



VRIJE  
UNIVERSITEIT  
BRUSSEL

FACULTY OF ENGINEERING  
DEPARTMENT OF MECHANICAL ENGINEERING

# A generic gesture method for social robots: development, validation and use in robot design

ir. Greet Van de Perre

March 2018

Supervisor: Prof. dr. ir. Bram Vanderborght  
Co-supervisor: Prof. dr. ir. Dirk Lefeber



# Jury Members

**Chairman**

Prof. dr. Ann Nowé  
*Dept. of Computer Science*  
*Vrije Universiteit Brussel, Belgium*

**Vice-chairman**

Prof. dr. Roger Vounckx  
*Dept. of Electronics and Informatics*  
*Vrije Universiteit Brussel, Belgium*

**Secretary**

dr. ir. Pablo Gómez Esteban  
*Dept. of Mechanical Engineering*  
*Vrije Universiteit Brussel, Belgium*

**External members**

Prof. dr. ir. Ana Paiva  
*Dept. of Computer Science and Engineering*  
*University of Porto, Portugal*

Prof. dr. Tony Belpaeme  
*School of computing, Electronics and Mathematics*  
*Plymouth University, Great Britain*  
  
*Dept. of Electronics and Information Systems*  
*Universiteit Gent, Belgium*

Prof. dr. ing. Jelle Saldien  
*Dept. of Industrial Systems Engineering*  
*and Product Design*  
*Universiteit Gent, Belgium*

**Advisor** Prof. dr. ir. Bram Vanderborght  
*Dept. of Mechanical Engineering*  
*Vrije Universiteit Brussel, Belgium*

**Co-advisor** Prof. dr. ir. Dirk Lefeber  
*Dept. of Mechanical Engineering*  
*Vrije Universiteit Brussel, Belgium*

# Acknowledgements

Several years after starting my PhD, I dare say I'm proud of the result when browsing through this document. Since I would not have been able to do all of this on my own, I would like to thank everyone who supported and helped me during the past years.

First of all, thanks to my supervisors Bram and Dirk, for giving me the opportunity to start a PhD at the Robotics and Multibody Mechanics Research Group. Thank you Bram, for your *incredibly* fast review of every single word I've put on paper. I think the mean time between sending you the draft and receiving back the comments was about 5 hours. Which, by the way, includes manuscripts that were e-mailed after 8 PM. Getting valuable feedback so fast really allowed me to work efficiently when it was crucial. Next, I would like to thank the FWO for giving me the financial support for my research.

Thanks to my colleagues, for all the interesting discussions, advise, help and of course, the nice moments besides the work. In specific, thanks to Michaël, for introducing me to the principles of robot kinematics and advising me in the start-up of the gesture method. Regarding the design and development of the arm system, my thanks goes to Albert, for sharing his large experience in robot design and for making the neck module for Elvis. Many thanks to Raphael for his help in the never-ending problems related to the 3D printing. There were some moments - ok, *a lot* of moments - where I wanted to throw the printer out of the window. Thanks for preventing that this never actually happened. Thank you Long, for providing the software to control Elvis, and your help in different experiments. Thanks to Karen, Stefanie, Laura and Romy, for the nice and relaxing coffee breaks and to Tom, Glenn, Pablo, Maarten, Joe and Elahe for enduring me in the office. Finally, I should not forget to thank our technicians Marnix, Marc, Geoffrey and Jean-Paul and secretaries Jenny and Birgit.

Thanks to Ramo for brightening up the VUB-life. I loved our lunches and - most importantly - cake-moments, which, occasionally, could last somewhat longer than planned. I'm happy to have such a good friend to talk about our amazing kids, sometimes *less* amazing kids, to complain about our better

halves and to encourage each other in our work.

Next, I would like to thank my family and friends for their support and interest in my work. Thanks for helping me think about a good name for my big fluffy elephant, especially to my mother, who came up with the winning suggestion *Elvis*.

Tim, our story started at VUB and will continue for a long, long time. During this PhD, our two wonderful kids were born. The fact that I successfully ended this thesis in combination with your full-time job and the chaos caused by two kids proves that we're a good team. Thanks Emelien, my little princess, for being the first guinea pig in interacting with Elvis. You immediately loved that 'big elephant' and kept talking and playing with it, even when it was standing unfinished at our living room table. And finally, thanks to Nolan, my sweet cuddle monster, to be so caring about my time management. I'm convinced you had the best intentions by keeping me up for hours at night, making sure I wouldn't waste my precious time to something so unnecessary as sleeping. Unfortunately, you underestimated the effects of chronic sleep deprivation. Anyway, now I'm writing down these final words to conclude my PhD, I'm happy to release you from this heavy task and wish you a nice, long night of sleep.



# Abstract

Since social robots are aimed to be operated by untrained users, interacting with this type of robots should be very intuitive and natural. Since nothing is more intuitive than our own communication skills, this new generation of robots should be able to use and understand speech, facial expressions and/or body language. These requirements have a direct impact on the design parameters. Together with the aim of social robots to be used in our daily lives, which implies they need to be adapted to our environments and tools, this typically results in robot designs with main human characteristics. Many social robots are therefore humanoid robots. The fact that these robots have similar common outer features, does not imply that the internal mechanics are similar. Different robot arms can have different degrees of freedom, with various joint configurations. The difference in joint configuration makes that motion patterns defined for a certain robot cannot be easily transferred to others. This issue is known as the correspondence problem. When imitating, copying, mimicking or learning from an agent, a correspondence between the demonstrator and imitator needs to be specified. When the agents have similar bodies, the mapping is obvious, however, when using agents with significantly different morphologies, this can become a difficult task. Therefore, in robotics, the correspondence problem is often omitted by coding the gestures for one specific robot configuration. Sharing gestures between robots is not straightforward and therefore, when working with a new robot platform, new joint trajectories to reach the desired postures need to be calculated and implemented.

In this thesis, we aim to provide a solution for the correspondence problem and make the implementation of gestures more efficient by developing a generic method to generate gestures for social robots. The innovative aspect of this method is that it is constructed independently of any robot configuration, instead, a human base model was used as a reference to construct the method's framework. To calculate gestures for a desired configuration, a limited set of morphological information, inputted by the user, is used to evaluate the generic framework. Since for different types of gestures, different features are important, our method was designed to work in two modes. The block mode is

used to calculate gestures whereby the overall arm placement is crucial, like for emotional expressions. The end effector mode, on the other hand, is developed for end-effector depending gestures, i.e. gestures whereby the placement of the end-effector is important, like for manipulation and pointing. For the latter type of gestures, the method provides the possibility of mood expression by modulating the functional behavior into an affective gesture, using a set of modification parameters. Furthermore, a mode mixer was implemented to allow gestures calculated by the two modes to be combined into one blended gesture. The different aspects of the gesture method were validated on both the virtual model as the physical model of different humanoid robots, including NAO, ASIMO, Pepper and Romeo.

In a next step, the developed gesture method was used as a novel tool in the design process of social robots. Since gestures can be calculated for any desired robot configuration with minimal effort by the programmer, the effect of different design aspects on a series of postures can be studied by generating a selected set of gestures for different morphologies and visualizing them on a single virtual model. The gesture method proves its usefulness in the design process of social robots by providing insights in the influence of specific joints, their collocation and joint angle range, helping the designer to make substantiated trade-off's in the design process.

As an ultimate validation of both aspects of the method, namely its usefulness in generating gestures for a random robot configuration, as well as its applicability in the design process of a social robot, a new version of the social robot Probo was developed. The resulting robot, called Elvis, has an actuated arm system, which was constructed semi-modular in order to allow for different configurations to be studied. The joint configurations, as well as the exact design of the different joint modules, resulted from an a-priori gesture study using the developed software. Three morphologies, called Elvis-Ca, Elvis-Cb, and Elvis-Cc, were physically realized. To conclude, gestures were successfully generated for all three Elvis configurations using the generic gesture method.

# Nomenclature

## Abbreviations

<i>AU</i>	Action Unit
<i>ASD</i>	Autism Spectrum Disorder
<i>BAU</i>	Body Action Unit
<i>CLIK</i>	Closed Loop Inverse Kinematics
<i>CGI</i>	Computer Generated Imagery
<i>DOF</i>	Degree Of Freedom
<i>DH</i>	Denavit-Hartenberg
<i>GUI</i>	Graphical User Interface
<i>HRI</i>	Human-Robot Interaction
<i>JRA</i>	Joint Range Availability
<i>MP</i>	Minimum Posture
<i>RAT</i>	Robot Assisted Therapy
<i>R&amp;MM</i>	Robotics and Multibody Mechanics
<i>SDK</i>	Software Development Kit
<i>val</i>	Valence

## Symbols

$a$	DH-parameter 1; link length
$\alpha$	DH-parameter 2; link twist
$d$	DH-parameter 3; link offset
$\theta$	DH-parameter 4; joint angle
$J_A$	Analytical Jacobian
$q$	Joint angle
$q_i$	Angle of joint $i$
$q_{max}$	Maximum joint angle
$q_{min}$	Minimum joint angle
$q_m$	Minimum posture angle
$q_a$	Affective posture angle

$R$	Rotation matrix
$v$	Motion speed
$x$	End-effector pose
$x_d$	Desired end-effector pose
$x_e$	Actual end-effector pose
$z$	Transformed joint angle

# Contents

<b>Jury Members</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Body language in human communication . . . . .	2
1.2.1 Nonverbal communication . . . . .	2
1.2.2 Classification of Body language and gestures . . . . .	2
1.2.3 Important nomenclature used in this thesis . . . . .	5
1.3 Non-verbal communication in human-robot interaction . . . . .	6
1.3.1 Gestures for natural communication . . . . .	6
1.3.2 Implementing gestures for robots . . . . .	10
1.3.2.1 The correspondence problem . . . . .	10
1.3.2.2 Coping with the correspondence problem . . . . .	11
1.4 Objectives . . . . .	13
1.5 Thesis outline . . . . .	14
<b>I Development of a generic gesture method for social robots</b>	<b>15</b>
<b>2 Methodology</b>	<b>17</b>

2.1	Working modes of the gesture method . . . . .	18
2.2	Achieving the generic aspect of the method . . . . .	20
2.2.1	Motion of the human body . . . . .	20
2.2.2	Body Action Units and the human base model . . . . .	23
2.3	Specifying the robot's configuration . . . . .	23
2.4	Conclusions . . . . .	27
<b>3</b>	<b>Generating emotional expressions using the block mode</b>	<b>29</b>
3.1	Methodology . . . . .	30
3.2	Target gestures . . . . .	30
3.3	Mapping the gestures to a configuration . . . . .	31
3.3.1	Complete configuration . . . . .	35
3.3.2	Configuration with a reduced number of DOF . . . . .	36
3.4	Results . . . . .	38
3.4.1	Emotional expression of <i>happiness</i> for different configurations . . . . .	38
3.4.2	Survey . . . . .	39
3.5	Conclusions . . . . .	41
<b>4</b>	<b>Calculating reaching and pointing gestures using the end-effector mode</b>	<b>47</b>
4.1	<i>Place-at</i> condition . . . . .	48
4.1.1	Calculating a posture for a specified end-effector position . . . . .	48
4.1.2	Natural postures . . . . .	49
4.2	Range of the robot . . . . .	49
4.2.1	Approximation of the workspace . . . . .	49
4.2.2	Evaluation of specified end-effector positions . . . . .	53
4.3	Trajectory generation . . . . .	54
4.4	<i>Pointing</i> condition . . . . .	55
4.5	Gazing . . . . .	56
4.6	Results . . . . .	60
4.6.1	Results for the Place-at condition . . . . .	60
4.6.2	Results for the Pointing condition . . . . .	60
4.6.3	Place-at condition imposed on different configurations . . . . .	61
4.7	Conclusions . . . . .	64

