{som}	18	51.5	4.9	2.0
8	r	-0.1	51.5	6.1	2.5
9	$\tau^{a^{som}}$	300	50.8	1.5	1.0
10	σ_w	30	50.0	0.0	0.5
11	k_{bp}	0	50.0	0.0	0.5
12	k_{bp}	0.15	50.0	0.0	0.5
13	C^{som}	12	50.0	0.0	0.5
14	C^{som}	30	50.0	0.0	0.5
15	C^{som}	38	50.0	0.0	0.5
16	C^{som}	40	50.0	0.0	0.5
17	C^{som}	42	50.0	0.0	0.5
18	C^{som}	44	50.0	0.0	0.5
19	C^{som}	48	50.0	0.0	0.5
20	r	-0.5	50.0	0.0	1.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k}_{\mathrm{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.6: The 20 best results from the tests with acquired data, ordered according to performance in grouping by grasp type

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
1	C^{som}	20	82.6	17.4	3.5
2	c_T	35	66.7	0.0	1.0
3	$\mathrm{k_{bp}}$	0.95	66.7	0.0	1.5
4	r	-0.5	66.7	4.2	1.0
5	$\mathrm{k_{bp}}$	0.85	66.7	4.2	1.0
6	c_T	30	66.7	4.2	1.5
7	$\mathrm{k_{bp}}$	0.8	66.7	4.2	2.0
8	$\tau^{a^{som}}$	600	66.7	4.4	1.5
9	$\tau^{a^{som}}$	900	66.7	16.7	1.5
10	$\mathrm{k_{bp}}$	0.75	66.7	16.7	1.5
11	C^{som}	14	66.7	16.7	1.5
12	$\tau^{a^{som}}$	750	66.7	16.7	2.0
13	C^{som}	22	66.7	20.1	2.0
14	$\mathrm{k_{bp}}$	0.3	51.5	0.4	1.0
15	$\mathrm{k_{bp}}$	0.2	51.5	1.5	1.0
16	$\mathrm{k_{bp}}$	0.35	51.5	1.5	1.0
17	C^{som}	18	51.5	4.4	2.0
18	r	-0.1	51.5	16.9	2.5
19	$\mathrm{k_{bp}}$	0.25	50.8	2.3	1.0
20	c_T	50	50.0	0.0	0.5

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.7: The 20 best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
1	C^{som}	20	61.6	29.5	2.7
2	σ_w	60	48.5	32.3	1.0
3	$\tau^{a^{som}}$	600	45.5	4.8	1.3
4	c_T	35	44.4	0.0	0.7
5	$\mathrm{k_{bp}}$	0.95	44.4	0.3	1.0
6	$\mathrm{k_{bp}}$	0.8	44.4	2.8	1.3
7	c_T	30	44.4	8.8	1.0
8	r	-0.5	44.4	10.1	0.7
9	$\mathrm{k_{bp}}$	0.75	44.4	11.1	1.0
10	C^{som}	14	44.4	12.1	0.7
11	$\mathrm{k_{bp}}$	0.85	44.4	13.6	0.7
12	C^{som}	22	44.4	15.2	1.3
13	$\tau^{a^{som}}$	900	44.4	15.9	1.0
14	$\tau^{a^{som}}$	750	44.4	19.9	0.7
15	$\mathrm{k_{bp}}$	1	44.4	20.2	0.7
16	σ_w	50	44.4	25.0	0.7
17	r	-0.4	44.4	37.9	0.7
18	r	-0.2	39.4	11.1	1.7
19	r	-0.1	38.4	16.7	1.3
20	r	-0.3	38.4	23.0	1.7

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.8: The 20 best results from the tests with acquired data, ordered according to performance in grouping by trajectory type

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
1	C^{som}	20	56.6	39.9	0.8
2	$\mathrm{k_{bp}}$	0.8	44.4	5.6	0.4
3	$\tau^{a^{som}}$	600	34.3	13.1	0.4
4	C^{som}	22	33.3	10.6	0.3
5	$\mathrm{k_{bp}}$	0.95	33.3	11.4	0.3
6	c_T	30	33.3	27.3	0.3
7	$\mathrm{k_{bp}}$	0.75	33.3	33.3	0.3
8	$\tau^{a^{som}}$	900	33.3	35.4	0.3
9	$\tau^{a^{som}}$	750	33.3	41.2	0.3
10	r	-0.1	27.3	21.0	0.4
11	σ_w	100	26.3	2.8	0.3
12	σ_w	110	26.3	11.1	0.3
13	σ_w	70	26.3	16.4	0.4
14	r	-0.2	26.3	17.4	0.4
15	c_T	15	26.3	30.8	0.3
16	C^{som}	16	24.2	6.8	0.4
17	C^{som}	18	23.2	15.4	0.3
18	c_T	35	22.2	2.8	0.2
19	$\mathrm{k_{bp}}$	0.45	22.2	3.0	0.2
20	C^{som}	24	22.2	13.9	0.2

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

4.4.5 Results analysis

From the results presented in Tables 4.4 to 4.8, it is possible to make several deductions on the performance of the system.

From an observation of the data in Table 4.4, it is possible to see that the best results for the position criteria (see Item 2a) occurred for $\mathrm{k_{bp}} = 0.8$. Upon further analysis of the results for this parametrization, $\mathrm{k_{bp}} = 0.8$, using Figs. 4.23 to 4.27 and the data presented in Tables 4.9 to 4.13 and Table 4.14, it is possible to see that the system was capable of organizing the centroids according to their position on 4 out of the 5 trials. Even though the 1st trial was not able to completely separate the trajectories according to position, upon closer inspection using Fig. 4.28, it becomes visible that the ordering of the data is still present although, in some cases, the dispersion areas overlap.

Despite the results obtained for the position criteria, Tables 4.4 and 4.5 show that the system was incapable of obtaining a similar accuracy for the object and grasp type criteria (see Items 2b and 2c). Both these cases displayed far inferior results than the position criteria. This can be explained by the difference in the amount of displacement of the joints required by each criteria. The arm's motions for grasping an object from two considerably different positions involve a far greater variation to motion of the majority of the joints (large variations of shoulder and wrist joints) than the motions for grasping an object in two slightly different ways (involving mostly variations of the wrist). As such, it stands to reason that the system would more easily pick up on the difference between the trajectories for reaching different positions rather than the trajectories for grasping an object in a different manner. This difference is even more evident when comparing the motions to grasp objects of different sizes in relation to the motions for reaching different positions or grasping in a different manner. Therefore, the results shown in Tables 4.4 to 4.6 are consistent with what would be expected and desirable from the system – to organize the motions in accordance to the different between one another (see Section 4.4.1).

Moving on to the performance analysis for the trajectory type criteria (see Item 2e),

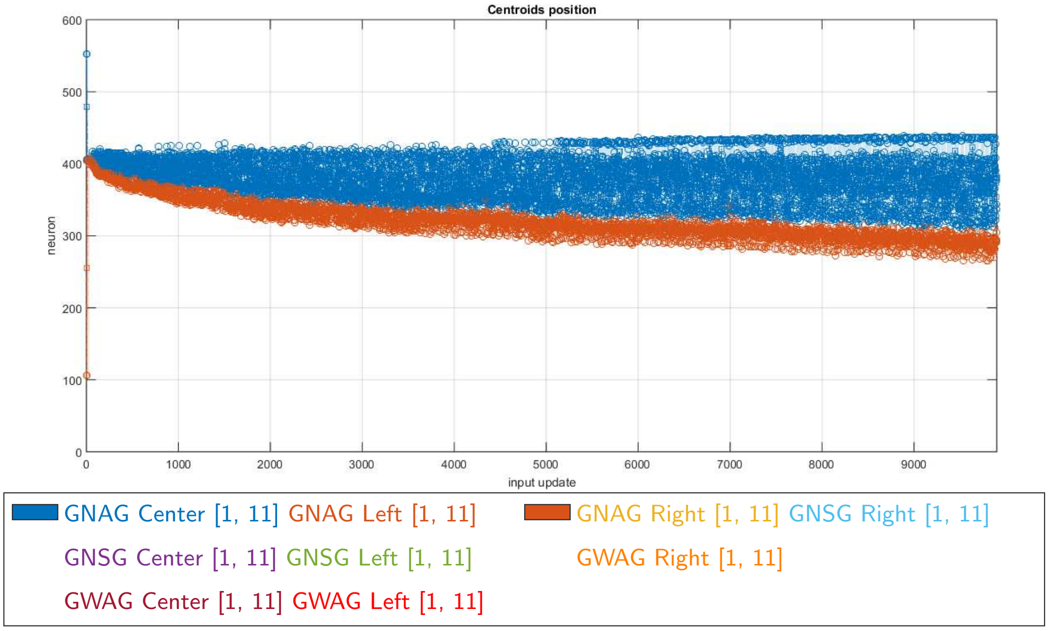


Figure 4.23: Evolution of the centroids' positions during the 1st trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

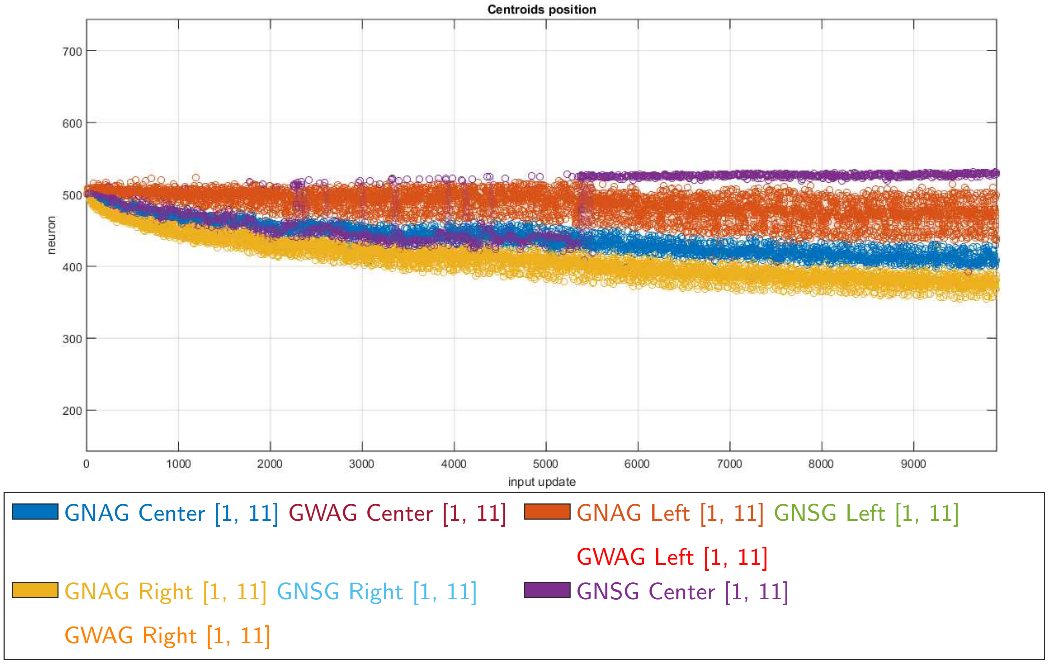


Figure 4.24: Evolution of the centroids' positions during the 2nd trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