<b>5 Generating blended gestures and affective functional behaviors</b>	<b>65</b>
5.1 Blended gestures . . . . .	66
5.1.1 Priority levels . . . . .	66
5.1.2 Examples of blended gestures . . . . .	68
5.2 Affective functional behaviors . . . . .	69
5.2.1 Expressivity models . . . . .	69
5.2.2 Generating affective gestures by influencing the motion speed . . . . .	70
5.2.3 Generating affective postures using the nullspace . . . . .	70
5.2.4 Example: deictic gesture during different states of affect	72
5.3 Conclusions . . . . .	72
<b>6 Generating gestures for physical robots</b>	<b>75</b>
6.1 Adjusting the method for physical robots . . . . .	76
6.1.1 Joint angle limits . . . . .	76
6.1.1.1 Determination of the optimal task vector $y$ for our method . . . . .	77
6.1.2 Joint speed limits . . . . .	82
6.2 Experimental results on physical robots . . . . .	84
6.3 Conclusions . . . . .	95
<b>II Designing social robots using a generic gesture method</b>	<b>97</b>
<b>7 Studying design aspects using a generic gesture method</b>	<b>99</b>
7.1 Introduction . . . . .	100
7.2 Methodology . . . . .	102
7.2.1 Effects of joint configuration on a set of emotional expressions . . . . .	106
7.2.2 Effects of joint angle limits on a set of emotional expressions	110
7.3 Conclusion . . . . .	111
<b>III Elvis, a new social robot</b>	<b>115</b>
<b>8 Design of Elvis; a new social robot</b>	<b>117</b>

## CONTENTS

8.1	The Probo-project . . . . .	118
8.2	Elvis . . . . .	120
8.3	Joint configurations . . . . .	120
8.3.1	Elvis-Ca . . . . .	121
8.3.2	Elvis-Cb . . . . .	121
8.3.3	Elvis-Cc . . . . .	127
8.4	Design principles . . . . .	127
8.5	Design of the different modules . . . . .	128
8.5.1	Shoulder . . . . .	128
8.5.2	Elbow . . . . .	128
8.5.3	Wrist . . . . .	130
8.5.4	Hand module . . . . .	130
8.6	Elvis's new look . . . . .	132
8.7	Conclusions . . . . .	132
<b>9</b>	<b>Generating motions for Elvis</b>	<b>135</b>
9.1	Controling the robot . . . . .	136
9.2	Results . . . . .	136
9.2.1	Uncovered model . . . . .	136
9.2.2	Covered model . . . . .	137
9.3	Conclusions . . . . .	142
<b>IV</b>	<b>Conclusions</b>	<b>145</b>
<b>10</b>	<b>Conclusions</b>	<b>147</b>
10.1	A generic gesture method for social robots . . . . .	148
10.1.1	Development of the gesture method . . . . .	148
10.1.2	Validation of the developed software . . . . .	149
10.1.3	Limitations and possible improvements of the method .	149
10.2	Use of the method for design purposes . . . . .	151
10.3	Elvis . . . . .	152
10.3.1	Development of the robot Elvis . . . . .	152
10.3.2	Limitations and possible improvements of Elvis . . . . .	152
<b>A</b>	<b>List of publications</b>	<b>155</b>

<b>B Calculation details</b>	<b>161</b>
B.1 Direct kinematics . . . . .	161
B.1.1 Denavit-Hartenberg convention . . . . .	161
B.1.2 Calculation of the end-effector pose . . . . .	163
B.2 Interpolation between postures . . . . .	164
B.2.1 Interpolation in the block mode . . . . .	164
B.2.2 Interpolation in the end-effector mode . . . . .	165
B.3 Inverse kinematics . . . . .	166
B.3.1 Geometric Jacobian . . . . .	167
B.3.2 Calculation of the analytical Jacobian . . . . .	168
B.4 Runge-Kutta . . . . .	169
<b>Bibliography</b>	<b>171</b>



# 1

# Introduction

## 1.1 Motivation

The focus of communication in robotics research made a major switch the past decades. Traditionally, robots are designed for industrial applications, with specific tasks to be performed individually. There is no question of cooperation with humans. On the contrary, for safety reasons, they are mostly kept in cages to avoid contact with humans. Communicating with these robots is typically achieved by trained operators using touch screens or other input devices. Social robots, on the other hand, are aimed to work side by side with humans, in numerous tasks in our daily life. Since this new generation of robots is aimed to be operated by untrained users, including children, elderly people, therapists and teachers, interacting with this type of robots should be very intuitive and natural. Since nothing is more intuitive than our own communication skills, social robots should be able to use and understand speech and non-verbal communication skills, such as facial expressions and gestures. These requirements have a direct impact on the design parameters. Together with the aim of social robots to be used in our daily lives, implying they need to be adapted to our environments and tools, this typically results in robot designs with main human characteristics. Many social robots are therefore humanoid robots. The fact these robots have similar common outer features does not imply that the internal mechanics are similar. Different robot arms can have different degrees of freedom (DOF), with various joint configurations. Figure 1.1 illustrates this; three humanoid robots with different joint configurations are shown. Figure 1.1a shows the humanoid ASIMO [1]. ASIMO's arm contains 7 DOF, whereof 3 are responsible for the shoulder movement, 1 for the elbow flexion/extension and another 3 for the wrist motion. Figure 1.1b shows the robot Justin [2]. Like ASIMO, Justin's arm contains 7 DOF. The joints are however positioned in a different way. While for ASIMO, different joints are grouped more or less into a joint complex for the shoulder and wrist, Justin's joints are all separated from each other by links, except for the two latter wrist joints. The robot

NAO (figure 1.1c) on the other hand, has an arm only consisting of 5 DOF. In contrast to the two previous robots, only 1 joint is positioned in the wrist zone.

This difference in joint configuration makes that motion patterns defined for a certain robot cannot be easily transferred to others. This issue is known as the correspondence problem [4][5]. When imitating, copying, mimicking or learning from an agent, a correspondence between the demonstrator and imitator needs to be specified by identifying a correct mapping between the two agents. When the agents have similar bodies, the correspondence is obvious, however, when using agents with significantly different morphologies, this can become a difficult task. Therefore, in robotics, the correspondence problem is often omitted by coding the gestures for one specific robot configuration. When working with a new robot platform, new joint trajectories to reach the desired postures need to be calculated and implemented. A generic method, however, to generate gestures for different robot morphologies can be useful for different research teams since it allows gestures to be shared between different robots and minimizes the workload when implementing gestures on a new robot platform.

## 1.2 Body language in human communication

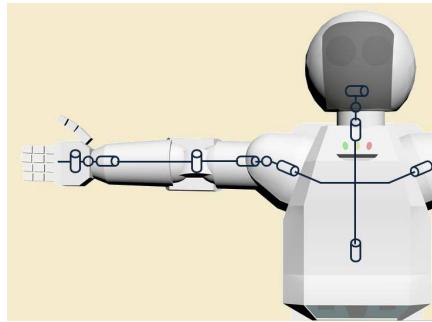
### 1.2.1 Nonverbal communication

Nonverbal communication is a crucial feature in human interaction. Facial expressions, body posture and gestures all convey information about a person's internal state, and contribute to the overall effectiveness of communication. At least 60 percent of our total communication is realized nonverbally [6][7]. It includes both conscious as non-conscious signs and reveals much information about our personality, thoughts and feelings [8]. Multiple nonverbal systems contribute to human communication. The tone of our speech, the way we use space, our posture and our eye behavior are only a few examples of nonverbal cues that shapes how a person is perceived by others. In this thesis, we are interested in kinesics; the language of body position and movement, or shortly, body language [6].

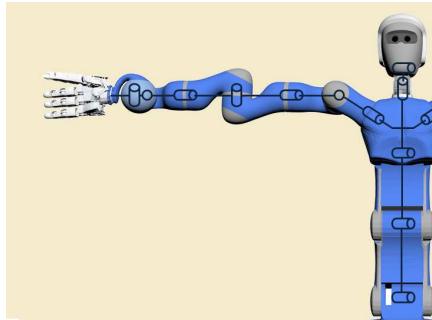
### 1.2.2 Classification of Body language and gestures

Ekman and Friesen's classification of body language is based on the work of Efron [9] and distinguishes five categories [10][11]:

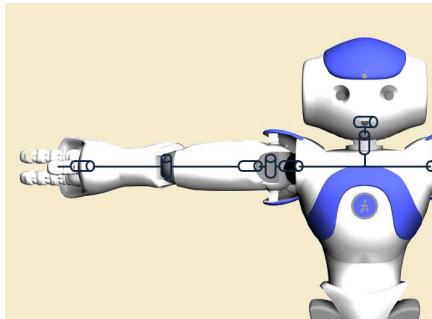
- **Emblems** have a set of precise meanings. They are socially learned and therefore, culturally variable. Emblems can repeat a word in a conversa-



(a)



(b)



(c)

Figure 1.1: A similar outer appearance does not imply similar internal mechanics. Three robots with different joint configuration are shown. (a) ASIMO [1]. (b) Justin [2]. (c) NAO [3].

tion, replace it, provide a separate comment related to the words said, or occur in the absence of speech. A commonly used emblem, for example, is the *thumbs up* sign, used to express approval or encouragement.

- **Manipulators**, also called **adaptors**, or **touching behavior** [12], are touching movements typically associated with internal states related to arousal, anxiety or stress. They can be targeted toward the own body, objects, or other persons. Examples are twisting the hair, tapping a pen and scratching the nose. Manipulators are movements that are performed on the edge of personal awareness and may thus serve unintentionally as clues to how a person is feeling.
- **Illustrators** are movements that illustrate speech. They are intimately related to what is verbally said, usually augmenting the spoken words. Seven types of illustrators were defined by Ekman and Friesen:
  - **Batons** are used to emphasize a particular word or phrase. They are closely coordinated with speech and do not convey any semantic content.
  - **Ideographs** are hand gestures that trace or sketch the speaker's direction of thought. They are related to the logical structure of the conversation
  - **Deictic** gestures, or **pointing** gestures point or refer to a person, object or place.
  - **Kinetographs** are hand gestures that depict a physical action. They mimic the action being described. An example is stretching the thumb and the little finger of a fist to mimic talking on the phone.
  - **Spatial illustrators** depict a spatial relationship between different objects or persons referred to in the conversation.
  - **Pictographs** are gestures that sketch what the speaker is referring to.
  - **Rhythmic movements** illustrate the rhythm or pacing of an event.
- **Regulators** are acts that have the purpose to regulate the flow of a conversation by helping to initiate and terminate the speech of different participants. Nodding the head during a conversation, for example, can indicate understanding, and encourages the speaker to continue.
- **Emotional expressions**, also called **affect displays**, are signals that display our internal affective state. They are mostly involuntary signals, and thus often occur spontaneously. This category includes for example bouncing to express happiness or raising the arms in front of the face when expressing fear.

In their categorisation, Ekman and Friesen do not specify the term *gesture*. In our daily use, the word *gesture* refers to a movement of the hands, face or other body part to express an idea, meaning or feeling<sup>1</sup>. In this perspective, all of the five described categories can be seen as gestures. Some researchers, however, maintain a more restricted meaning of the word. Kendon uses the term *gesture* to refer to body movements that are made to communicate particular messages and that can be used in conjunction with, or in place of speech [13]. Therefore, emotional expressions and manipulators are in his eyes not considered as gestures [14]. In his work, Kendon focuses on gestures found in spoken interaction, gestures that are not interpretable in the absence of speech. This is what Ekman and Friesen classified as *illustrators*. McNeill is interested in the same category [15]. His classification is less detailed than that of Ekman and Friesen, and distinguishes four main categories [16]:

- **Iconic gestures** display aspects of a concrete event, person or object described in the conversation.
- **Metaphoric gestures** present an image of an abstract concept.
- **Deictic gestures** are pointing movements, as discussed above.
- **Beat gestures** are rhythmic movements that mark the speaker's conception of the discourse. A beat has no own semantic content but emphasizes important aspects of the conversation.

Each of these categories coincides with one or more illustrator types defined by Ekman and Friesen. Table 1.1 lists the correspondences.

Knapp [12] uses a similar interpretation of the term *gesture* and distinguishes two categories;

- **Speech independent gestures**, coincident with the *emblems* defined by Ekman and Friesen.
- **Speech related gestures**, coincident with the *illustrators* defined by Ekman and Friesen.