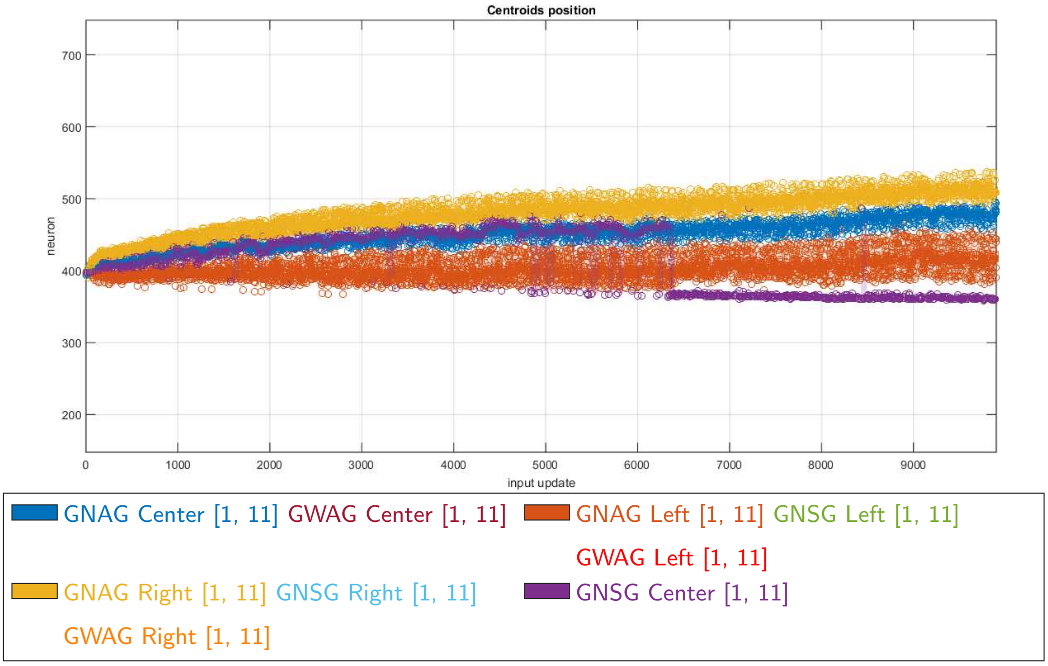


Figure 4.25: Evolution of the centroids' positions during the 3rd trial for the best parametrization according to the position criteria, $k_{bp} = 0.8$

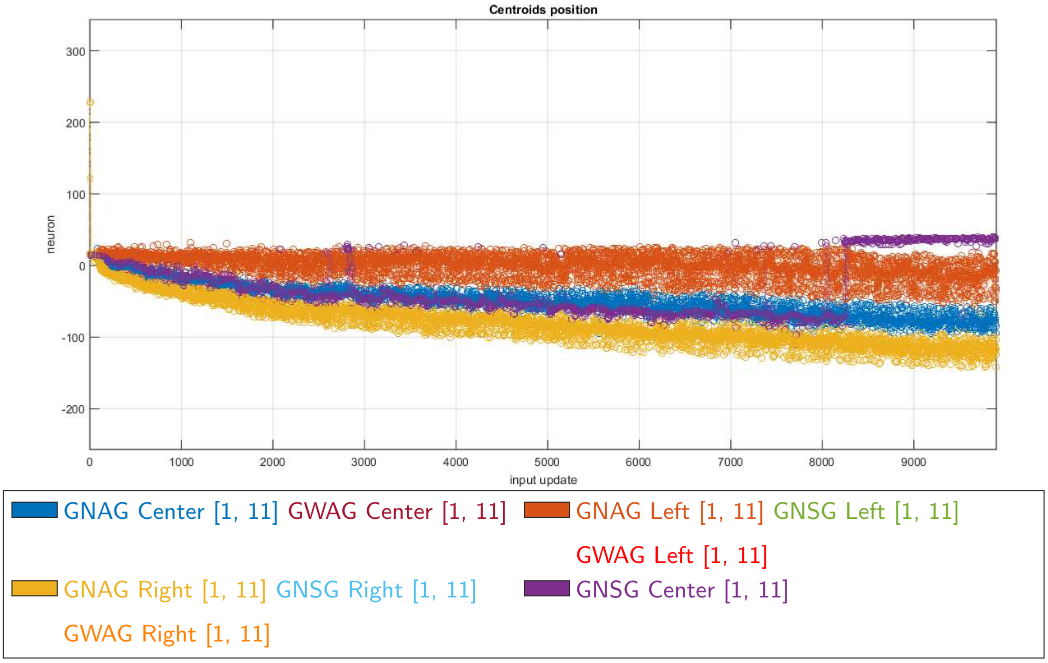


Figure 4.26: Evolution of the centroids' positions during the 4th trial for the best parametrization according to the position criteria, $k_{bp} = 0.8$

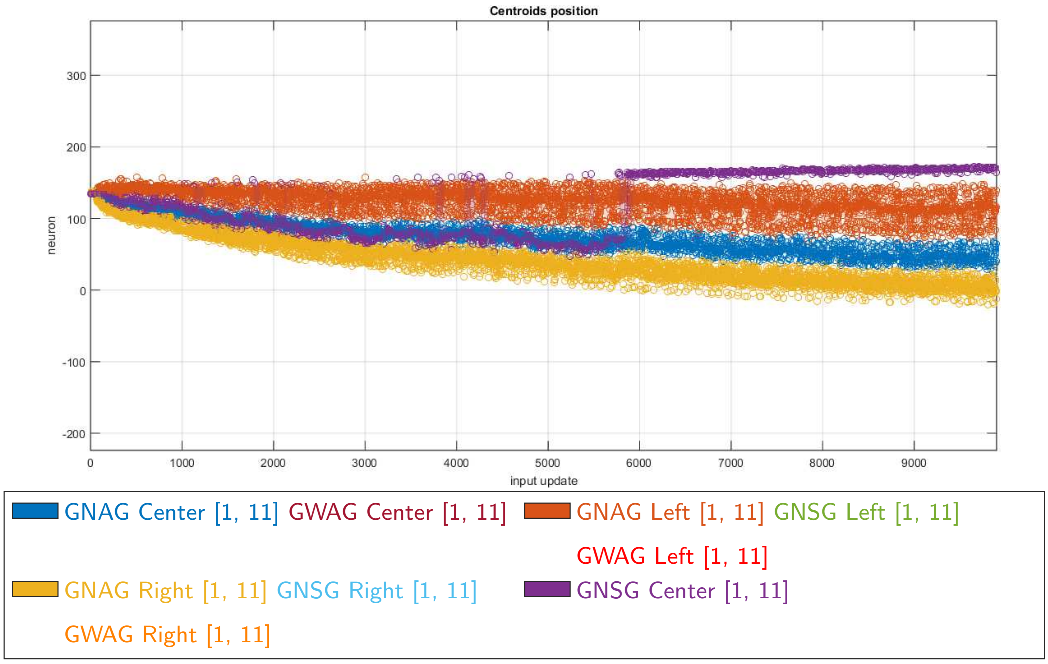


Figure 4.27: Evolution of the centroids' positions during the 5th trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

Table 4.9: Formed groups during the 1st trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Groups	Ignored
Observed	{GNAG Center [1, 11], GNAG Left [1, 11], GNSG Center [1, 11], GNSG Left [1, 11], GWAG Center [1, 11], GWAG Left [1, 11]} {GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}	
Total	2	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.
For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.10: Formed groups during the 2nd trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Groups	Ignored

Observed	{GNAG Center [1, 11], GWAG Center [1, 11]}
	{GNAG Left [1, 11], GNSG Left [1, 11], GWAG Left [1, 11]}
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}
	{GNSG Center [1, 11]}	

Total	4	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.11: Formed groups during the 3rd trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Groups	Ignored

Observed	{GNAG Center [1, 11], GWAG Center [1, 11]}
	{GNAG Left [1, 11], GNSG Left [1, 11], GWAG Left [1, 11]}
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}
	{GNSG Center [1, 11]}	

Total	4	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.12: Formed groups during the 4th trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Groups	Ignored

Observed	{GNAG Center [1, 11], GWAG Center [1, 11]}
	{GNAG Left [1, 11], GNSG Left [1, 11], GWAG Left [1, 11]}
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}
	{GNSG Center [1, 11]}	

Total	4	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.13: Formed groups during the 5th trial for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Groups	Ignored

Observed	{GNAG Center [1, 11], GWAG Center [1, 11]}
	{GNAG Left [1, 11], GNSG Left [1, 11], GWAG Left [1, 11]}
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}
	{GNSG Center [1, 11]}	

Total	4	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.14: Analysis data for the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$

	Correctly groupped			Ignored		
	Left	Center	Right	Left	Center	Right
Max percentage	100.0%	100.0%	100.0%	0.0%	0.0%	0.0%
Min percentage	100.0%	0.0%	100.0%	0.0%	0.0%	0.0%
Median percentage	100.0%	100.0%	100.0%	0.0%	0.0%	0.0%
IQR percentage	0.0%	25.0%	0.0%	0.0%	0.0%	0.0%
Median split across	1	2	1	–	–	–

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.
For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

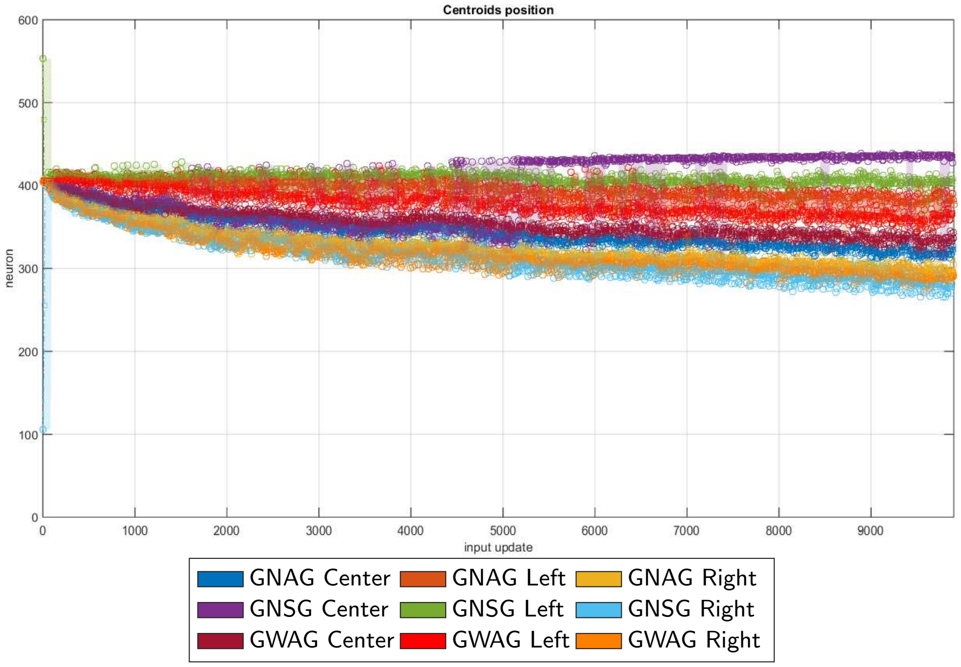


Figure 4.28: Temporal evolution of the centroid positions the best parametrization according to the position criteria, $\mathrm{k_{bp}} = 0.8$, grouped according to trajectory type