### 1.2.3 Important nomenclature used in this thesis

In this thesis, we will use the categorisation of Ekman and Friesen. The term *gesture* is used to depict a wide range of hand, face, or body movements. In addition, the term *emotional expression* is used to describe an explicit, full body action representing an internal emotional state. An *affective gesture*, on

---

<sup>1</sup><https://en.oxforddictionaries.com>, <http://dictionary.cambridge.org>

Table 1.1: Correspondence between the gesture categories defined by McNeill and Ekman and Friesen’s illustrator types (adapted from [16]).

<b>Ekman and Friesen</b>	<b>McNeill</b>
kinetographs pictographs	iconics
ideographs spatialis	metaphorics
deictics	deictics
batons rhythemics	beats

the other hand, denotes another type of gesture, such as an illustrator gesture or emblem, that is performed in such a way that emotional content is conveyed in addition to the meanings of the gesture. Important features are hereby, amongst others, the speed at which the gesture is performed and the amplitude of the postures. Affective gestures are a subset of a broad collection of affective functional behaviors, which contains, next to gestures, other initially neutral behaviors such as walking and grasping that are modulated into affective motions.

## 1.3 Non-verbal communication in human-robot interaction

### 1.3.1 Gestures for natural communication

The design of social robots is a challenging task. In contrast to classical industrial robots, which can be more thought at as tools, social robots are aimed to interact with people in an interpersonal manner [17]. They are used to achieve social or emotional goals in different applications, such as education [18][19], communication [20][21][22], collaboration [23][24][25] and health [26][27][28]. An intensively explored research field is the use of robots to ameliorate or enhance different types of therapy, called robot assisted therapy (RAT). One of the promising applications is the use of RAT for autism therapies. The robot Probo, for example, designed at the Robotics and Multibody Mechanics

Research (R&MM) Group of the Vrije Universiteit Brussel to study human-robot interaction with children, was used for several experiments in this field [29][30][31]. Also other robots have been used in this research area [32][33]. Another interesting research project running at the R&MM group is the EU-project DREAM, whereof one of the challenges lies in building a complete platform-independent cognitive architecture. The capabilities of this architecture are to be evaluated in RAT with children with Autism Spectrum Disorder (ASD). While the method discussed in this PhD thesis offers the flexibility of automatically creating gestures for different robots, the DREAM project allows to extend this flexibility of changing between robot platforms for a complete experimental protocol [34][35][36].

For humans and robots to be able to work closely together in every day settings, it is important to ensure a natural, intuitive interaction. Instead of controlling a robot with a keyboard, touch screen or similar input device, using human communication skills to guide the robot towards a task would be more interesting and intuitive for all kinds of users. Therefore, social robots need to be able to communicate using both verbal and nonverbal signs.

The effectiveness of using gestures for a fluent and natural human-robot interaction was investigated by different research teams. In [37], a gesturing robot was perceived as having a higher level of conversation proficiency than a robot using speech only. Furthermore, the use of gestures appeared to have a positive effect on the familiarity and human-likeness of the robot. The positive effect of gestures on the likability and perceived anthropomorphism of a robot was also investigated by Salem et al. [38], while Breazeal et al. found promising evidence that nonverbal communication does not only enhances the likeability of robots, but also improves the effectiveness of human-robot teamwork [39]. Similarly, an experimental study conducted by Elprama et al. [23] indicated that factory workers are more eager to cooperate with a robot exhibiting social cues.

Numerous research projects have been set up to investigate different aspects of gesturing in HRI. The recognition of human gestures by robots comprises a large domain. However, here, we focus on gestures performed by robots, and specifically, how they are implemented.

## Emblems

In [40], the use of emblematic gestures was investigated. A number of emblems were generated by a human and the humanoid robot NAO and the perception of both gesture sets was studied. Nine frequently used emblems were selected. To generate the gestures for the robot, several representative key poses were selected from the human's gesture and mapped to the robot's configuration.

## Regulators

Chao et al. [41] developed an architecture to control turn-taking in social human-robot interaction, named CADENCE, and validated it on the robot Simon (Meka Robotics). The implemented robot actions are speech, manipulation, gaze and gestures. The gestures are pre-recorded animations resulting from human motion capture. To enable a range of social dynamics for the human-robot interaction, the model is parametrized. By manipulating the turn-taking parameters, different robot behaviors can be obtained. As a result, different personalities are attributed to the robot by the interaction partner.

Yokoyama et al. [42] focused on the output timings of regulators. In a first step, the tendency of output timings in human interaction was analysed by monitoring the timings at which the nonverbal cues appear in reference to the start of the utterance. Afterwards, the use of different types of regulators and a variety of output timings were validated during a human-robot interaction.

## Illustrators

In the category *illustrators*, most of the performed research is related to the use of deictic gestures. Human interaction experiments performed by Gullberg [43] indicated that deictic illustrators are mostly used (56%), followed by iconics (23%), metaphorics (11%) and beats (10%). This distribution was largely confirmed by Allen [44], who experimentally found that deictic illustrators are mostly used (68%), followed by iconics (20%), beats (11%), and metaphorics (2%). Furthermore, deictic gestures are important features in establishing joint attention, a crucial skill for language development and the theory of mind in children [45][46]. Considering their importance, it is not remarkable that this type of illustrator gains most attention in human-robot interaction research.

In [47], the perception of robotic deictic gestures was studied. Four types of deictic signals were implemented on the robot Bandit, namely pointing with the head, with a straight arm, with a bent arm, and a combined head-arm pointing gesture. As can be expected, pointing with synchronized combined modalities appeared to outperform the single modalities. Furthermore, the physical characteristics of the gesture (pointing with a straight or bent arm) were proven to have an important influence on the perceptual accuracy.

Sauppé and Muthu [48] investigated the communicational effectiveness of several deictic gestures in different settings. Six types of gestures, namely pointing, presenting, touching, exhibiting, grouping and sweeping, were performed by NAO. To implement the gestures, the puppeteering technique was used. Here, the robot is manually put in position by the designer, while the joint angles are saved by a capturing program at several keyframes. These keyframes are then used to generate arm-motion trajectories, which are replayed during

interaction.

In [49], the use of deictic gestures in giving route directions was studied. The effectiveness of route directions given by the robot Robovie was studied in a setting with and without the use of pointing gestures. Next to an increase in the correctness of the interpreted directions when using gestures, also the subjective evaluation of easiness and naturalness was rated higher.

### Emotional expressions

People may wonder why it is necessary for robots to express emotions. While deictic gestures, for example, are obviously "functional" gestures, the usefulness of implementing emotional expressions may not be straightforward. The primary reason of implementing emotion in robots, is that it helps in creating a believable human-robot interaction [50][51]. It is a necessary feature for robots to be socially accepted [52]. Expressing emotions helps in creating an illusion of life in robots, and makes users attribute a personality and feelings to the robot [53]. Furthermore, emotion provides feedback to the user, indicating the internal state, goals and intentions of the robot [54]. They contribute to the transparency of the robot's behavior, and as such, facilitate the human-robot interaction [17]. A robot that uses expressive cues, next to speech and speech accompanying gestures, is able to create a more cooperative relationship with a human [55]. In addition, results indicate that, if a robot has a compelling personality, people are more willing to interact and establish a relationship with it [56][57]. Numerous robots with facial expression modalities have been developed, such as Kismet [58], Probo [59], ROMAN [60] and WE-4RII [61]. Next to facial expressions, also body language has proven to be useful in expressing emotion [62][63][64].

The robot WE-4RII [61][65] was developed to study human-like emotion. The robot features a head with 24 DOF to successfully generate facial expressions. Furthermore, the robot uses its waist, neck and 9 DOF arms to output emotional expressions. Motion patterns were implemented for the six basic emotions; happiness, sadness, disgust, fear and surprise and evaluated through user studies.

In [64], the emotional expressiveness of the robot KOBIAN is discussed. KOBIAN's head is based on that of WE-4RII, but only contains 7 DOF. In a first experiment, the difference in perception of body emotion expression, facial expression and the combination of the two features is investigated. Results showed that the expressiveness significantly increases when using a combination of body and facial expression. Since for the same emotion, people use different bodily expressions, in a second experiment, multiple motion patterns for the expression of seven emotions (happiness, sadness, fear, disgust, anger, surprise and perplexity) were created and validated. Next to the original pat-

terns created by the researchers itself, additional expressions were created by a photographer and a cartoonist. Both artists selected an optimal posture representing the seven emotions. In a next step, a professional actor was asked to replicate these emotional expression. These movements were recorded and used to generate corresponding motion patterns for the robot.

Beck et al. [52] investigated the effect of the head position on the recognizability of emotional expressions. Six emotions were selected; anger, sadness, fear, pride, happiness and excitement. Expressive key poses were constructed for NAO by using motion capture data from a professional actor performing the expressions. The experimental poses were created by systematically altering the head positions for the six key poses. Results showed that the head position has a strong effect on the perception of the poses. Lowering the head leads to a decrease in perceived arousal and valence, while moving the head up increases the dimensions. The influence was so significant that, for example, the altered anger display with the head moved was interpreted as happiness.

## Adaptors

Like emotional expressions, adaptor gestures can be used to increase the aliveness of the robot. Aldebaran's Choreograph Suite, which can be used to create and play animations for NAO or Pepper, features a Autonomous Life option, making the robot to wag slightly, alter its head position and blink. Similarly, the robot Probo uses eye-blinking, changes its gaze randomly and flaps his ears regularly to create an illusion of aliveness.

Cuijpers and Knops found that a robot that uses idle motions, including a number of adaptor gestures such as posture shifts and head motions, is perceived as more human-like, alive and empathic [66]. Esteban et al. investigated the effect of socially reactive behaviours, including interactive eye blinking, and reactions to physical contact, and found that these feature help in achieve longer interactions and a higher degree of engagement [67].

### 1.3.2 Implementing gestures for robots

#### 1.3.2.1 The correspondence problem

As demonstrated above, a number of robots capable of gesturing have been developed and different aspects of robot gesture have been studied. When analysing the different research projects, an important feature can be noted. The gestures implemented in robots are mostly restricted to a set of gestures necessary for the current research, and often limited to one type of gestures.

The reason for this can be found in the way gestures are implemented. Gestures are mostly preprogrammed off-line for the current robot configuration.

The created postures are stored in a database and are replayed during interaction. This is the case for, amongst others, Robovie [68] and HRP-2 [69]. In specific, the robot postures can be based on human video recordings or photographs. This technique was for example used for generating emblematic gestures for NAO [40] and for constructing emotional expressions for Kobian [64]. As mentioned above, another possibility is the puppetering technique, used in [48] to create deictic gestures for NAO. Since the created postures composing the gesture are dependent on the morphology, they are robot specific and cannot be used for other robots with other configurations. Another common way to generate gestures is by mapping human motion capture data to the robot. This technique was used to create regulators for Simon [41], and emotional expressions for NAO in [52]. Another example are the motions generated for Repliee Q2 [70], for which a marker-based motion capture system is used. In [71], both a marker-based (Vicon) as a markerless motion capture system was used to reproduce human motion for the robot ARMAR-IIIb. Another possibility is to use the Kinect to perform skeleton tracking [72]. When transferring motion capture data to the robot, the mapping of the captured data is robot specific. Therefore, also these resulting gestures are dependent on the morphology and not usable for other robots. This issue is known as the correspondence problem [4][5]. When imitating, copying, mimicking or learning from an agent, a correspondence between the demonstrator and imitator needs to be specified. This means that a correct mapping between the two agents has to be identified. When the agents have similar bodies, the correspondence is obvious, however, when using agents with significantly different morphologies, this can become a difficult task. Therefore, in robotics, the correspondence problem is often omitted by coding the gestures for one specific robot configuration, as shown in the examples above. When working with a new robot platform, new joint trajectories to reach the desired postures need to be calculated and implemented. In this thesis, we aim to provide a solution for the correspondence problem and minimizing the workload when generating gestures for different robot platforms by providing a method that automatically calculates mapped gestures for a specified robot morphology.

### 1.3.2.2 Coping with the correspondence problem

Different attempts are made to ease the animation of social robots. [73] suggested to use the knowledge of animation artists to generate lifelike robotic motions by providing a generic software, whereby different types and combinations of gestures can be created by keyframing or by 3D character articulation. However, since the generated motions are still dependent on the used joint configuration, this does not address the correspondence problem.