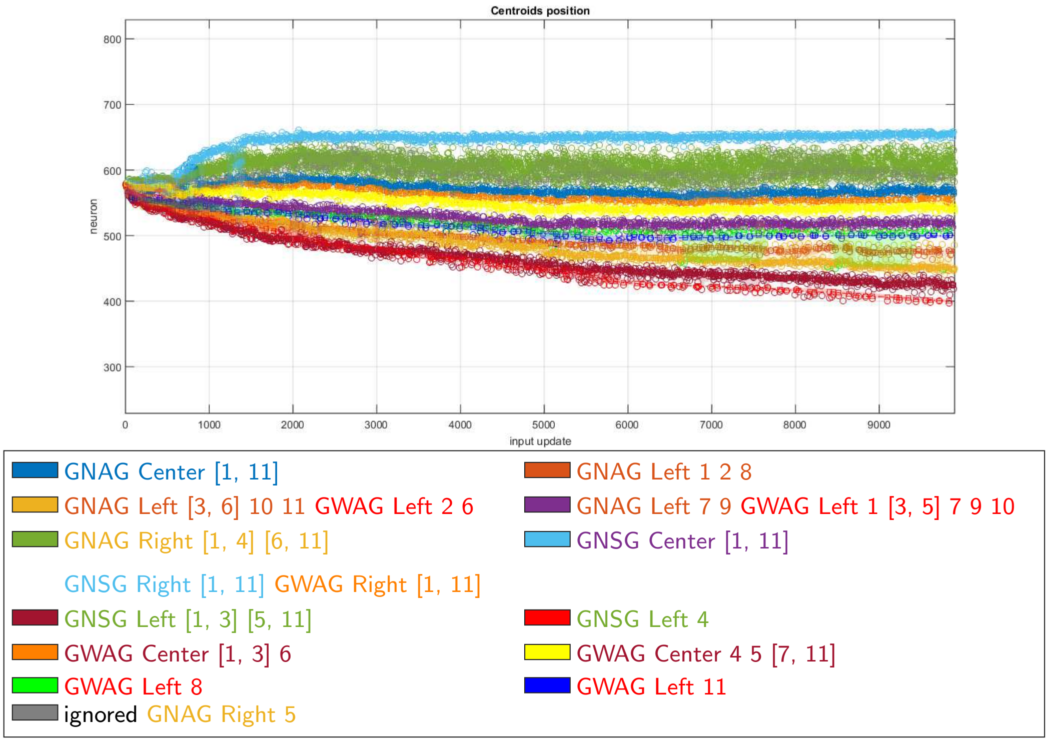


Figure 4.29: Evolution of the centroids' positions during the 1st trial for the best parametrization according to the position criteria, $C^{som} = 20$

using the data specific to the best parametrization according to Table 4.8, $C^{som} = 20$, it is possible to get a better understanding of how the system performed. This data is comprised by the temporal evolution of the centroids in each trial, Figs. 4.29 to 4.33, the groups formed and the ignored trajectories in each trial, Tables 4.15 to 4.19, as well as the values used for the analysis of the data, Table 4.20. By analysing the data from the various trials, it is possible to see that the trajectories corresponding to the object in the left position were easier to separate, followed by those from the centre and the trajectories for the right position were the hardest to separate. This stands to reason as these are the largest motions and, therefore, they are the ones that present the greatest flexibility and possibility for variability of the movement. Meaning that the greater the variability of the movements for the same type of trajectory, the easier they are to separate. Another desirable characteristic that is also evident in the data is the topographic ordering of the trajectories like in a normal the SOM. Placing the

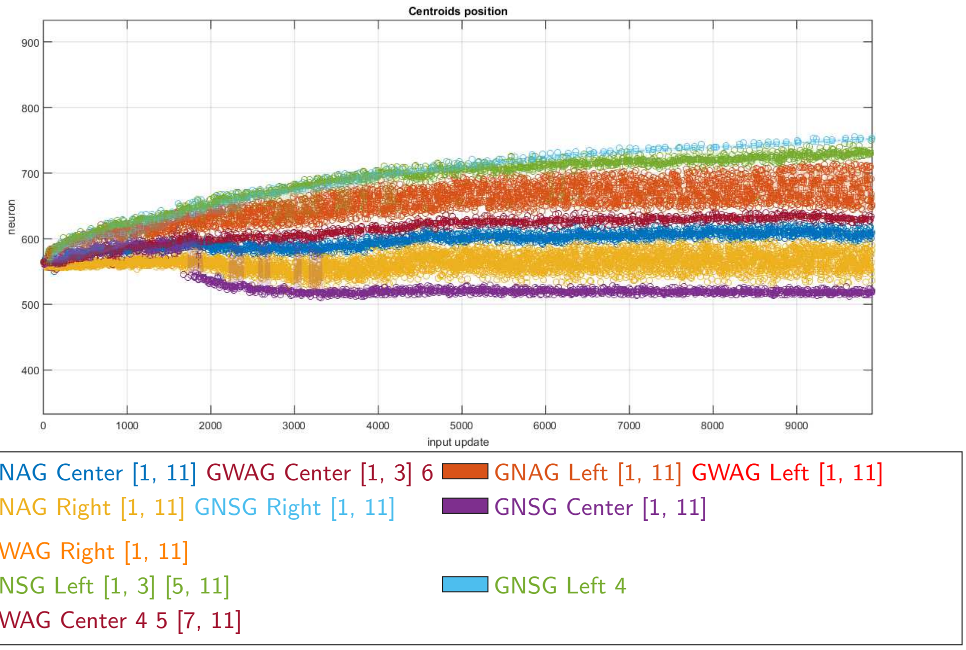

Centroids position

Figure 4.30: Evolution of the centroids' positions during the 2nd trial for the best parametrization according to the position criteria, $C^{som} = 20$

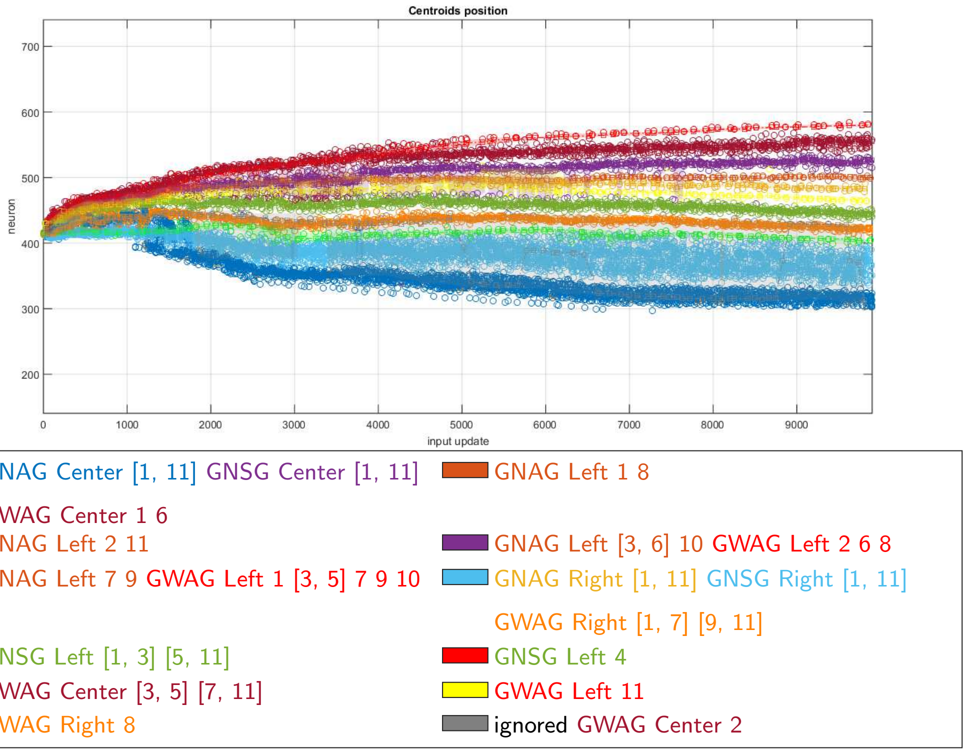

Centroids position

Figure 4.31: Evolution of the centroids' positions during the 3rd trial for the best parametrization according to the position criteria, $C^{som} = 20$

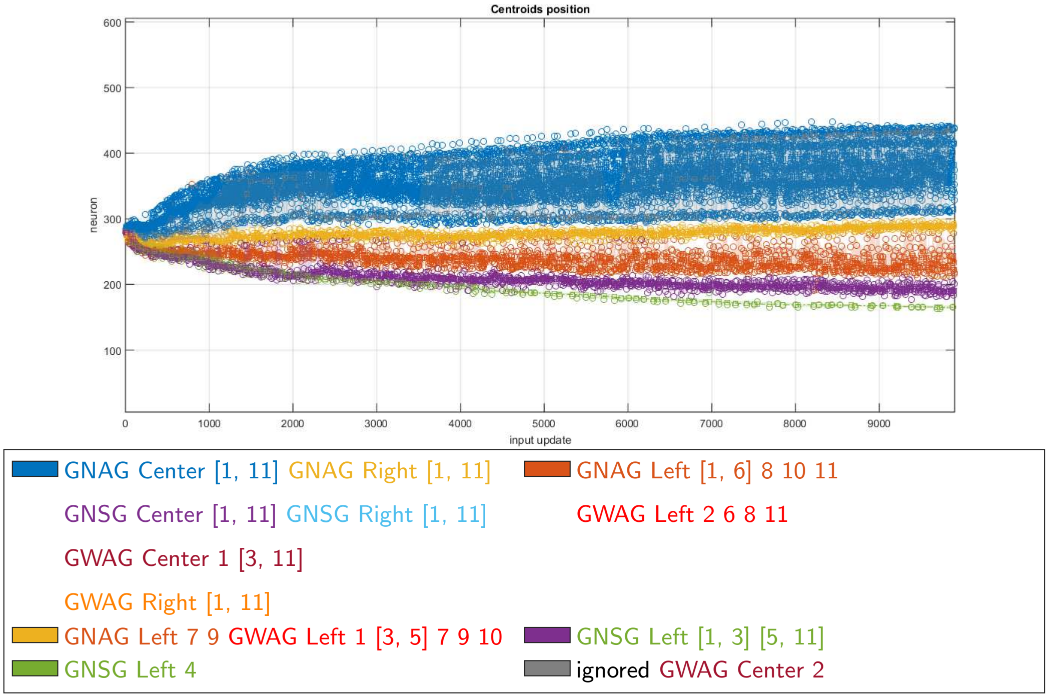


Figure 4.32: Evolution of the centroids' positions during the 4th trial for the best parametrization according to the position criteria, $C^{som} = 20$