A possible solution lies in the field of developmental robotics, by using neural networks to learn the correspondence between a posture and the robot's

joint angles [74]. A technique to teleoperate a humanoid robot without an explicit kinematic modeling by using neural networks was proposed by Santon et al.[75]. Muhlig et al. [76] eased the correspondence problem between a human tutor and robot in imitation learning by representing demonstrated movement skills using a flexible task space representation. Another approach of addressing the correspondence problem in imitation learning was suggested by Azad et al. [77], by using a reference kinematic model, the Master Motor Map, to convert motion capture data to an arbitrary robot morphology. This is a similar strategy as we use to map target gestures from a database to a robot configuration, which will be explained in chapter 3. In a later stage, the Master Motor Map was extended with a dynamic model and improved to allow for on-line reproduction of human motion to a humanoid robot [78]. In [79], a semi-general approach for generating natural arm motions, specifically for manipulation tasks is presented. Their inverse kinematics algorithm is based on neurophysiological findings, and decouples the problem of calculating joint angles for the arm from calculating those for the wrist. The sensorimotor transformation model of [80] is used to determine the arm posture, while the wrist angles are found by assuming a spherical wrist and using orientation inverse kinematics.

To animate virtual and robotic characters, Ribeiro et al. [81] developed a symbolic animation engine, called Nutty Tracks. The engine is based on computer animation techniques (CGI) and provides a high flexibility regarding the design, blending and modulation of animations. Animations are processed on symbolic joints, which can be mapped to a single, or set of real robotic joints. To control a specific robot with the engine, a corresponding output plugin should be developed, containing the representation of the robot structure and its joint specifications. To allow an arbitrary kinematic chain to orient its endpoint towards a target, while at the same time providing expressive control over the posture of the overall chain, a new inverse kinematics algorithm called ERIK, based on different techniques from CGI, was implemented.

In both [82] and [83], a gesture framework initially developed for virtual agents is applied on a humanoid robot. In [82], speech-accompanying gestures are generated for ASIMO by using the speech and gesture production model initially developed for the virtual agent MAX. For a specified gesture, the end effector positions and orientations are calculated by the MAX system and used as input for ASIMO’s whole body motion controller [84]. Similarly, in [83], speech-accompanying gestures are generated for NAO by using the GRETA system. The gestures are described independently of the embodiment by specifying features as the hand shape, wrist position and palm orientation. However, to obtain the corresponding joint values, a predetermined table listing values for the shoulder and elbow joints for all possible wrist positions is used. So although the gestures are described independently of the robot configuration, mapping these gestures to the robot requires hard coded joint information.

## 1.4 Objectives

The main objective of this research is to investigate how gestures can be efficiently generated for social robots with different morphologies. We are interested in arm gestures and upper-body postures generated by humans, and how similar motions can be generated for different morphologies, using one single generic method. The framework of the gesture method should be flexible, in a way that different types of gestures can be calculated. As discussed above, emotional expressions are crucial for creating socially accepted and fluent robot interactions. Additionally, deictic gestures appeared to be the most used type of illustrators, and have proven to be useful in human-robot interaction as well. Therefore, these two types of gestures were chosen to be incorporated in the research. Since in this work, the focus lies on the replicability of specific human gestures, the target group for which the method should be usable are robots that have features in the upper body that can be modelled as human-like elements, such as a head, arms, and/or an actuated torso. As mentioned above, because social robots are designed to operate in our daily environment, using human communication skills, many social robots are humanoid robots. For other types of social robots, such as zoomorphic designs, the gesture software should be usable to generate arm, body or head motions, on the condition that the robot can be modelled as such.

Although speech and gesture are closely related to each other, the use of speech is not incorporated in this research. In this thesis, we focus on how the different postures composing a gesture can be achieved by different robots. The autonomous generation of gestures in conjunction with a spoken utterance falls out of the scope of this thesis. In addition, the incorporation of our system into a global cognitive architecture, and therefore, the generation of gestures as a result to external or internal stimuli, is not covered here. For this, we refer to the EU-project DREAM.

A second objective is to investigate the usability of the method in the design process of social robots. Can the method be used to achieve insights in the importance of specific design aspects? Can it help in making substantiated trade-off's in the design process of new robots?

The final objective of this research is to develop a new version of the robot Probo, which should serve as an ultimate validation of both aspects of the developed gesture method; namely its usefulness in generating gestures for a random robot configuration, as well as its applicability in the design process of a new social robot.

## 1.5 Thesis outline

The work described in this dissertation is grouped into three major parts, addressing the three objectives discussed above. Part I comprises chapters 2 to 6 and discusses the development and validation of the different features of the gesture method. As a global introduction to the developed software, chapter 2 discusses the usability and overall characteristics of the method. Since for different types of gestures, different features are important, our method was designed to work in two modes. The *block mode*, which is used to calculate gestures whereby the overall arm placement is crucial, like emotional expressions, will be handled in chapter 3. Chapter 4 discusses the features of the *end-effector mode*, developed for calculating end-effector depending gestures, like pointing gestures. An interesting aspect of the method is that, not only different types of gestures can be calculated by providing two working modes, in addition, these two modes can be used in combination to create blended gestures. Chapter 5 covers how a *mode mixer* is used to generate these combined gestures. In some situations, it is desirable to express an emotional condition through an ongoing functional behavior, such as a deictic gesture. How a neutral behavior is modulated into an affective motion by using a set of characteristic performance parameters, is covered in the second part of chapter 5. Part II of this thesis, containing chapter 7, focusses on the use of the developed gesture method as a tool in the design process. As an ultimate validation of both aspects of the gesture method, Part III discusses the development of a new social robot, Elvis, designed within the frame of the Probo-project. Chapter 8 handles how three different joint configuration are chosen and how they are designed and physically realized, while chapter 9 lists a number of gestures generated by all three configurations.

# Part I |

## Development of a generic gesture method for social robots



# 2 | Methodology

In robot animation, the correspondence problem is often omitted by coding gestures for one specific robot configuration. Sharing gestures between robots is not straightforward and therefore, when working with a new robot platform, new joint trajectories to reach the desired postures need to be calculated and implemented. In this thesis, we aim to provide a solution for the correspondence problem and make the implementation of gestures more efficient by developing a generic method to generate gestures for social robots. The innovative aspect of this method is that it is constructed independently of any robot configuration, instead, a human base model is used as a basis to construct the software's framework. At runtime, the configuration of the robot is used as input to evaluate the method's generic structure, and the joint angles needed to establish a desired gesture are calculated. The framework is very flexible, allowing for easy modifications and improvements of the method, while adding new gestures to the database is also straightforward. In this chapter, the usability and global characteristics of the method are discussed. The next chapters cover the specific working principles and calculation aspects of different features of the developed method. An outline to this is given in section 2.1. In section 2.2, we explain how the generic aspect of the method is achieved by defining Body Action Units and a human base model. Finally, when a user wants to generate gestures for a certain robot, its configuration is specified to the method by a certain set of parameters. An example of how this is done is given in section 2.3.

## 2.1 Working modes of the gesture method

The aim of the developed method is to allow different types of gestures to be calculated for various robot configurations. An overview of the classification of gestures was given in section 1.2. Given their significance in creating meaningful and natural human-robot interaction, emotional expressions and deictic gestures were selected to be the two main gesture types of interest. Since for those two types of gestures, different features are important, the developed method will use different principles to generate them. For deictic gestures, especially the end-effector placement is crucial; the combination of the position and orientation of the end-effector determines to which point in space is referred to. In their work, Salem et al. [82] indeed decided to work with the end-effector pose and calculated task-space trajectories using inverse kinematics to calculated speech-accompanied gestures, based on the findings of [85]. On the other hand, for emotional expressions, the overall pose of the arms is very important to convey the gesture. In [86], experiments showed that emotions can be conveyed by body movements, even when the shape of the arm is minimised by using point-light displays, which indeed implies that the relative placement of the different bones or links, determining the overall shape of the arms, is important to convey an emotional expression. To take those two aspects into account, our method was designed to work in two modes: the *block mode*, developed to calculate gestures whereby the overall arm placement is crucial and the *end effector mode*, developed for end-effector depending gestures.

The block mode of the method will be handled in chapter 3 while chapter 4 discusses the features of the end-effector mode. An interesting aspect of the method is that, not only can different types of gestures be calculated by providing two working modes, in addition, these two modes can be used in combination to create blended gestures. For example, a person with an anxiety of, let us say, spiders, can create an emotional expression of fear when encountering one by moving the upper body backwards and raising a hand in front of the face, in combination with a pointing gestures towards the little creature using the other hand. In our method, an emotional expression in the sense of an explicit, full body action as calculated by our block mode, can take place in combination with a deictic gesture as calculated by our end-effector mode, by assigning each gesture to other body parts. Chapter 5 covers how these blended gestures are generated. In some situations, it is desirable to express an emotional condition in a different manner than by using explicit emotional expression. In such cases it is possible to express an emotional state through an ongoing functional behavior, such as a deictic gesture. How a neutral behavior is modulated into an affective motion by using a set of characteristic performance parameters, is covered in the second part of chapter 5.

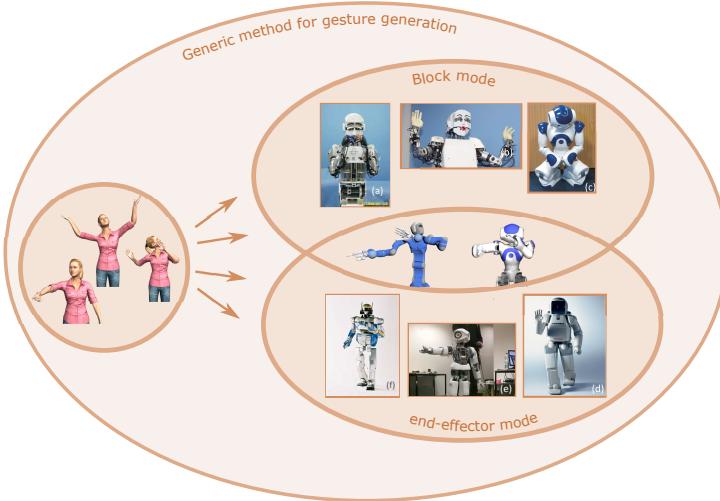


Figure 2.1: This schema represents the methodology of the developed gesture method, aiming to overcome the limitations of the current state of art where gestures are implemented for a specific robot. The method uses a human base model to store target gestures independently of any configuration in a database, and to calculate a mapping at runtime, based on the robot configuration specified by the user. Two modes are used to allow for different types of gestures to be calculated. The *block mode* is used to calculate gestures whereby the overall arm placement is crucial, like for emotional expressions, while the *end-effector mode* was developed for end-effector depending gestures, like deictic gestures. The two modes can be combined to generate blended emotional and deictic gestures. Furthermore, emotional states can be conveyed by modulating functional behaviors into affective motions. Robots: (a) WE-4RII ([61]). (b) KOBIAN ([64]). (c) NAO ([87]). (d) ASIMO ([88]). (e) Myon ([89]). (f) HRP-2 ([90]).

## 2.2 Achieving the generic aspect of the method

To ensure a generic method usable for different kinds of robots, the framework was developed without using any kind of robot morphology. Instead, a simplified model of the rotational possibilities of a human, the human base model, is used as a reference to construct the method. To define this model, firstly, a set of Body Action Units was developed. This approach is similar to the Facial Action Coding System of Ekman and Friesen [91], which defines a number of (Facial) Action Units to describe facial expressions. While the Facial AU's are defined as a muscle or a muscle group, our BAU's are based on the anatomical terms of motion.