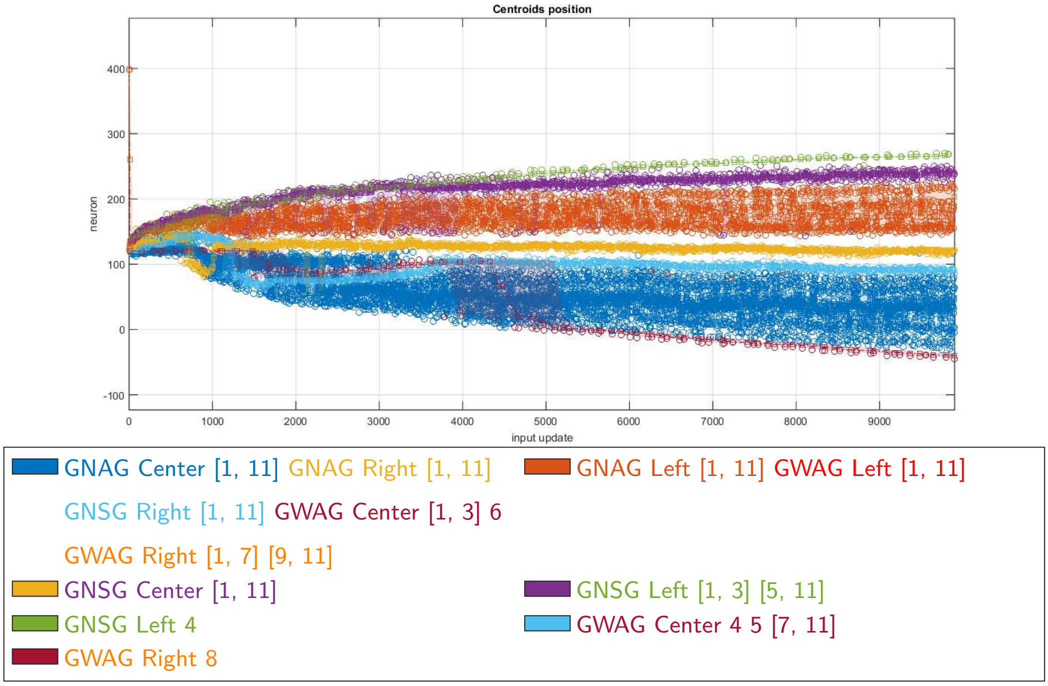


Figure 4.33: Evolution of the centroids' positions during the 5th trial for the best parametrization according to the position criteria, $C^{som} = 20$

Table 4.15: Formed groups during the 1st trial for the best parametrization according to the position criteria, $C^{som} = 20$

	Groups	Ignored
Observed	{GNAG Center [1, 11]}	GNAG Right 5
	{GNAG Left 1 2 8}	
	{GNAG Left [3, 6] 10 11, GWAG Left 2 6}	
	{GNAG Left 7 9, GWAG Left 1 [3, 5] 7 9 10}	
	{GNAG Right [1, 4] [6, 11], GNSG Right [1, 11], GWAG Right [1, 11]}	
	{GNSG Center [1, 11]}	
	{GNSG Left [1, 3] [5, 11]}	
	{GNSG Left 4}	
	{GWAG Center [1, 3] 6}	
	{GWAG Center 4 5 [7, 11]}	
	{GWAG Left 8}	
	{GWAG Left 11}	
Total	12	1

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.16: Formed groups during the 2nd trial for the best parametrization according to the position criteria, $C^{som} = 20$

	Groups	Ignored
Observed	{GNAG Center [1, 11], GWAG Center [1, 3] 6}	
	{GNAG Left [1, 11], GWAG Left [1, 11]}	
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 11]}	
	{GNSG Center [1, 11]}	
	{GNSG Left [1, 3] [5, 11]}	
	{GNSG Left 4}	
	{GWAG Center 4 5 [7, 11]}	
Total	7	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.
For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

trajectories that are the most similar to one another closer together and towards the same region of the field. To get a better understanding of how the trajectories were grouped together, Fig. 4.34 displays the number of times each trajectory was grouped with another. Using this figure it is possible to see how the trajectories for the right position were always grouped together and, sometimes with the centre positions. It is also visible how the trajectories for the object in the left position were only grouped with one another, being that the system had difficulty telling the object type type apart but not the grasp type for this object position. Finally, in the case of the object in the centre position, it is possible to see a mid point between what happened for the other two object positions. The system is starting to distinguish between the side and above grip types while also mistaking some trajectories for the GNAG and the GWAG, as was the case for the object in the left position, but will also group together trajectories from different object positions, as was the case with the object in the right position.

Tables 4.4 to 4.8 allow for the analysis os the best parametrizations making it

Table 4.17: Formed groups during the 3rd trial for the best parametrization according to the position criteria, $C^{som} = 20$

	Groups	Ignored
Observed	{GNAG Center [1, 11], GNSG Center [1, 11], GWAG Center 1 6}	GWAG Center 2
	{GNAG Left 1 8}	
	{GNAG Left 2 11}	
	{GNAG Left [3, 6] 10, GWAG Left 2 6 8}	
	{GNAG Left 7 9, GWAG Left 1 [3, 5] 7 9 10}	
	{GNAG Right [1, 11], GNSG Right [1, 11], GWAG Right [1, 7] [9, 11]}	
	{GNSG Left [1, 3] [5, 11]}	
	{GNSG Left 4}	
	{GWAG Center [3, 5] [7, 11]}	
	{GWAG Left 11}	
	{GWAG Right 8}	
Total	11	1

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.
For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.18: Formed groups during the 4th trial for the best parametrization according to the position criteria, $C^{som} = 20$

	Groups	Ignored
Observed	{GNAG Center [1, 11], GNAG Right [1, 11], GNSG Center [1, 11], GNSG Right [1, 11], GWAG Center 1 [3, 11], GWAG Right [1, 11]}	GWAG Center 2
	{GNAG Left [1, 6] 8 10 11, GWAG Left 2 6 8 11}	
	{GNAG Left 7 9, GWAG Left 1 [3, 5] 7 9 10}	
	{GNSG Left [1, 3] [5, 11]}	
	{GNSG Left 4}	
Total	5	1

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.19: Formed groups during the 5th trial for the best parametrization according to the position criteria, $C^{som} = 20$

	Groups	Ignored
Observed	{GNAG Center [1, 11], GNAG Right [1, 11], GNSG Right [1, 11], GWAG Center [1, 3] 6, GWAG Right [1, 7] [9, 11]}	
	{GNAG Left [1, 11], GWAG Left [1, 11]}	
	{GNSG Center [1, 11]}	
	{GNSG Left [1, 3] [5, 11]}	
	{GNSG Left 4}	
	{GWAG Center 4 5 [7, 11]}	
	{GWAG Right 8}	
Total	7	0

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.20: Analysis data for the best parametrization according to the position criteria, $C^{som} = 20$

	Correctly groupped								
	GNAG L	GNAG C	GNAG R	GNSG L	GNSG C	GNSG R	GWAG L	GWAG C	GWAG R
Max %	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	81.8%	100.0%	9.1%
Min %	81.8%	100.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Median %	81.8%	100.0%	0.0%	100.0%	100.0%	0.0%	63.6%	63.6%	0.0%
IQR %	18.2%	0.0%	100.0%	0.0%	100.0%	25.0%	75.0%	31.8%	9.1%
Median split	1	1	0	2	1	0	1	1	0

	Ignored								
	GNAG L	GNAG C	GNAG R	GNSG L	GNSG C	GNSG R	GWAG L	GWAG C	GWAG R
Max %	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%
Min %	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Median %	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
IQR %	0.0%	0.0%	2.3%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%
Median split	–	–	–	–	–	–	–	–	–

For the SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

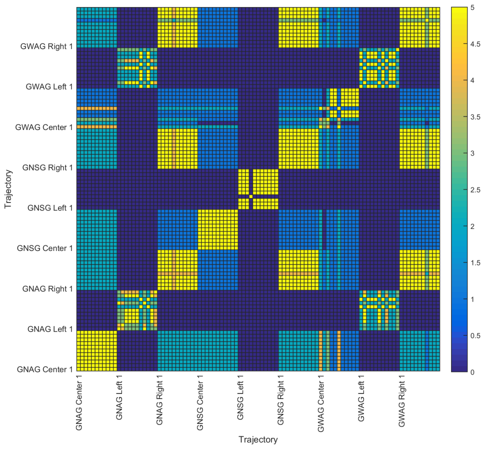


Figure 4.34: Number of times that each trajectory was grouped with another $\mathrm{k_{bp}} = 0.8$

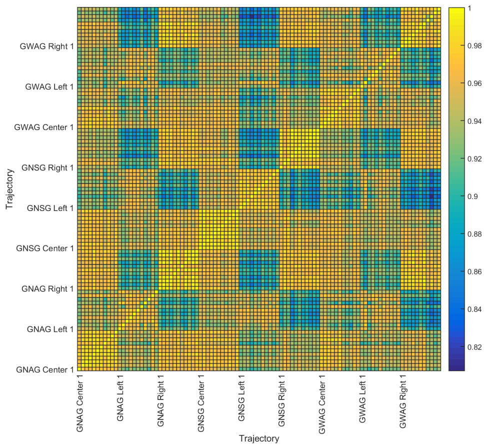


Figure 4.35: Dot product between the input vectors of the field

possible to deduce that the most important parameter is the input gain for the DNFL, C^{som} (which always ranked amongst the top 3 parameters in terms of performance). It also makes it possible to see which of the remaining parameters influence the groupings for each criteria the most, such as the the back propagation gain, $\mathrm{k_{bp}}$ which always ranked amongst the top 5 parameters except for the case of the grouping by object type (see Table 4.5). Despite this, it does not allow for an analysis of the influence on the stability of the system. For that purpose Table 4.21 can be used. In this table, any value for the mean of the median percentage far below 33.3% means that a large number of populations had to be ignored. The only situation in which this happens is when the fraction of iterations that produced a stable centroid in the DNFL was inferior to the fraction of the iterations that must have produced a stable centroid in the DNFL, $\mathrm{success\,rate}$. In these cases the results are disregarded as the system is either unstable or has difficulty creating representations for the trajectories. As expected, given the alterations done to (2.18) in order to stabilize the field (obtaining (3.7) and (3.8), see Chapter 3), the system is only incapable of forming sufficient stable centroids when the field the SODNF is incorrectly parametrized. In the case at hand it happened when the resting level, r is inferior to -0.5 (case in which the system has a difficulty becoming

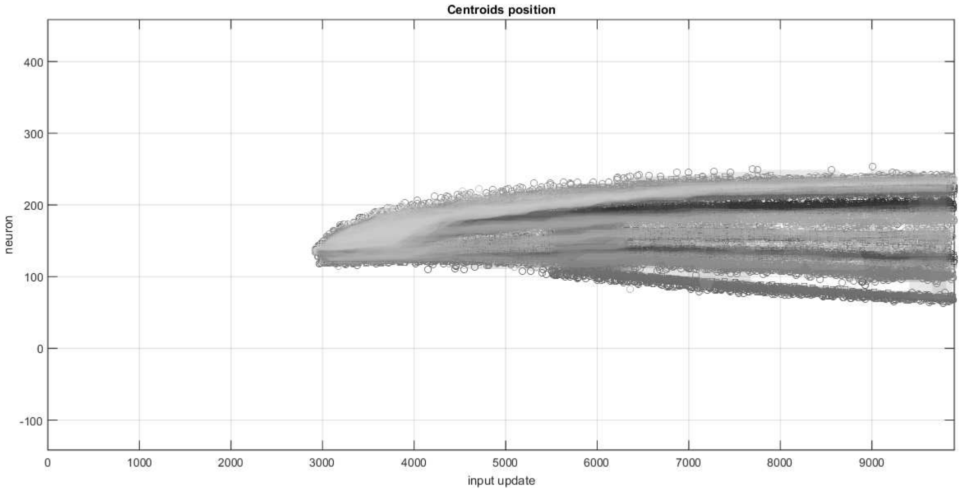


Figure 4.36: Evolution of the centroids' positions during the 1st trial for $r = -0.6$, in which the field has a difficulty forming the first stable centroid

active given the value of the input gain for the DNFL, C^{som}, see Fig. 4.36), the field did not have enough time to form a stale peak ($c_T = 5$) or the kernel was too narrow ($\sigma_w = 200$).

Another important aspect to analyse, especially given the alterations to the original system (see Chapter 3, (3.1)) is the effect of the back propagation gain, $\mathrm{k_{bp}}$. This can be achieved by analysing Tables 4.22 to 4.26. From these tables it is possible to see that the best performance is achieved when $\mathrm{k_{bp}} \approx 0.8$[2], not 1. This behaviour is most likely doe to the fact that, for $\mathrm{k_{bp}} = 1$ and given enough time, the system will reach a point where it will stop updating the weights of the the SODNF. For a value of approximately 0.8 this does not happen and the system will continuously update but at a slower rate. This fact can have an adverse effect as, for longer iteration times, other populations may be overwritten.

[2] Given that the tables presented were made considering that the greater the variation of the median the worst the performance (see Item 4b), than, when the median percentage is approximately $1/\mathrm{the\,number\,of\,criteria}$, the values in the table with a greater IQR most likely showed better results than those with a lower IQR (as is the case for the object type criteria).

Table 4.21: The 20 worst results from the tests with acquired data, ordered according to performance in grouping by position (best performances first)

			Mean of		
pos	parameter	value	Median %	IQR %	Split %
71	$\mathrm{k_{bp}}$	0.7	33.3	41.7	0.3
72	$\mathrm{k_{bp}}$	0.35	33.3	49.0	0.3
73	σ_w	180	33.3	62.6	0.3
74	σ_w	160	33.3	63.6	0.3
75	$\mathrm{k_{bp}}$	0.3	33.3	65.7	0.3
76	$\tau^{a^{som}}$	1200	33.3	65.9	0.3
77	$\tau^{a^{som}}$	1050	33.3	66.7	0.3
78	C^{som}	8	33.3	66.7	0.3
79	C^{som}	28	33.3	66.7	0.3
80	C^{som}	32	33.3	66.7	0.3
81	C^{som}	34	33.3	66.7	0.3
82	C^{som}	10	33.3	74.7	0.3
83	σ_w	170	4.0	41.7	0.3
84	$\mathrm{k_{bp}}$	0.2	1.0	74.0	0.3
85	r	-1	0.0	0.0	0.0
86	r	-0.9	0.0	0.0	0.0
87	r	-0.8	0.0	0.0	0.0
88	r	-0.7	0.0	0.0	0.0
89	c_T	5	0.0	0.0	0.0
90	σ_w	200	0.0	16.7	0.0
91	r	-0.6	0.0	52.8	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.22: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the back propagation gain

		Mean of		
pos	$\mathrm{k_{bp}}$	Median %	IQR %	Split %
1	0.80	100.0	8.3	1.3
2	0.95	77.8	22.2	1.0
3	0.75	77.8	66.7	1.0
4	0.45	66.7	8.3	0.7
5	0.60	66.7	33.3	0.7
6	0.65	66.7	33.3	0.7
7	1.00	45.5	35.4	1.0
8	0.90	45.5	42.4	1.0
9	0.85	44.4	38.6	0.7
10	0.25	34.3	65.7	0.7
11	0.00	33.3	0.0	0.3
12	0.05	33.3	0.0	0.3
13	0.55	33.3	8.1	0.3
14	0.15	33.3	16.4	0.3
15	0.40	33.3	16.4	0.3
16	0.10	33.3	16.7	0.3
17	0.50	33.3	33.3	0.3
18	0.70	33.3	41.7	0.3
19	0.35	33.3	49.0	0.3
20	0.30	33.3	65.7	0.3
21	0.20	1.0	74.0	0.3

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.23: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the back propagation gain

		Mean of		
pos	$\mathrm{k_{bp}}$	Median %	IQR %	Split %
1	0.00	50.0	0.0	0.5
2	0.15	50.0	0.0	0.5
3	0.85	50.0	0.0	1.0
4	0.80	50.0	0.0	2.0
5	0.10	50.0	0.2	0.5
6	0.70	50.0	0.2	0.5
7	0.20	50.0	0.2	1.0
8	0.65	50.0	0.2	1.0
9	0.75	50.0	0.2	1.5
10	0.90	50.0	0.4	1.0
11	0.95	50.0	0.4	1.5
12	1.00	50.0	0.6	1.5
13	0.05	50.0	0.8	0.5
14	0.25	50.0	0.9	1.0
15	0.50	49.2	0.2	0.5
16	0.45	49.2	0.2	1.0
17	0.40	49.2	0.8	0.5
18	0.60	49.2	0.8	1.0
19	0.35	49.2	1.0	1.0
20	0.55	49.2	1.5	0.5
21	0.30	49.2	1.5	1.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.24: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the back propagation gain

		Mean of		
pos	$\mathrm{k_{bp}}$	Median %	IQR %	Split %
1	0.95	66.7	0.0	1.5
2	0.85	66.7	4.2	1.0
3	0.80	66.7	4.2	2.0
4	0.75	66.7	16.7	1.5
5	0.30	51.5	0.4	1.0
6	0.20	51.5	1.5	1.0
7	0.35	51.5	1.5	1.0
8	0.25	50.8	2.3	1.0
9	0.00	50.0	0.0	0.5
10	0.05	50.0	0.0	0.5
11	0.60	50.0	0.0	1.0
12	0.65	50.0	0.0	1.0
13	0.10	50.0	0.2	0.5
14	0.55	50.0	0.2	0.5
15	0.15	50.0	0.4	0.5
16	0.40	50.0	0.4	0.5
17	0.50	50.0	0.4	0.5
18	0.45	50.0	0.4	1.0
19	0.70	50.0	4.2	0.5
20	0.90	50.0	5.3	1.0
21	1.00	50.0	6.4	0.5

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table 4.25: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the back propagation gain

		Mean of		
pos	k_{bp}	Median %	IQR %	Split %
1	0.95	44.4	0.3	1.0
2	0.80	44.4	2.8	1.3
3	0.75	44.4	11.1	1.0
4	0.85	44.4	13.6	0.7
5	1.00	44.4	20.2	0.7
6	0.30	34.3	0.3	0.7
7	0.20	34.3	1.0	0.7
8	0.35	34.3	1.0	0.7
9	0.25	34.3	17.7	0.7
10	0.00	33.3	0.0	0.3
11	0.05	33.3	0.0	0.3
12	0.10	33.3	0.0	0.3
13	0.60	33.3	0.0	0.7
14	0.65	33.3	0.0	0.7
15	0.15	33.3	0.3	0.3
16	0.40	33.3	0.3	0.3
17	0.50	33.3	0.3	0.3
18	0.45	33.3	0.3	0.7
19	0.70	33.3	2.8	0.3
20	0.90	33.3	14.1	0.7
21	0.55	33.3	16.7	0.3

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table 4.26: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the back propagation gain

		Mean of		
pos	$\mathrm{k_{bp}}$	Median %	IQR %	Split %
1	0.80	44.4	5.6	0.4
2	0.95	33.3	11.4	0.3
3	0.75	33.3	33.3	0.3
4	0.45	22.2	3.0	0.2
5	0.85	22.2	19.2	0.2
6	0.30	12.1	0.3	0.2
7	0.20	12.1	1.0	0.2
8	0.25	12.1	1.0	0.2
9	0.35	12.1	1.0	0.2
10	0.00	11.1	0.0	0.1
11	0.05	11.1	0.0	0.1
12	0.10	11.1	0.0	0.1
13	0.15	11.1	0.3	0.1
14	0.40	11.1	0.3	0.1
15	0.55	11.1	2.8	0.1
16	0.50	11.1	11.4	0.1
17	0.60	11.1	13.9	0.1
18	0.65	11.1	13.9	0.1
19	0.70	11.1	16.7	0.1
20	0.90	11.1	22.5	0.1
21	1.00	11.1	34.1	0.1

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Chapter 5

Conclusion and outlook

5.1 Conclusion

From the data analysed in Sections 4.3.3 and 4.4.4 it is possible to conclude that this system is capable of handling acquired data and processing it, to a certain extent (see Section 4.4.4), without the assistance of any filter thanks to the systems internal characteristics. Given that the default parameters used for the system were not the optimum parameters, the results presented in this work could well be improved further. To improve the performance of the system further a filter could be applied to the joint angles. This would smooth out some of the errors from incorrect estimations of the joint positions (see Fig. 4.22) as well as small noises. If the data from other metrics is added, such as the object colour (as was the case in [31]), the system could then more easily distinguish between the various objects – aspect that the system performed the worst at.

5.2 Future work

Despite the system being capable of functioning correctly and displaying some good results with unfiltered acquired data, there is still room for further improvement.

Although, in this work, the object position was taken into consideration, in some cases it may be irrelevant or undesirable for the task at hand. Given that this was the main characteristic that the system picked up on, one alternative may be to use only the final position and orientation of the hand – the position and orientation right before grasping the object or at the moment the object is grasped – relative to the target object rather than the absolute joint angles. This would remove part of the variability of the motion caused by obstructions in the environment, given that in certain cases the obstruction and relative position of the user may limit the possible grasping positions.

Another problem that still needs to be addressed is the fact that some parametrizations displayed a large IQR for the number of correctly grouped trajectories (see Table 4.22 for instance). This is due to the reduced number of trials for each parametrization (only 5), a restriction imposed by the time that it takes to run the simulations for all the parametrizations, given the available resources. As such more trials should be added to ensure the consistency of the results. Also, although this study involved a considerable amount of parameters, the effect of the combinations of the parameters was not, due to the time required for such a task given the available resources. In the case of parameters such as the input gain for the DNFL, C^{som} and the resting level, r this will be an important factor to take into account. Therefore, these tests still need to be added. In addition, the result should also be compared to those obtained using filtered data to see how it will effect the performance of the system.

Lastly, a comparison between the performance of a uni-dimensional and a bi-dimensional map should be tested. This will require some alterations to the way in which the data is analysed since the dispersion area can not be calculated the same way as in a uni-dimensional field. To that end, a method such as cluster analysis (see [13]) could be used to determine the formed groups from all the centroid positions. These groups could then be compared with what was desirable and the performance measured (possibly in terms of the percentage of elements in each obtained group that belonged to a specific desirable group).