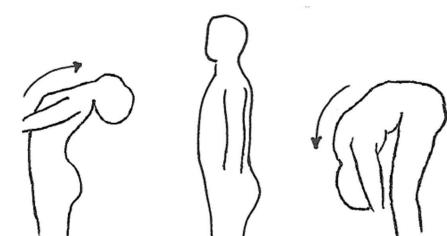
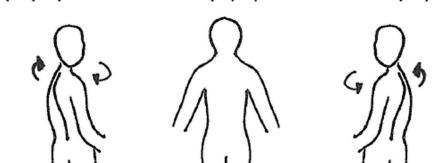
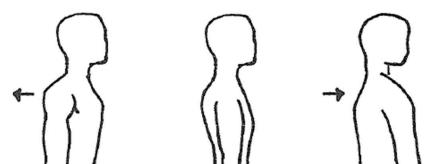
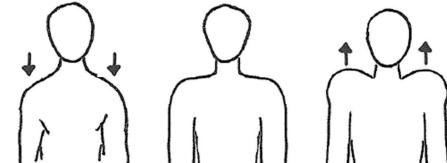
### 2.2.1 Motion of the human body

The anatomical terms of motion describe the different motion possibilities in the human body. Most of the motions in the body are osteokinematic motions; angular movements of bones around a joint axis. Most motions have an opposite, and are therefore often treated in pairs. The following major motion types can be distinguished [92][93][94]:

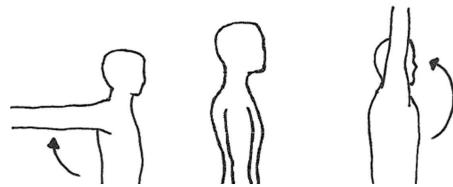
- **Flexion** and **extension** are motions that take place in a sagittal plane; a plane dividing the body into a right and left part. Flexion is the bending movement that, in general, moves the body part forward. Extension, on the other hand, involves the straightening movements, usually directed backwards, returning the body part into the neutral position after flexion.
- **Lateral flexion** or **lateral bending** is the term used to describe the sideways movement of the trunk.
- **Abduction** is a movement away from the midline of the body in the coronal plane; a vertical plane dividing the body into a ventral and dorsal section. **Adduction** is the opposite motion that restores the body part toward the midline.
- **Medial rotation**, or **internal rotation**, is the movement of a body part around its longitudinal axis, whereby the anterior surface rolls inward towards the midline. In contrast, for a **lateral rotation**, or **external rotation**, the anterior surface rolls outward, away from the midline. In specific, to refer to the rotation of the forearm, the terms **supination** and **pronation** is used.

By listing the possible motions of the most important joints of the body, an overview of the range of human movement can be obtained. In this thesis, we focus on upper-body motion. Table 2.1 lists the major upper-body movements, together with an illustrative figure.

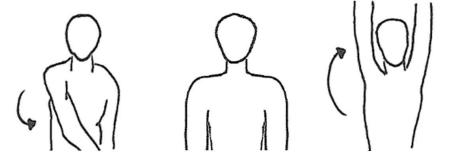
Table 2.1: List of upper-body motions

Term of motion	Illustration
Flexion/extension of neck	
Abduction/adduction of neck	
Rotation of neck	
Flexion/extension of spinal column	
Lateral flexion of spinal column	
Transversal rotation of spinal column	
Abduction/adduction of shoulder girdle	
Elevation/depression of shoulder girdle	

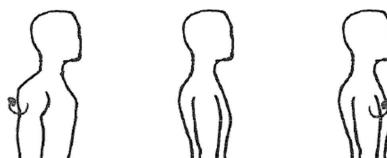
Flexion/extension of shoulder



Abduction/adduction of shoulder



Medial/lateral rotation of shoulder



Flexion/extension of elbow



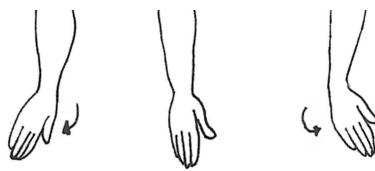
Pronation/supination of elbow



Flexion/extension of wrist



Abduction/adduction of wrist



## 2.2.2 Body Action Units and the human base model

Table 2.1 forms the basis in defining our human base model. The listed motions were numbered, resulting into 15 *Body Action Units*. All upper-body movements executed by humans can now be taxonomized into the activation of one or more defined BAU's. The defined BAU's are listed in table 2.2. The units are grouped into different *blocks*, corresponding to one human joint complex, such as the shoulder or the wrist. The grouping is based on the body part where the motion takes place and therefore straightforward, except for BAU 13. This Action Unit corresponds to the pronation/supination of the elbow. However, since in robots this motion is often implemented in the wrist zone, it was grouped into the wrist block. The defined blocks can subsequently be grouped into three body parts, namely the head, body and arm, which we refer to as *chains*. Using this structure, the base human model was defined. The model exists of four chains; the head, the body, the left arm and the right arm. The head chain consists of one joint block made up of the three joints corresponding to BAU 1 to 3. To get a reasonable model for the body, the body was modelled as consisting of three joint complexes, replacing the 24 articulating vertebrae of the spinal column. Therefore, the body chain consists of three similar body blocks, all including three joints corresponding to BAU 4 to 6. The base model of the arm consists of four blocks; the clavicle block, consisting of two joints corresponding to BAU 7 and 8, the shoulder and wrist consisting of three joints, corresponding respectively to BAU 9 to 11 and BAU 13 to 15, and the elbow consisting of one joint corresponding to BAU 12 (see table 2.2). The structure of the human base model corresponds to the ‘ultimate’ robot configuration. Since it is an approximate model of the rotational possibilities of a human, and most humanoid robots have a morphology that is less complex, the human base model is a structure that comprises the configuration of most existing humanoid robots. To describe postures in a quantitative way, a standard reference frame was defined. The standard x-axis was chosen to be in the walking direction, while the z-axis is the vertical pointing upwards. Subsequently, a frame was assigned to each block. For the bottom body block (called body 1), the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle’s reference axes are the body 3 embedded axes. For all other blocks of the arm, the axes are the embedded axes of the previous block when the model is placed in T-pose (Figure 2.2).

## 2.3 Specifying the robot’s configuration

When a user desires to generate gestures for a certain robot or model, its morphological information is specified by inputting a limited amount of rotational

Table 2.2: The Body Action Coding System

<b>Chain</b>	<b>Block</b>	<b>BAU</b>	<b>Description</b>
Head	Head	1	Flexion/extension of neck
		2	Abduction/adduction of neck
		3	Rotation of neck
Body	Body	4	Flexion/extension of spinal column
		5	Lateral flexion of spinal column
		6	Transversal rotation of spinal column
Arm	Clavicle	7	Abduction/adduction of shoulder girdle
		8	Elevation/depression of shoulder girdle
Arm	Shoulder	9	Flexion/extension of shoulder
		10	Abduction/adduction of shoulder
		11	Medial/lateral rotation of shoulder
Arm	Elbow	12	Flexion/extension of elbow
	Wrist	13	Pronation/supination of elbow
		14	Flexion/extension of wrist
		15	Abduction/adduction of wrist

information and the configuration's Denavit-Hartenberg (DH) parameters into the program. A small introduction on the DH-convention is available in appendix B. To specify the configuration, the different joints of the robot are grouped into the chains and blocks of the human base model. The human base model's main purpose it thus to provide a structure to code the framework of the method. When the program is launched for a desired configuration, the variables corresponding to the structure of the human base model, such as the amount of chains, amount of blocks and joints per block, are evaluated corresponding to the current joint configuration. As such, the method can be used for any robot that consists at least of one arm, a body, or a head.

Table 2.3 illustrates how a configuration is specified to the method by considering the case of the robot NAO [3]. For this process, the robot is placed in T-pose; a pose whereby the agent is standing straight, the head facing forward and the arms lifted in an angle of 90° with the body. The elbow's joint axis is oriented vertically, and the hand palms are faced forward. The top row of table 2.3 shows the actuated joints in the upper-torso of NAO, superposed on

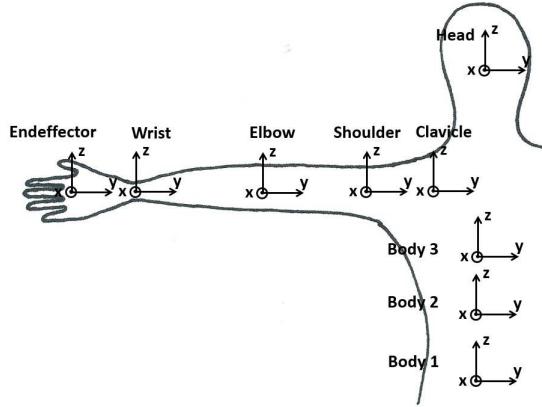


Figure 2.2: A reference frame was assigned to each block. For the body 1 block, the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle's reference axes are the body 3 embedded axes. For all other blocks of the arm, the axes are the embedded axes of the previous block.

the virtual model of the robot placed in T-pose. In a first step, the available joints are grouped into the corresponding blocks and chains. The robot has five joints in the arm, whereof three are responsible for the shoulder motion. The first three arm joints are therefore grouped into the shoulder block. The next joint in the arm chain is responsible for the flexion/extension of the elbow and therefore makes up the elbow block. The last arm joint corresponds to the pronation/supination of the lower arm and is the only joint that is to be placed in the wrist block. Only one of the three possible wrist joints is available; we call this an *incomplete block*. NAO does not feature any joint that can be modelled into the clavicle block. Therefore, this block is not available for this configuration, which results in a left and right arm chain consisting of only three blocks; the shoulder, elbow and wrist. The head contains only two of the three possible head joints; the joint corresponding to the abduction and adduction is missing. So as for the wrist block, also the head block is incomplete. Furthermore, the torso does not feature actuated joints and therefore, the complete body chain is missing. In a next step, suitable Denavit-Hartenberg (DH) frames are assigned to each joint. Since no body chain is available here, every chain is treated separately. The end-effector of the arm chain is placed in the hand palm, while that of the head chain is situated at the level of the eyes. The corresponding DH-parameters can then be listed for every available block

Table 2.3: To specify a robot configuration to the method, the model is placed in T-pose. The available joints are grouped into the different blocks and chains of the human base model. DH-frames are assigned to each joint, and the corresponding DH-parameters are listed and used as input for the method. Furthermore, the relative orientation of the base frame of every available chain with respect to the standard reference frame needs to be specified.

Joint configuration						
Chain	Joint	DH-parameters				$base R_{stand}$
		$\alpha$ (rad)	a (cm)	d (cm)	$\theta$ (rad)	
Head	Head 1	$\pi/2$	0	0	0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -21.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
	Head 2	0	5	0	$-\pi/2$	
Right	Shoulder 1	$-\pi/2$	0	0	0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & -1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
	Shoulder 2	$\pi/2$	0	0	0	
	Shoulder 3	$-\pi/2$	0	10	0	
	Elbow	$\pi/2$	0	0	0	
	Wrist	0	0	17	0	
Left	Shoulder 1	$-\pi/2$	0	0	0	$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & 1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
	Shoulder 2	$\pi/2$	0	0	0	
	Shoulder 3	$-\pi/2$	0	10	0	
	Elbow	$\pi/2$	0	0	0	
	Wrist	0	0	17	0	

in the present chains. This information is listed in the center columns in the bottom part of table 2.3. Next to the DH-parameters, also the orientation of the base frame of every available chain needs to be indicated. The orientations are specified with respect to the standard reference frame, placed in the pelvis of the robot. The relative orientation of the chain base frame and the standard reference frame,  ${}^{base}R_{stand}$ , for the three available chains is listed in the most right column of table 2.3.

## 2.4 Conclusions

This aim of this thesis is to investigate how gestures can be flexibly generated for different robot morphologies and to develop such a generic method. The developed method provides a framework to overcome the correspondence problem by describing target gestures independently of a configuration, and calculating a mapping based on a random configuration chosen by the user. In this chapter, the global characteristics of the method were discussed. Since for different types of gestures different features are important, the method was designed to work in two modes. The block mode is used to calculate gestures whereby the overall arm placement is crucial, like for emotional expressions, while the end-effector mode is developed for end-effector depending gestures, like deictic gestures. To achieve the generic aspect of the method, a human base model was defined to serve as a reference to construct the method. This human base model, consisting of several chains and blocks, is an approximation of the rotational possibilities of a human and therefore comprises most of the available humanoid robot configurations. To generate gestures for a certain robot, the different robot joints need to be grouped into the blocks of the human base model. As such, the method can be used for any robot consisting of a head, body and/or arm. The exact morphology is specified to the program by inputting the Denavit-Hartenberg parameters of the configuration and a maximum of four rotation matrices. The next chapters cover how the necessary joint angles to establish a desired gesture are calculated for a specified configuration.