Appendix A

Mathematical profs

A.1 Equation 4.26

$$
\begin{aligned}
{}^{\text{joint}}\hat{v}_{\text{next}} &= \operatorname{Rot}\Big((0,1,0)^T, \theta_y\Big) \operatorname{Rot}\Big((1,0,0)^T, \theta_x\Big)\, {}^{\text{next}}\hat{v}_{\text{next}} && \text{(A.1)} \\
&= \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && \text{(A.2)} \\
&= \begin{pmatrix} \cos(\theta_y) & \sin(\theta_x)\sin(\theta_y) & \cos(\theta_x)\sin(\theta_y) \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ -\sin(\theta_y) & \sin(\theta_x)\cos(\theta_y) & \cos(\theta_x)\cos(\theta_y) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && \text{(A.3)} \\
&= \begin{pmatrix} \cos(\theta_x)\sin(\theta_y) \\ -\sin(\theta_x) \\ \cos(\theta_x)\cos(\theta_y) \end{pmatrix} && \text{(A.4)}
\end{aligned}
$$

A.2 Equation 4.36

$$
\begin{aligned}
{}^{\mathrm{pos\,rot\,z}}\hat{v}_{\mathrm{after\,next\,l}} &= \mathrm{Rot}\Big((1,0,0)^T, -\theta_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\theta_y\Big)\,{}^{\mathrm{after\,next}}\hat{v}_{\mathrm{after\,next\,l}} && \text{(A.5)}\\
&= \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && \text{(A.6)}\\
&= \begin{pmatrix} \cos(\theta_z)\cos(\theta_y) & -\sin(\theta_z) & \cos(\theta_z)\sin(\theta_y) \\ \sin(\theta_z)\cos(\theta_y) & \cos(\theta_z) & \sin(\theta_z)\sin(\theta_y) \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && \text{(A.7)}\\
&= \begin{pmatrix} \cos(\theta_z)\sin(\theta_y) \\ \sin(\theta_z)\sin(\theta_y) \\ \cos(\theta_y) \end{pmatrix} && \text{(A.8)}
\end{aligned}
$$

A.3 Equation 4.42

$$
\begin{aligned}
{}^{\mathrm{next}}\hat{v}_{\mathrm{joint}} &= \mathrm{Rot}\Big((1,0,0)^T, -\theta_x\Big)\,\mathrm{Rot}\Big((0,1,0)^T, -\theta_y\Big)\,{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} && \text{(A.9)}\\
&= \Big(\mathrm{Rot}\Big((0,1,0)^T, \theta_y\Big)\,\mathrm{Rot}\Big((1,0,0)^T, \theta_x\Big)\Big)^T\,{}^{\mathrm{joint}}\hat{v}_{\mathrm{link}} && \text{(A.10)}\\
&= \begin{pmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ \sin(\theta_x)\sin(\theta_y) & \cos(\theta_x) & \sin(\theta_x)\cos(\theta_y) \\ \cos(\theta_x)\sin(\theta_y) & -\sin(\theta_x) & \cos(\theta_x)\cos(\theta_y) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && \text{(A.11)}\\
&= \begin{pmatrix} -\sin(\theta_y) \\ \sin(\theta_x)\cos(\theta_y) \\ \cos(\theta_x)\cos(\theta_y) \end{pmatrix} && \text{(A.12)}
\end{aligned}
$$

A.4 Equation 4.78

Let

$$\hat{x} = (1,0,0)^T \tag{A.13}$$

$$\hat{y} = (1,0,0)^T \tag{A.14}$$

$$\hat{z} = (1,0,0)^T \tag{A.15}$$

than

$$\mathrm{Rot}\big(\hat{y}, -(\theta_y+\pi)\big)\,\mathrm{Rot}\big(\hat{y},\theta_y\big) = \mathrm{Rot}(\hat{y},\pi) \tag{A.16}$$

$$\mathrm{Rot}(\hat{x}, -(\pi-\theta_x)) = \mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{x},\pi) \tag{A.17}$$

$$\mathrm{Rot}(\hat{x},\pi)\,\mathrm{Rot}(\hat{y},\pi) = \mathrm{Rot}(\hat{z},\pi) \tag{A.18}$$

$$\mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{z},\pi)\,\mathrm{Rot}(\hat{x},\theta_x) = \mathrm{Rot}(\hat{z},\pi). \tag{A.19}$$

As such

$$\begin{aligned}
{}^{\mathrm{pre\,rot\,z}}P &= \mathrm{Rot}(\hat{x}, -(\pi-\theta_x))\,\mathrm{Rot}\big(\hat{y}, -(\theta_y+\pi)\big)\,{}^{\mathrm{joint}}\mathrm{R}_{\mathrm{next}}\,{}^{\mathrm{next}}P \\
&= \mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{x},\pi)\,\mathrm{Rot}\big(\hat{y}, -(\theta_y+\pi)\big)\,\mathrm{Rot}\big(\hat{y},\theta_y\big)\,\mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{z},\theta_z)\,{}^{\mathrm{next}}P \\
&= \mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{x},\pi)\,\mathrm{Rot}(\hat{y},\pi)\,\mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{z},\theta_z)\,{}^{\mathrm{next}}P \\
&= \mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{z},\pi)\,\mathrm{Rot}(\hat{x},\theta_x)\,\mathrm{Rot}(\hat{z},\theta_z)\,{}^{\mathrm{next}}P \\
&= \mathrm{Rot}(\hat{z},\pi)\,\mathrm{Rot}(\hat{z},\theta_z)\,{}^{\mathrm{next}}P \\
&= \mathrm{Rot}(\hat{z},\theta_z+\pi)\,{}^{\mathrm{next}}P
\end{aligned} \tag{A.20}$$

This page has been intentionally left blank.

Appendix B

Additional equations

B.1 Joint scaling

$$\theta'_j(t) = \frac{\theta_j(t) - \theta_{j_{min}}}{\theta_{j_{max}} - \theta_{j_{min}}} \tag{B.1}$$

B.2 Dependent iteration time

$$T_p = c_T \Big(t^{en}_{p+1} - t^{en}_p \Big) \tag{B.2}$$

This page has been intentionally left blank.

Appendix C

Field Parameters

The global parameters selected for the field were:

Table C.1: Field parameters

Parameter	Value
Δt	0.01 seconds
n_{som}	600
τ_{DNFL}	$7\Delta t$
A_w	1.2
w_{inhib}	-0.5
ζ^k_{stoch}	1
$\mathrm{d}x$	1

for the output function $f(u)$ the Heaviside step function was used.

This page has been intentionally left blank.

Appendix D

Additional tables

D.1 standard deviation of the interaction kernel

Table D.1: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the standard deviation of the interaction kernel

		Mean of		
pos	σ_w	Median %	IQR %	Split %
1	100	66.7	7.6	1.0
2	110	66.7	29.3	1.0
3	60	66.7	43.4	1.0
4	70	66.7	57.3	1.0
5	50	44.4	41.4	0.7
6	120	37.4	0.0	0.7
7	130	37.4	7.6	0.7
8	80	37.4	16.4	1.0
9	90	37.4	16.4	1.0
10	140	37.4	37.6	0.7
11	150	37.4	66.9	0.7
12	40	34.3	12.4	0.7
13	10	33.3	0.0	0.3
14	20	33.3	0.0	0.3
15	30	33.3	16.7	0.3
16	190	33.3	33.3	0.3
17	180	33.3	62.6	0.3
18	160	33.3	63.6	0.3
19	170	4.0	41.7	0.3
20	200	0.0	16.7	0.0

For default the SODNF parameters$\tau^{a^{som}} = 600$, $\mathrm{k}_{\mathrm{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.2: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the standard deviation of the interaction kernel

		Mean of		
pos	σ_w	Median %	IQR %	Split %
1	40	51.5	1.9	1.0
2	30	50.0	0.0	0.5
3	60	50.0	0.0	1.5
4	80	50.0	0.0	1.5
5	70	50.0	0.2	1.5
6	50	50.0	0.4	1.0
7	20	50.0	0.8	0.5
8	100	49.2	1.1	1.5
9	180	49.2	0.7	0.5
10	10	49.2	1.1	0.5
11	120	48.5	0.2	1.0
12	110	48.5	0.2	1.5
13	130	48.5	0.6	1.0
14	90	48.5	1.9	1.5
15	140	48.5	0.2	1.0
16	160	48.5	0.4	0.5
17	150	48.5	13.1	0.5
18	170	48.5	49.2	0.5
19	190	48.4	49.2	0.5
20	200	0.0	48.7	0.0

For default the SODNF parameters$\tau^{a^{som}} = 600$, $\mathrm{k}_{\mathrm{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.3: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the standard deviation of the interaction kernel

		Mean of		
pos	σ_w	Median %	IQR %	Split %
1	20	50.0	0.8	0.5
2	80	50.0	0.8	1.5
3	30	50.0	4.2	0.5
4	70	50.0	5.5	1.5
5	60	50.0	8.3	1.5
6	40	50.0	16.7	1.0
7	50	50.0	16.7	1.0
8	180	49.2	0.9	0.5
9	10	49.2	1.6	0.5
10	100	49.2	0.8	1.5
11	130	48.5	0.8	1.0
12	90	48.5	1.0	1.5
13	120	48.5	0.2	1.0
14	110	48.5	0.2	1.5
15	140	48.5	0.2	1.0
16	160	48.5	0.4	0.5
17	190	48.5	49.2	0.5
18	170	48.5	49.2	0.5
19	150	48.4	12.9	0.5
20	200	0.0	48.5	0.0

For default the SODNF parameters$\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.4: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the standard deviation of the interaction kernel

		Mean of		
pos	σ_w	Median %	IQR %	Split %
1	60	48.5	32.3	1.0
2	50	44.4	25.0	0.7
3	130	37.4	65.0	1.0
4	80	37.4	16.4	1.0
5	110	36.7	42.4	1.0
6	90	36.4	11.4	1.0
7	100	36.4	44.4	1.0
8	120	36.3	16.2	0.7
9	40	34.3	12.4	0.7
10	70	34.3	39.9	1.0
11	30	33.3	2.8	0.3
12	20	33.3	16.7	0.3
13	180	33.3	61.6	0.3
14	10	33.3	65.7	0.3
15	160	32.3	16.2	0.3
16	150	4.0	64.9	0.3
17	140	4.0	68.7	0.3
18	200	0.0	16.4	0.0
19	170	0.0	27.5	0.0
20	190	0.0	41.4	0.0

For default the SODNF parameters$\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.5: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the standard deviation of the interaction kernel

		Mean of		
pos	σ_w	Median %	IQR %	Split %
1	100	26.3	2.8	0.3
2	110	26.3	11.1	0.3
3	70	26.3	16.4	0.4
4	50	22.2	25.0	0.2
5	60	22.2	32.1	0.2
6	120	15.2	0.0	0.2
7	130	15.2	3.1	0.2
8	80	15.2	5.3	0.3
9	90	15.2	5.8	0.2
10	140	15.2	9.8	0.2
11	150	15.2	22.5	0.2
12	40	12.1	12.4	0.2
13	10	11.1	0.0	0.1
14	20	11.1	0.0	0.1
15	30	11.1	5.6	0.1
16	190	11.1	11.1	0.1
17	180	11.1	18.2	0.1
18	160	11.1	19.2	0.1
19	170	4.0	13.9	0.1
20	200	0.0	11.1	0.0

For default the SODNF parameters$\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

D.2 input gain for the DNFL

Table D.6: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the input gain for the DNFL

		Mean of		
pos	C^{som}	Median %	IQR %	Split %
1	20	99.0	19.9	2.3
2	16	67.7	17.2	1.3
3	18	66.7	3.5	1.3
4	22	66.7	21.5	1.0
5	14	45.5	25.8	1.0
6	24	44.4	41.7	0.7
7	6	33.3	0.0	0.3
8	30	33.3	0.0	0.3
9	38	33.3	0.0	0.3
10	40	33.3	0.0	0.3
11	42	33.3	0.0	0.3
12	44	33.3	0.0	0.3
13	46	33.3	0.0	0.3
14	26	33.3	16.7	0.3
15	36	33.3	16.7	0.3
16	48	33.3	16.7	0.3
17	50	33.3	16.7	0.3
18	12	33.3	33.3	0.3
19	8	33.3	66.7	0.3
20	28	33.3	66.7	0.3
21	32	33.3	66.7	0.3
22	34	33.3	66.7	0.3
23	10	33.3	74.7	0.3