PART I. DEVELOPMENT OF A GENERIC GESTURE METHOD FOR SOCIAL ROBOTS

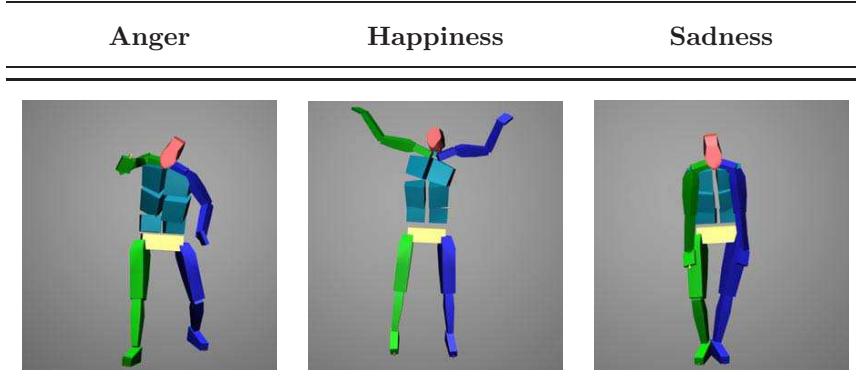
# 3 | Generating emotional expressions using the block mode

To investigate the effect of gestures in human-robot interaction, a number of social robots capable of gesturing have been designed. Gestures are often pre-programmed off-line or generated by mapping motion capture data to the robot. Since these gestures are dependent on the robot's joint configuration, they cannot be used for other robots. Therefore, when using a new robot platform with a different morphology, new joint trajectories to reach the desired postures need to be implemented. The developed method discussed in this dissertation aims to minimize the workload when implementing gestures on a new robot platform and facilitate the sharing of gestures between different robots. This chapter discusses the block mode of the method, which is used to calculate gestures whereof the overall arm placement is crucial, such as emotional expressions. To calculate a gesture for a certain configuration, the block mode uses a set of target gestures listed in a database and maps them to that specific configuration. The method was validated on the virtual model of different robots and an online survey was performed to evaluate the user's perception of the output of the method. The results of this survey showed that the calculated gestures for a certain robot configuration well resemble the target gestures, and thus that our methodology to map gestures to different robot morphologies gives good results.

This chapter is based on the following publication:

Greet Van de Perre, Michael Van Damme, Dirk Lefeber and Bram Vanderborght. Development of a generic method to generate upper-body emotional expressions for different social robots. *Advanced Robotics*, 29(9):597–609, 2015.

Table 3.1: Examples of emotional expressions for anger, happiness and sadness. Originating from the UCLIC Affective Body Posture and Motion Database [95]



### 3.1 Methodology

The block mode is designed to generate gestures whereby the overall posture is important, such as emotional expressions. As clarified in section 1.2.3, we use this term to indicate a gesture whereby a full body posture is used to convey an emotional state. Some examples of emotional expressions are given in table 3.1. These postures origin from the UCLIC Affective Body Posture and Motion Database [95] and are recorded using human motion capture.

To calculate emotional expressions, the block mode uses a set of target gestures listed in a database and maps them to a desired configuration. To generate this kind of gestures by one generic method for robots with different joint configurations and link lengths, it is not sufficient to only impose the pose of the end-effector. Inverse kinematics for robots with a different configuration and different relative arm lengths could result in unrecognisable global postures when only using the end-effector pose. A good scaling, depending on the robot configuration, is crucial to guarantee a natural and human-like overall calculated posture. Therefore, to ensure a good overall posture, the orientation of every joint complex the robot has in common with a human needs to be imposed.

### 3.2 Target gestures

The database lists a number of target gestures that are used as a reference to calculate gestures for a specific configuration. Since the speed of the movements

contributes to the recognizability of gestures [61, 96], the database does not only specifies static postures but offers the possibility of using motion sequences. The target gestures in the database therefore consist of one or more important postures, accompanied by a timing specification. The approach of starting from expressive key poses to create convincing and believable displays is a common animation technique [97][98].

Most of the target gestures were chosen by using the UCLIC Affective Body Posture and Motion Database [95]. This database consists of a number of motion capture data sets for several emotional expressions. For every data set, an expressive avatar was generated by selecting the static posture from the motion sequence, that the actor himself evaluated as the most expressive instant. These static postures were subsequently labelled and rated by a number of observers from three different cultures. For every emotion, the motion sequence corresponding to one of the best-scoring postures was chosen for our database. Because of the easily-extendable library and the flexible framework, also other gestures corresponding to the emotions can be incorporated if desired, to allow for some variance of the gestures during human-robot interaction.

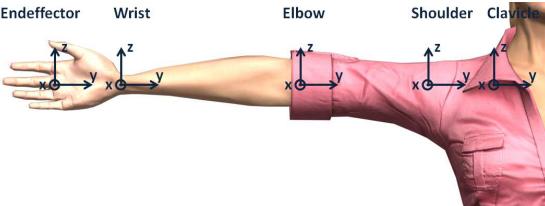
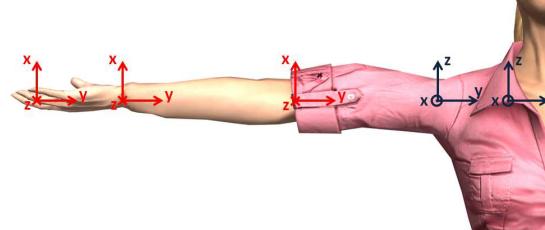
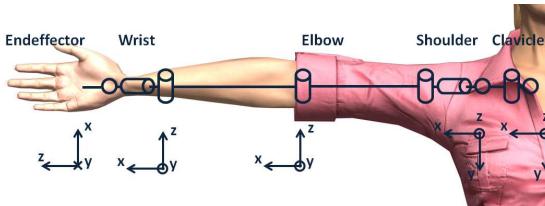
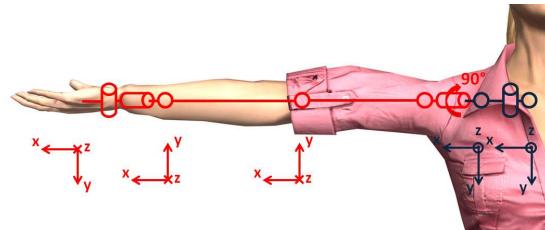
The target postures are quantitatively described by specifying the orthopaedic angles of every block of the human base model. Orthopaedic angles are similar to Euler angles, but are defined according to clinical terms such as flexion and abduction [99]. Using the reference frames assigned to each block in section 2.2.2, the orientation of block  $i$  is determined by the  $zyx$ -Eulerangles of frame  $i + 1$  (the base frame of block  $i + 1$ ) with respect to frame  $i$  (the base frame of block  $i$ ) (see figure 2.2). The data is stored in the program as rotation matrices.

### 3.3 Mapping the gestures to a configuration

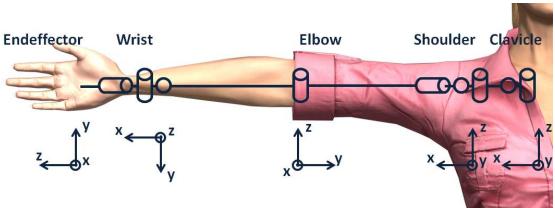
To make a model or robot perform a desired gesture, the target posture sequences described in section 3.2 are mapped to the joint configuration. The method can be used for any robot or model whereof its configuration consists of *one or more* parts of the human base model, namely a head, a body, a left and/or right arm. The joints of each chain must be grouped into the different blocks composing that chain, whereby the number of driven joints placed in a block cannot exceed the number of joints in the corresponding block in the human base model.

To specify the robot's joint configuration in the program, the Denavit-Hartenberg (DH) parameters of every present block need to be specified. A target posture is mapped to the configuration by imposing the orientation of the end-effector of the different blocks and calculating the corresponding joint angles. Missing chains or blocks are ignored. The robot Keepon [100] for example, is a snowman-like robot without arms. Therefore, the mapping of a posture will be

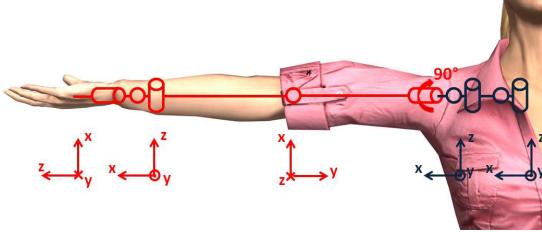
Table 3.2: This example illustrates the mapping of gestures. Since different robot configurations lead to different Denavit-Hartenberg matrices, the mapped rotation matrices will differ as well.

	T-pose (neutral position)				Desired posture			
Base model	 Endeffector    Wrist    Elbow    Shoulder    Clavicle				 Desired posture			
	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$				$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$			
Configuration 1	 Endeffector    Wrist    Elbow    Shoulder    Clavicle				 Desired posture			
	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$				$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$			

Configuration 2



$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

restricted to the body and head. Since the target postures are stored in the program under the form of Euler angles with respect to the standard reference frame, they need to be transformed to the current Denavit-Hartenberg frames to be able to calculate the correct joint angles. Therefore, besides the Denavit-Hartenberg parameters, the rotation matrix between the standard and the DH-base frame of every present chain needs to be specified as well (maximum four matrices) as input for the method. A simple example illustrating the mapping is shown in table 3.2. The first row displays the base model while two different arm configurations, both consisting of 9 degrees of freedom, are shown in the second and third row. In the first column, all the configurations are in T-pose and the relative orientation of all blocks with respect to their predecessor is displayed. For the base model these are the assigned frames as discussed in section 2.2.2, while for the two configurations these are the Denavit-Hartenberg base frames of every block. The second column displays the targeted posture; a stretched arm with the hand palm facing up. In order to reach this posture, an outward medial rotation of the shoulder (BAU 11) around 90° is necessary. For the base model, this means a rotation of -90° around the y-axis of the shoulder block reference frame. All the lower lying blocks of the arm (elbow and wrist) are included in this movement, and therefore only the relative rotation of the elbow with respect to the shoulder will change. This new rotation matrix, depicted in red in the *Base model* row of table 3.2, serves as the target rotation matrix and will be mapped to the configurations to calculate the desired posture. In order to correctly map the desired orientation to the current configuration, the orientation of the Denavit-Hartenberg frames with respect to the standard frame of the base model needs to be considered. The correct mapped matrix can be calculated as follows:

$$R_i = {}^{b,i} R_{st} \cdot R_{i,des} \cdot {}^{st} R_{e,i} \quad (3.1)$$

Here,  $R_i$  is the mapped rotation matrix for block i,  ${}^{b,i} R_{st}$  the rotation matrix between the base frame of block i and the standard reference frame,  $R_{i,des}$  the target rotation matrix in standard axes for block i, loaded from the database and  ${}^{st} R_{e,i}$  the rotation matrix between the standard reference frame and the end frame of block i, i.e. the base frame of block i+1.

The mapped rotation matrices for the shoulder can then be calculated by substituting the correct rotation matrices in Eq. B.1. Since the Denavit-Hartenberg frames are different for the two configurations, the rotation matrices between these frames and the standard frame will differ, resulting in a different mapped rotation matrix for the shoulder block. The difference in matrices is the reason why in the state of the art, gestures are always implemented for one specific robot platform; only the mapped matrices for that robot are specified. By using matrices defined in a standard reference frame and scaling them by using the Denavit-Hartenberg matrices corresponding to the robot's

joint configuration, our method makes it possible to map gestures to different robots. As described in section 3.2, the gestures listed in the database consist of a set of postures specified in time. For every posture, the mapped rotation matrices are calculated as explained above. Depending on the specified time constraints, a set of intermediate postures is calculated by interpolating between the current posture of the robot and the desired one, i.e. the next posture specified in the gesture database. In that way, a fluent motion with the desired speed characteristics can be obtained. For every block, the necessary joint angles to establish a desired posture can be calculated from the mapped rotation matrix by using inverse kinematics.