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.7: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the input gain for the DNFL

		Mean of		
pos	C^{som}	Median %	IQR %	Split %
1	20	60.6	18.0	4.0
2	16	51.5	0.8	2.0
3	18	51.5	4.9	2.0
4	12	50.0	0.0	0.5
5	30	50.0	0.0	0.5
6	38	50.0	0.0	0.5
7	40	50.0	0.0	0.5
8	42	50.0	0.0	0.5
9	44	50.0	0.0	0.5
10	48	50.0	0.0	0.5
11	32	50.0	0.2	0.5
12	36	50.0	0.2	0.5
13	46	50.0	0.2	0.5
14	50	50.0	0.2	0.5
15	24	50.0	0.2	1.0
16	6	50.0	0.8	0.5
17	14	50.0	1.5	1.0
18	22	50.0	3.0	2.0
19	10	49.2	0.2	0.5
20	8	49.2	0.8	0.5
21	34	49.2	0.8	0.5
22	28	49.2	0.0	0.5
23	26	49.2	0.8	0.5

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.8: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the input gain for the DNFL

		Mean of		
pos	C^{som}	Median %	IQR %	Split %
1	20	82.6	17.4	3.5
2	14	66.7	16.7	1.5
3	22	66.7	20.1	2.0
4	18	51.5	4.4	2.0
5	12	50.0	0.0	0.5
6	30	50.0	0.0	0.5
7	36	50.0	0.0	0.5
8	38	50.0	0.0	0.5
9	40	50.0	0.0	0.5
10	42	50.0	0.0	0.5
11	44	50.0	0.0	0.5
12	46	50.0	0.0	0.5
13	50	50.0	0.0	0.5
14	34	50.0	0.2	0.5
15	48	50.0	0.2	0.5
16	16	50.0	0.2	2.0
17	10	50.0	0.4	0.5
18	8	50.0	0.8	0.5
19	32	50.0	0.8	0.5
20	24	50.0	4.4	1.0
21	6	49.2	0.8	0.5
22	26	49.2	0.9	0.5
23	28	49.2	0.9	0.5

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.9: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the input gain for the DNFL

		Mean of		
pos	C^{som}	Median %	IQR %	Split %
1	20	61.6	29.5	2.7
2	14	44.4	12.1	0.7
3	22	44.4	15.2	1.3
4	16	35.4	12.4	1.3
5	18	35.4	15.2	1.3
6	12	33.3	0.0	0.3
7	30	33.3	0.0	0.3
8	36	33.3	0.0	0.3
9	38	33.3	0.0	0.3
10	40	33.3	0.0	0.3
11	42	33.3	0.0	0.3
12	44	33.3	0.0	0.3
13	46	33.3	0.0	0.3
14	48	33.3	0.0	0.3
15	50	33.3	0.0	0.3
16	26	33.3	1.0	0.3
17	10	33.3	11.1	0.3
18	6	33.3	16.7	0.3
19	28	33.3	16.7	0.3
20	32	33.3	16.7	0.3
21	34	33.3	16.7	0.3
22	24	33.3	25.0	0.7
23	8	33.3	66.7	0.3

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.10: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the input gain for the DNFL

		Mean of		
pos	C^{som}	Median %	IQR %	Split %
1	20	56.6	39.9	0.8
2	22	33.3	10.6	0.3
3	16	24.2	6.8	0.4
4	18	23.2	15.4	0.3
5	24	22.2	13.9	0.2
6	14	22.2	17.7	0.2
7	6	11.1	0.0	0.1
8	26	11.1	0.0	0.1
9	28	11.1	0.0	0.1
10	30	11.1	0.0	0.1
11	36	11.1	0.0	0.1
12	38	11.1	0.0	0.1
13	40	11.1	0.0	0.1
14	42	11.1	0.0	0.1
15	44	11.1	0.0	0.1
16	46	11.1	0.0	0.1
17	48	11.1	0.0	0.1
18	50	11.1	0.0	0.1
19	10	11.1	2.8	0.1
20	8	11.1	5.6	0.1
21	32	11.1	5.6	0.1
22	34	11.1	5.6	0.1
23	12	11.1	11.1	0.1

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

D.3 time constant for the SOL

Table D.11: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the time constant for the SOL

		Mean of		
pos	$\tau^{a^{som}}$	Median %	IQR %	Split %
1	600	77.8	9.3	1.0
2	900	77.8	64.9	1.0
3	750	77.8	65.2	1.0
4	450	66.7	64.6	0.7
5	300	35.4	66.9	1.0
6	1200	33.3	65.9	0.3
7	1050	33.3	66.7	0.3

For the default SODNF parameters $\sigma_w = 50$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.12: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the time constant for the SOL

		Mean of		
pos	$\tau^{a^{som}}$	Median %	IQR %	Split %
1	600	51.5	4.4	2.0
2	300	50.8	1.5	1.0
3	1050	50.0	0.2	1.0
4	900	50.0	0.4	1.5
5	450	50.0	1.1	1.0
6	1200	50.0	1.1	1.0
7	750	50.0	5.9	1.5

For the default SODNF parameters $\sigma_w = 50$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.13: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the time constant for the SOL

		Mean of		
pos	$\tau^{a^{som}}$	Median %	IQR %	Split %
1	600	66.7	4.4	1.5
2	900	66.7	16.7	1.5
3	750	66.7	16.7	2.0
4	1050	50.0	0.0	1.0
5	300	50.0	0.2	1.0
6	1200	50.0	0.2	1.0
7	450	50.0	20.8	1.0

For the default SODNF parameters $\sigma_w = 50$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.14: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the time constant for the SOL

		Mean of		
pos	$\tau^{a^{som}}$	Median %	IQR %	Split %
1	600	45.5	4.8	1.3
2	900	44.4	15.9	1.0
3	750	44.4	19.9	0.7
4	300	34.3	17.2	0.7
5	1200	33.3	0.3	0.7
6	1050	33.3	5.3	0.7
7	450	33.3	14.1	0.3

For the default SODNF parameters $\sigma_w = 50$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.15: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the time constant for the SOL

		Mean of		
pos	$\tau^{a^{som}}$	Median %	IQR %	Split %
1	600	34.3	13.1	0.4
2	900	33.3	35.4	0.3
3	750	33.3	41.2	0.3
4	1200	22.2	14.1	0.2
5	300	12.1	6.1	0.2
6	450	11.1	19.7	0.1
7	1050	11.1	27.5	0.1

For the default SODNF parameters $\sigma_w = 50$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$, $c_T = 30$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

D.4 proportionality constant of the iteration time

Table D.16: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the proportionality constant of the iteration time

		Mean of		
pos	c_T	Median %	IQR %	Split %
1	25	100.0	74.7	1.0
2	30	77.8	16.2	1.0
3	15	70.7	97.5	1.0
4	10	65.7	97.5	0.7
5	35	44.4	8.3	0.7
6	50	33.3	16.7	0.3
7	45	33.3	17.4	0.3
8	40	33.3	26.5	0.3
9	20	33.3	41.7	0.3
10	5	0.0	0.0	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.17: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the proportionality constant of the iteration time

		Mean of		
pos	c_T	Median %	IQR %	Split %
1	20	50.0	0.0	1.0
2	35	50.0	0.0	1.0
3	15	50.0	0.0	1.5
4	50	50.0	0.8	0.5
5	10	50.0	0.8	1.0
6	25	50.0	1.1	1.0
7	40	50.0	1.7	0.5
8	45	50.0	2.1	0.5
9	30	50.0	4.9	1.5
10	5	0.0	0.0	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.18: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the proportionality constant of the iteration time

		Mean of		
pos	c_T	Median %	IQR %	Split %
1	35	66.7	0.0	1.0
2	30	66.7	4.2	1.5
3	50	50.0	0.0	0.5
4	15	50.0	0.0	1.5
5	45	50.0	0.2	0.5
6	20	50.0	0.4	1.0
7	25	50.0	4.2	1.0
8	40	50.0	4.4	1.0
9	10	49.2	1.3	1.0
10	5	0.0	0.0	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.19: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the proportionality constant of the iteration time

		Mean of		
pos	c_T	Median %	IQR %	Split %
1	35	44.4	0.0	0.7
2	30	44.4	8.8	1.0
3	15	37.4	41.9	1.0
4	50	33.3	0.5	0.3
5	45	33.3	1.3	0.3
6	25	33.3	3.0	0.7
7	20	33.3	6.1	0.7
8	40	33.3	25.8	0.3
9	10	33.3	28.3	0.7
10	5	0.0	0.0	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.20: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the proportionality constant of the iteration time

		Mean of		
pos	c_T	Median %	IQR %	Split %
1	30	33.3	27.3	0.3
2	15	26.3	30.8	0.3
3	35	22.2	2.8	0.2
4	25	22.2	25.3	0.2
5	50	11.1	0.5	0.1
6	45	11.1	1.3	0.1
7	40	11.1	9.1	0.1
8	10	11.1	19.4	0.1
9	20	11.1	19.9	0.1
10	5	0.0	0.0	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and $r = -0.4$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

D.5 resting level

Table D.21: The best results from the tests with acquired data, ordered according to performance in grouping by position, for the resting level

		Mean of		
pos	r	Median %	IQR %	Split %
1	-0.1	68.7	10.1	1.3
2	-0.2	68.7	37.6	1.7
3	-0.3	67.7	26.3	1.7
4	-0.4	55.6	66.7	0.7
5	-0.5	44.4	41.7	0.7
6	-1	0.0	0.0	0.0
7	-0.9	0.0	0.0	0.0
8	-0.8	0.0	0.0	0.0
9	-0.7	0.0	0.0	0.0
10	-0.6	0.0	52.8	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k}_{\mathrm{bp}} = 1$, $C^{som} = 15$ and$c_T = 30$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.22: The best results from the tests with acquired data, ordered according to performance in grouping by object type, for the resting level

		Mean of		
pos	r	Median %	IQR %	Split %
1	-0.2	53.0	14.4	2.5
2	-0.3	51.5	3.0	2.5
3	-0.1	51.5	6.1	2.5
4	-0.5	50.0	0.0	1.0
5	-0.4	50.0	2.8	1.0
6	-1	0.0	0.0	0.0
7	-0.9	0.0	0.0	0.0
8	-0.8	0.0	0.0	0.0
9	-0.7	0.0	0.0	0.0
10	-0.6	0.0	50.4	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and$c_T = 30$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.23: The best results from the tests with acquired data, ordered according to performance in grouping by grasp type, for the resting level

		Mean of		
pos	r	Median %	IQR %	Split %
1	-0.5	66.7	4.2	1.0
2	-0.1	51.5	16.9	2.5
3	-0.4	50.0	4.4	1.0
4	-0.3	50.0	4.4	2.0
5	-0.2	50.0	26.1	2.5
6	-1	0.0	0.0	0.0
7	-0.9	0.0	0.0	0.0
8	-0.8	0.0	0.0	0.0
9	-0.7	0.0	0.0	0.0
10	-0.6	0.0	66.7	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k}_{\mathrm{bp}} = 1$, $C^{som} = 15$ and$c_T = 30$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th}_{\mathrm{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th}_{\mathrm{disp}} = 0.7$ and $\mathrm{th}_{\mathrm{ignored}} = 0.8$.