### 3.3.1 Complete configuration

A full configuration is a configuration similar to the base model; consisting of four chains, each containing a specific number of blocks, which are in turn made up of a specified number of joints as listed in table 2.2. For each block, a mapped rotation matrix is calculated as described in section 3.2. This matrix is the necessary orientation the end-effector of the block needs to adopt in order to reach the desired overall posture. To calculate the corresponding joint angles numerically, an inverse kinematics algorithm is necessary. For each block, the joint angles are calculated by the closed-loop inverse kinematics algorithm shown in figure 3.1 [101]. In a first step, the time derivate of the joint angles  $\dot{q}$  is calculated:

$$\dot{q} = J_A^\dagger(q)(\dot{x}_d + K(x_d - x_e)) \quad (3.2)$$

Here,  $x_d$  is the desired end-effector pose. Since the maximum number of joints in one block is three, it is not necessary to use all six parameters of the pose; the consideration of the orientation of the end-effector is sufficient. Therefore,  $x_d$  is reduced to the zyx-Euler angles corresponding to the mapped rotation matrix.  $J_A^\dagger(q)$  is the Moore-Penrose pseudo inverse of the analytical Jacobian  $J_A(q)$ ,  $x_e$  the current end-effector pose; i.e. the current zyx-Eulerangles, and  $K$  a positive definite gain matrix. The analytical Jacobian and the current end-effector pose are calculated as a function of the current joint angles.

The desired joint angles  $q$  are then calculated by integrating  $\dot{q}$  with the Runge-Kutta algorithm [102]. More information on how the inverse kinematics algorithm is implemented in the gesture software and how the different terms can be determined is available in appendix B.

Since the complete configuration has the same rotational possibilities as the human base model, it will always be possible to calculate a correct set of joint angles to generate the desired posture. The calculated angles are then sent to a virtual model to visualize the posture. The loop of the algorithm is closed

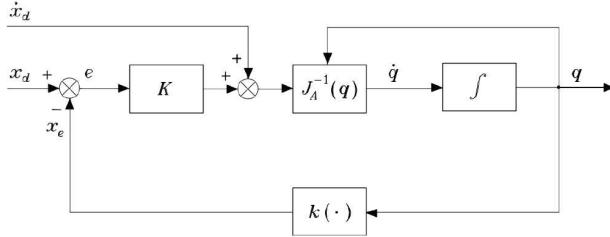


Figure 3.1: Schematic representation of the closed loop inverse kinematic algorithm used to calculate the joint angles for the desired body posture [101].

by calculating the new actual end-effector orientation by direct kinematics (depicted by  $k(\cdot)$  in figure 3.1) and using it as input to determine the current error.

### 3.3.2 Configuration with a reduced number of DOF

In most cases, the robot will have a simplified configuration, and will therefore not have the same amount of degrees of freedom as the human base model. The robot WE-4RII [61] is one of the few robots having a complete 9 DOF arm with an actuated clavicle. Most other robots will miss the BAU's corresponding to the clavicle. This is for example the case for iCub [103] and ASIMO [104]. Joints in the wrist are also often omitted. NAO [105] and QRIO [106] for example, have no possible wrist movements except for the pronation/supination corresponding to BAU 13. Working with incomplete configurations implies that some desired orientations will not be reachable for certain blocks, and therefore the exact desired posture cannot be established. In that case, a good approximated posture needs to be calculated. Mapping the Facial Action Coding System onto an incomplete robot face, is relatively easy. The Facial Action Units correspond to 'stand alone' joints and hence, missing Action Units can be ignored without disturbing the placement of the other Units. In our Body Action Coding System, the Body Action Units are grouped in blocks whereof the end-effector's orientation is specified. When a complete block is missing, this problem becomes similar to that for a missing Facial Action Unit and the entire block can be ignored. However, in the case of incomplete blocks (i.e. blocks with one or two missing joints), the mapping becomes complicated since a missing joint in a block will have an influence on the values of the other joints and can therefore not be simply ignored. Consider, for example, a configuration whereof the joint responsible for the elevation and depression of the shoulder girdle (BAU 8) is missing. If a targeted gesture includes both the adduction/abduction and elevation/depression of the shoulder girdle, the desired orientation matrix for the clavicle can never be obtained from only

the one joint in the configuration. In that case, the method will calculate the necessary joint angles in order to perform an overall posture as close as possible to the target posture. In order to do so, virtual joints are added to the blocks when necessary. The virtual joints are chosen so that they complete the current block, making the rotational possibilities equal to those of the corresponding block of the human base model. Then, as for the complete configuration, the correct joint angles needed to establish the desired posture can be calculated by the inverse kinematics algorithm depicted in figure 3.1. After the calculation, the angles corresponding to the real and virtual joints are separated; only the real joint angles are sent to the virtual model to visualize the posture. Figure 3.2 summarises the program's work flow to calculate the correct joint angles for a desired posture.

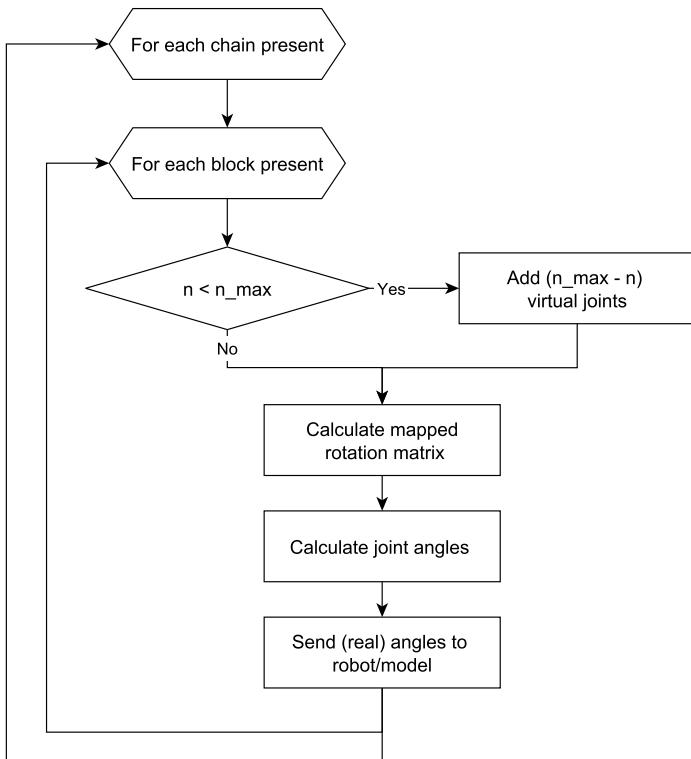


Figure 3.2: Schematic representation of the program flow. For every block present in the configuration, a mapped rotation matrix is calculated. The corresponding joint angles are calculated by inverse kinematics and sent to the virtual model or robot.

## 3.4 Results

### 3.4.1 Emotional expression of *happiness* for different configurations

The method was validated on several configurations. Table 3.3 shows a calculated posture for different arm configurations. To generate the renderings, the joint trajectories calculated by the method were loaded to the corresponding virtual model in Autodesk 3DS MAX. The top row shows the targeted posture, in this case the end posture for the emotional expression for *happiness*. The posture is visualised using a human virtual model, purchased from the Rock-  
etBox Libraries [107]. In the second row, a complete configuration is shown, with a 9 DOF arm, 3 DOF head and 9 DOF body. In this configuration, all the blocks are complete and therefore, a set of joint angles can be calculated for every block wherefore the corresponding overall posture equals the target posture. The same human model was used to visualize the posture calculated for this complete configuration. Configuration 2 shows the ASIMO robot [1]. This virtual model was downloaded from the tf3dm-website [108] and adapted to our specific needs. The robot ASIMO has a 7 DOF arm, with a complete 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist, only the clavicle block is missing. The head chain is complete, since it contains all three joints that make up the head block. ASIMO's body contains only one joint corresponding to BAU 6, so the body chain is modelled as consisting of one single body block, body 1, containing one joint. For the head, shoulder, elbow and wrist blocks, a set of joint angles can be calculated to reach the desired orientation of the blocks. But since the body 1 block is incomplete, virtual joints need to be added to calculate an approximate solution. When observing the calculated posture, one can see that, although this is not a complete configuration, the obtained posture is very recognizable. Notable is the lower placed left arm, because of the lacking of a clavicle block and the possibility of lateral flexion of the body. Configuration 3 shows the Justin robot [2]. The virtual model is an original file from the designers at DLR Robotics and Mechatronics Center. The robot's body contains three joints and can be modelled as consisting of three incomplete body blocks, each containing only one joint corresponding to BAU 4; flexion and extension of the spinal column. Also the head block is incomplete. It consists of two joints, corresponding to BAU 1 and 3, while the joint corresponding to the abduction and adduction is missing. Justin has a 7 DOF arm, with a complete 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist, similar to ASIMO's arm configuration. The same remarks for the calculated posture as for ASIMO can be made. In addition, the exact orientation of the head differs from that of the target posture, because of the missing joint responsible for the abduction and adduction of the head. The last configuration is that of NAO

[3]. The virtual model was shared online by Aldebaran. NAO's head has a similar configuration as that of Justin. However, no joint is located in the body and therefore, the complete body chain is missing. NAO's arm consists of 5 joints; composing a complete shoulder and elbow. The wrist only consists of one joint, therefore, to calculate an approximate solution, two virtual joint needs to be added to complete the wrist block. In the resulting posture, especially the absence of a joint responsible for flexion and extension of the wrist has an influence on the resulting wrist placement.

The calculated joint trajectories for the left arm chain for the four configurations listed in table 3.3 when going from the T-pose to the end posture for the expression for *happiness* are plotted in figure 3.3. For the human configuration, a trajectory is calculated for each joint. But since missing blocks are ignored by the method, no trajectories are calculated for the clavicle block for the three robot configurations. Because the joint configuration of the shoulder is different in the four examples (for Justin and ASIMO, the first shoulder joint is placed at a different angle), four different trajectories are calculated to reach the same end orientation of the end-effector of the shoulder block. Since the elbow block is the same for the four configurations, the same joint trajectory is obtained for all four models. However, the trajectory for the human configuration is biased from the others because of a difference in DH-parameters. Concerning the wrist block, the calculated trajectories for ASIMO and Justin are the same, since they have a similar joint configuration. As NAO's wrist only contains one single joint, two virtual joints are added to the block to calculate an approximate solution. The trajectories of the virtual joints are depicted by dotted lines.

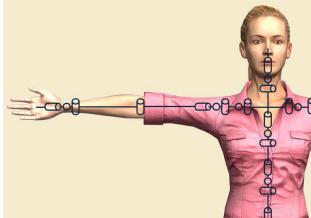
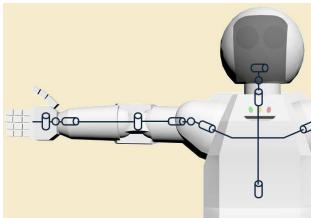
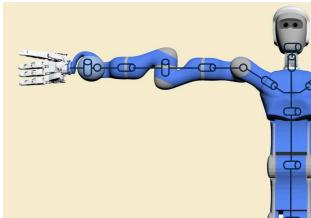
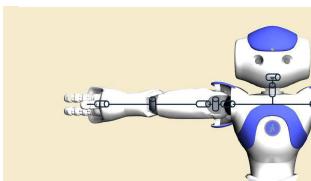
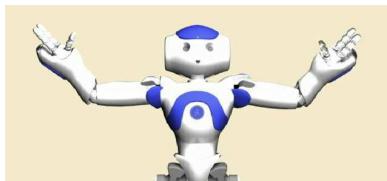
### 3.4.2 Survey

To validate the output of the method, an online survey was performed. The trajectories for the gestures corresponding to the six basic emotions were calculated for three robots, namely ASIMO, Justin and NAO. A separate movie for every robot performing each gesture and six additional movies showing a human virtual model performing the target gestures were made. These movies and the link to the online survey can be found on the Probo-website<sup>1</sup>. The end postures of the gestures can be found in table 3.4. The survey's objective was to investigate the quality of the mapped gestures, so to check whether the calculated gestures for different configurations can be correctly matched to the initial target gestures. Therefore, in the first part of the survey, the six target gestures, labelled with the emotion they convey, were shown. In the next part of the survey, the videos showing the different gestures by the three robots were shown in a randomized order. After watching every video, the participant was

---

<sup>1</sup><http://probo.vub.ac.be/GestureMethod/BlockMode.htm>

Table 3.3: Results of the method for different arm configurations. The first column shows the joint configuration, while the second column shows the mapped end posture for the expression of *happiness* for that configuration.