Table D.24: The best results from the tests with acquired data, ordered according to performance in grouping by object and grasp types, for the resting level

		Mean of		
pos	r	Median %	IQR %	Split %
1	-0.5	44.4	10.1	0.7
2	-0.4	44.4	37.9	0.7
3	-0.2	39.4	11.1	1.7
4	-0.1	38.4	16.7	1.3
5	-0.3	38.4	23.0	1.7
6	-1	0.0	0.0	0.0
7	-0.9	0.0	0.0	0.0
8	-0.8	0.0	0.0	0.0
9	-0.7	0.0	0.0	0.0
10	-0.6	0.0	44.7	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and$c_T = 30$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

Table D.25: The best results from the tests with acquired data, ordered according to performance in grouping by trajectory type, for the resting level

		Mean of		
pos	r	Median %	IQR %	Split %
1	-0.1	27.3	21.0	0.4
2	-0.2	26.3	17.4	0.4
3	-0.5	22.2	16.7	0.2
4	-0.4	22.2	26.8	0.2
5	-0.3	16.2	28.5	0.4
6	-1	0.0	0.0	0.0
7	-0.9	0.0	0.0	0.0
8	-0.8	0.0	0.0	0.0
9	-0.7	0.0	0.0	0.0
10	-0.6	0.0	25.3	0.0

For the default SODNF parameters $\sigma_w = 50$, $\tau^{a^{som}} = 600$, $\mathrm{k_{bp}} = 1$, $C^{som} = 15$ and$c_T = 30$.

For the analysis parameters $\mathrm{success\,rate} = 0.95$, $\mathrm{eval\,window} = 10$, $p = 0.7$, $\mathrm{th_{eval\,window}} = 0.7$, $Q_{\mathrm{window}} = 10$, $\mathrm{th_{disp}} = 0.7$ and $\mathrm{th_{ignored}} = 0.8$.

This page has been intentionally left blank.

Bibliography

[1] Shun-ichi Amari. 'Dynamics of pattern formation in lateral-inhibition type neural fields'. In: *Biological Cybernetics* 27.2 (June 1977), pp. 77–87. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00337259.

[2] Audi. *New human-robot cooperation in Audi's production processes — Audi MediaCenter*. 2nd Feb. 2015. URL: https://www.audi-mediacenter.com/en/press-releases/new-human-robot-cooperation-in-audis-production-processes-1206 (visited on 21/02/2016).

[3] Suzanna Becker. 'Unsupervised learning procedures for neural networks'. In: *International Journal of Neural Systems* 02.01n02 (26th Feb. 1991), pp. 17–33. ISSN: 0129-0657. DOI: 10.1142/S0129065791000030.

[4] E. Broadbent, R. Stafford and B. MacDonald. 'Acceptance of Healthcare Robots for the Older Population: Review and Future Directions'. In: *International Journal of Social Robotics* 1.4 (3rd Oct. 2009), pp. 319–330. ISSN: 1875-4791, 1875-4805. DOI: 10.1007/s12369-009-0030-6.

[5] Adrian Burton. 'Dolphins, dogs, and robot seals for the treatment of neurological disease'. In: *The Lancet Neurology* 12.9 (Sept. 2013), pp. 851–852. ISSN: 1474-4422. DOI: 10.1016/S1474-4422(13)70206-0.

[6] Eliana Costa e Silva et al. 'Nonlinear Optimization for Human-Like Movements of a High Degree of Freedom Robotics Arm-Hand System'. In: *Computational Science and Its Applications - ICCSA 2011*. Ed. by Beniamino Murgante et al. Red. by David Hutchison et al. Vol. 6784. Lecture Notes in Computer Science.

Berlin, Heidelberg: Springer Berlin Heidelberg, 20th June 2011, pp. 327–342. ISBN: 978-3-642-21930-6 978-3-642-21931-3. DOI: 10.1007/978-3-642-21931-3_26.

[7] M. Ehrenmann et al. 'Programming service tasks in household environments by human demonstration'. In: Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on. IEEE, 2002, pp. 460–467. ISBN: 978-0-7803-7545-1. DOI: 10.1109/ROMAN.2002.1045665.

[8] Shilpa Gulati et al. 'Toward autonomous scientific exploration of ice-covered lakes - Field experiments with the ENDURANCE AUV in an Antarctic Dry Valley'. In: 2010 IEEE International Conference on Robotics and Automation (ICRA). Anchorage, Alaska, USA: IEEE, 3rd–7th May 2010, pp. 308–315. ISBN: 978-1-4244-5040-4 978-1-4244-5038-1. DOI: 10.1109/ROBOT.2010.5509224.

[9] Koji Hashimoto et al. 'A field study of the human support robot in the home environment'. In: 2013 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO). Tokyo, Japan: IEEE, 7th–9th Nov. 2013, pp. 143–150. DOI: 10.1109/ARSO.2013.6705520.

[10] Simon O. Haykin. *Neural Networks and Learning Machines*. 3rd ed. New York: Prentice Hall, 2009. 906 pp. ISBN: 978-0-13-147139-9.

[11] iRobot Corporation. *iRobot: Your Partner for a Cleaner Home*. URL: http://www.irobot.com/ (visited on 06/06/2016).

[12] Samuel Kaski, Jari Kangas and Teuvo Kohonen. 'Bibliography of self-organizing map (SOM) papers: 1981–1997'. In: *Neural computing surveys* 1.3&4 (1998), pp. 102–350.

[13] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Wiley series in probability and mathematical statistics. Hoboken, NJ: Wiley, 2005. 342 pp. ISBN: 978-0-471-73578-6.

[14] Chih-Hung King et al. 'Towards an assistive robot that autonomously performs bed baths for patient hygiene'. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Taipei, Taiwan: IEEE, 18th–22nd Oct.

2010, pp. 319–324. ISBN: 978-1-4244-6674-0. DOI: 10.1109/IROS.2010.5649101.

[15] Teuvo Kohonen. 'The self-organizing map'. In: *Proceedings of the IEEE* 78.9 (Sept. 1990), pp. 1464–1480. ISSN: 0018-9219. DOI: 10.1109/5.58325.

[16] Clayton Kunz et al. 'Toward extraplanetary under-ice exploration: Robotic steps in the Arctic'. In: *Journal of Field Robotics* 26.4 (1st Apr. 2009), pp. 411–429. ISSN: 1556-4967. DOI: 10.1002/rob.20288.

[17] Carlo R. Laing et al. 'Multiple Bumps in a Neuronal Model of Working Memory'. In: *SIAM Journal on Applied Mathematics* 63.1 (2002), pp. 62–97. ISSN: 0036-1399. DOI: 10.1137/S0036139901389495.

[18] Yugang Liu and Goldie Nejat. 'Robotic Urban Search and Rescue: A Survey from the Control Perspective'. In: *Journal of Intelligent & Robotic Systems* 72.2 (12th Mar. 2013), pp. 147–165. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-013-9822-x.

[19] Luís Louro. 'Controlo e representações para tarefas de colaboração humano-robô'. PhD thesis. Universidade do Minho, 2nd July 2010.

[20] Microsoft. *Buy Kinect Adapter for Windows - Microsoft Store*. URL: http://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-Adapter-for-Windows/productID.308803600 (visited on 30/03/2016).

[21] Microsoft. *Kinect hardware*. URL: https://developer.microsoft.com/en-us/windows/kinect/hardware (visited on 21/04/2016).

[22] Wendy Moyle et al. 'Social robots helping people with dementia: Assessing efficacy of social robots in the nursing home environment'. In: 2013 The 6th International Conference on Human System Interaction (HSI). Sopot, Poland: IEEE, 6th–8th June 2013, pp. 608–613. ISBN: 978-1-4673-5637-4 978-1-4673-5635-0. DOI: 10.1109/HSI.2013.6577887.

[23] Robin R. Murphy et al. 'Search and Rescue Robotics'. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano Prof and Oussama Khatib Prof. Springer Berlin Heidelberg, 2008, pp. 1151–1173. ISBN: 978-3-540-23957-4 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_51.

[24] Neato Robotics, Inc. *Robot Vacuums*. URL: https://www.neatorobotics.com/robot-vacuums/ (visited on 06/06/2016).

[25] Illah R. Nourbakhsh et al. 'Human-robot teaming for search and rescue'. In: *IEEE Pervasive Computing* 4.1 (Jan.–Mar. 2005), pp. 72–79. ISSN: 1536-1268. DOI: 10.1109/MPRV.2005.13.

[26] Merja Oja, Samuel Kaski and Teuvo Kohonen. 'Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum'. In: *Neural computing surveys* 3.1 (2003), pp. 1–156.

[27] Joelle Pineau et al. 'Towards robotic assistants in nursing homes: Challenges and results'. In: *Robotics and Autonomous Systems*. Socially Interactive Robots 42.3–4 (31st Mar. 2003), pp. 271–281. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(02)00381-0.

[28] Manuel Pinheiro, Estela Bicho and Wolfram Erlhagen. 'A dynamic neural field architecture for a pro-active assistant robot'. In: 2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). Tokyo, Japan: IEEE, 26th–29th Sept. 2010, pp. 777–784. ISBN: 978-1-4244-7709-8 978-1-4244-7708-1. DOI: 10.1109/BIOROB.2010.5627812.

[29] Matti Pöllä, Timo Honkela and Teuvo Kohonen. 'Bibliography of self-organizing map (SOM) papers: 2002-2005 addendum'. In: *Neural Computing Surveys* (2009).

[30] Sony Corporation. *Sony Global - Press Release - Sony Launches Four-Legged Entertainment Robot "AIBO" Creates a New Market for Robot-Based Entertainment*. URL: http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046/ (visited on 06/06/2016).

[31] Emanuel Augusto Freitas de Sousa. 'On Learning and Generalizing Representations in Dynamic Field Based Architecture for Human-Robot Interaction'. PhD thesis. Universidade do Minho, 10th Apr. 2015.

[32] Miguel Sousa et al. 'Multi-robot cognitive formations'. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 7th–12th Oct. 2012, pp. 545–550. ISBN: 978-1-4673-1736-8 978-1-4673-1737-5 978-1-4673-1735-1. DOI: 10.1109/IROS.2012.6385833.

[33] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd edition draft. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, Oct. 2015. 356 pp. ISBN: 978-0-262-19398-6 978-0-262-33276-7.

[34] Toyota Motor Corporation. *Toyota Shifts Home Helper Robot R&D into High Gear with New Developer Community and Upgraded Prototype — TOYOTA Global Newsroom*. 16th July 2015. URL: http://newsroom.toyota.co.jp/en/detail/8709541/ (visited on 11/02/2016).

[35] World Health Organization, ed. *World report on ageing and health*. Geneva, Switzerland: World Health Organization, 2015. 246 pp. ISBN: 978-92-4-069481-1 978-92-4-156504-2 978-92-4-069479-8 978-92-4-069480-4.

[36] WowWee®. *WowWee® Robots*. URL: http://wowwee.com/products/robots (visited on 06/06/2016).

[37] Tingfang Yan et al. 'Review of assistive strategies in powered lower-limb orthoses and exoskeletons'. In: *Robotics and Autonomous Systems* 64 (Feb. 2015), pp. 120–136. ISSN: 0921-8890. DOI: 10.1016/j.robot.2014.09.032.