	Configuration	Calculated posture
Base model		
full config		
ASIMO		
Justin		
NAO		

asked to match the shown gesture with one of the target gestures by means of a multiple choice form (figure 3.4). This methodology is similar as the strategy used to evaluate the expressive behavior of Kismet, where subjects were asked to perform a comparison task between the robot's expressive faces and a series of line drawings of human expressions [109].

73 participants with origins from six different countries and ages varying from 18 to 82 years old filled out the survey. The recognition rates, expressed in percentage, are listed in table 3.5. The overall rates for the correct linking of the gestures are relatively high, whereof we can conclude that the calculated gestures in general well resemble the target gestures from the database and therefore, that our methodology to map gestures to different robot configurations gives good results. Especially the gesture for *sadness* gave good results: a recognition rate of 99 percent for Justin and NAO, and even of 100 percent for ASIMO was obtained. This can be attributed to the fact that the *sadness* gesture is a very distinctive one. Also the gestures for *disgust*, *fear* and *happiness* have high recognition rates for all three configurations. Striking is the low recognition rate for the mapped gesture corresponding to *surprise* for ASIMO; only 55 percent of the participants correctly linked this gesture to the corresponding target gesture, 31 percentage linked it to the target gesture for *disgust*. The mapped gesture for *surprise* significantly differs from the target gesture for *disgust*. Therefore, we assume that the cause for this low recognition rate lies in the choice of the target gesture. Probably, not all our subjects recognized this gesture as an expression for *surprise* and let the recognition of the mapped gesture as a certain emotion prevail over the linkage of it with one of the target gestures.

## 3.5 Conclusions

In this chapter, the block mode of our developed gesture method was described. This mode is used to calculate gestures whereby the overall pose is important. To generate gestures for a desired configuration, a set of target gestures, stored in a database is mapped to the configuration. The target gestures are described by the orthopaedic angles of every block of the human base model that was defined in the previous chapter. Based on this information and the specifications of the chosen robot configuration, a mapped rotation matrix is calculated for every block. The inverse kinematics problem for every available block is solved by imposing the mapped rotation matrix as end-effector constraint. The block mode was validated on different configurations, including those of the robots ASIMO, Justin and NAO, with arm configurations ranging from 9 DOF to only 5 DOF. The results are visualized by sending the calculated joint trajectories to a virtual model. To evaluate the user's perception of the output of the method, an online survey was performed. The aim of the survey was to

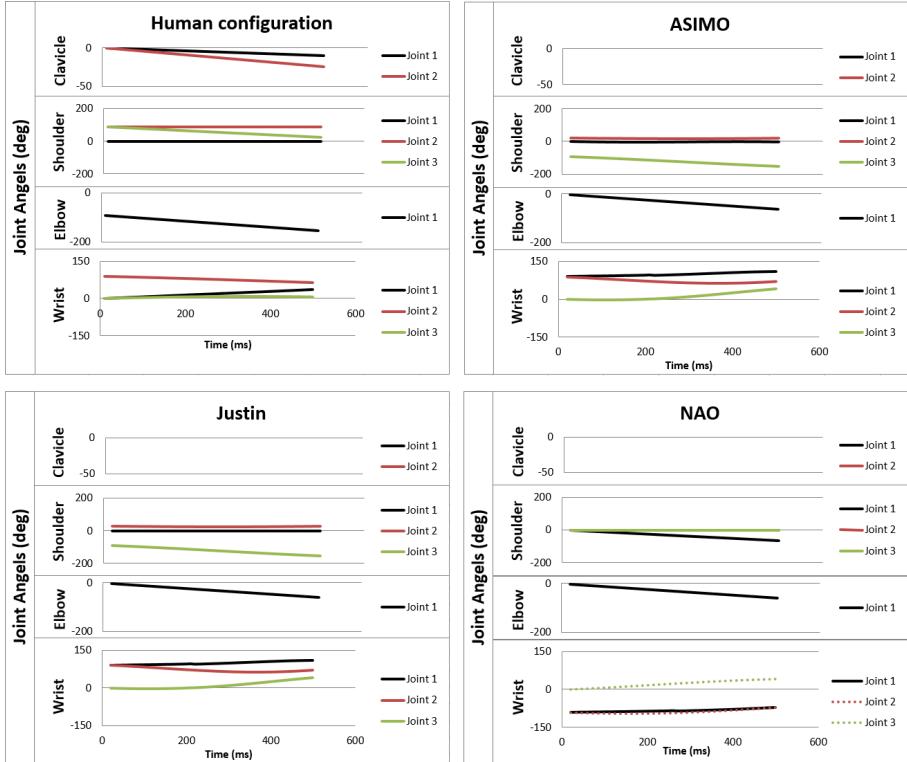


Figure 3.3: This figure shows the calculated joint trajectories for the left arm chain for the four configurations listed in table 3.3 when going from the T-pose to the end posture for the expression for *happiness*. As missing blocks are ignored by the method, no trajectories are calculated for the clavicle block for the three robot configurations. As NAO's wrist only contains one single joint, two virtual joints are added to the block to calculate an approximate solution. The trajectories of the virtual joints are depicted by a dotted line.

Table 3.4: End posture of the gestures used in the survey. The first column shows the end posture of the target gestures for expressing the six basic emotions, while columns 2, 3 and 4 respectively show the calculated end posture for the robots ASIMO, Justin and NAO.

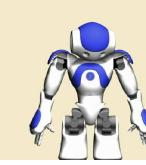
Target posture	ASIMO	Justin	NAO
			
			
			
			
			
			



Figure 3.4: The answer form used in the survey. The participant is asked to link a shown robot gesture with one of the target gestures previously shown in the survey. The end posture of every gesture is depicted together with its label.

Table 3.5: Results of the survey with 73 participants. The correct recognition rates of the mapped gestures for the six basic emotions are expressed in percentage match.

	ASIMO	Justin	NAO
Anger	76	76	82
Disgust	87	86	87
Fear	90	94	94
Happiness	86	99	88
Sadness	100	99	99
Surprise	55	81	87

check whether the calculated gestures are recognizable from the initial target gestures. Therefore, subjects were asked to link a series of calculated gestures with the set of target gestures by means of a multiple choice form. The results of this survey showed that the mapped gestures resemble the target gestures well, and thus that the block mode successfully generates gestures for different robot morphologies.

PART I. DEVELOPMENT OF A GENERIC GESTURE METHOD FOR SOCIAL ROBOTS

# 4

# Calculating reaching and pointing gestures using the end-effector mode

Since the implementation of gestures for a certain robot generally involves the use of specific information about its morphology, these gestures are not easily transferable to other robots. To cope with this problem, we proposed a generic method to generate gestures, constructed independently of any configuration and therefore usable for different robots. This chapter focuses on the end-effector mode of the method, which is used for gestures whereby the placement of the end-effector is crucial. In some situations, for example when reaching for an object, the exact position of the end-effector is important and specified by the user. This situation is called the *place-at* condition. On the other hand, when pointing *towards* an object, several end-effector poses are possible to achieve a pointing gesture to the specified target. In the *pointing* condition, an optimal solution is selected from the collection of postures fulfilling the pointing constraint. When working with end-effector positions, an important feature to consider is the workspace of the robot. When a desired position is specified by the user, the method verifies if this point is in reach of the robot, by using an approximation of the robot's workspace. If the desired position is in range of the robot, a suitable trajectory towards this point is calculated. The end-effector mode was validated on several configurations, including those of the robots ASIMO, NAO and Justin.

This chapter is based on the following publication:

Greet Van de Perre, Albert De Beir, Hoang-Long Cao, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Reaching and pointing gestures calculated by a generic gesture system for social robots. *Robotics and Autonomous Systems*, 83:32–43, 2016.

## 4.1 *Place-at* condition

### 4.1.1 Calculating a posture for a specified end-effector position

In the place-at condition, the user imposes the desired end-effector position for the left and/or right arm. The end-effector is in this case located in the middle of the hand palm. A set of joint angles corresponding to the positional constraint can be calculated by solving the well-known inverse kinematics problem. The interesting feature of our method is that the framework is constructed very generally and independent of any configuration. Mapping information is only calculated during runtime by using the DH-parameters and a set of rotation matrices specified by the user. The desired end-effector position is specified by defining its Cartesian coordinates in the standard reference frame. The corresponding position in the arm base frame - depending on the configuration, this is most probably the clavicle or shoulder base frame (see figure 2.2) - can be calculated by taking into account the current orientation of the body chain. This position  $x_d$  can then be used as input for the same closed-loop inverse kinematics algorithm as used in the block mode to calculate a set of joint angles. Firstly, the derivative  $\dot{q}$  of the joint angles is calculated [101]:

$$\dot{q} = J_A^\dagger(q)(\dot{x}_d + K(x_d - x_e)) + (I - J_A^\dagger(q)J_A(q))\dot{q}_0 \quad (4.1)$$

Here,  $J_A^\dagger(q)$  is the Moore-Penrose pseudo inverse of the analytical Jacobian  $J_A(q)$ . Since in this mode, we only impose the positional coordinates in  $x_d$ ,  $J_A(q)$  is reduced to its translational part only.  $x_e$  is the current end-effector position, and  $K$  a positive definite gain matrix. In the highly probable case of an arm chain consisting of more than three degrees of freedom, the functional redundancy is used to guide the configuration into a natural posture. In that case; the term  $(I - J_A^\dagger(q)J_A(q))$  will differ from zero, activating the influence of  $\dot{q}_0$  on the calculated joint speeds.  $\dot{q}_0$  introduces the cost function  $w(q)$  (see section 4.1.2):

$$\dot{q}_0 = k_0 \left( \frac{\partial w(q)}{\partial q} \right)^T \quad (4.2)$$

with  $k_0$  a positive weight factor. The desired joint angles  $q$  are calculated by integrating  $\dot{q}$  with the Runge-Kutta algorithm [102]. More information on how the inverse kinematics algorithm is implemented in the gesture software and how the different terms can be determined is available in appendix B.

### 4.1.2 Natural postures

In case of redundancy, the cost function  $w(q)$  will push the configuration into a natural, human-like posture. The optimization of arm motions using cost functions is widely studied and different types of functions were proposed in the literature. Possible optimization criteria are minimal work [110], jerk [111], angular displacement (MAD) [112] or torque [113][114][110]. Another possibility is to use the joint range availability (JRA) criterion [115]. Here, the algorithm will try to find an optimal humanlike posture by keeping the joints close to their central position, away from their limits [116]:

$$JRA = \sum_{i=1}^n w_{0,i} \frac{(q_i - q_{ci})^2}{(q_{max,i} - q_{min,i})^2} \quad (4.3)$$

where  $q_i$  is the current value of joint i and  $q_{ci}$  its center value.  $q_{max,i}$  en  $q_{min,i}$  are the maximum and minimum joint limits, and  $w_{0,i}$  a weight factor for joint i.

Cruse et al. [117] intensively studied the control of arm movements in the horizontal plane. He observed that the strategies used by human subjects to control the shoulder, elbow and wrist could be simulated by assigning a cost function to each joint and selecting the arm configuration corresponding to the minimized sum of the costs. The cost functions appeared to consist of two parabolic branches that could have different slopes. The minimum of the cost function for respectively the horizontal flexion of the shoulder, elbow flexion and flexion of the wrist were  $0^\circ$ ,  $80^\circ$  and  $10^\circ$ , which are referred to as *minimum posture* angles. In our method, we simplified the joint cost functions to parabolic functions, which basically comes down to using the JRA criterion with minimum posture angles instead of center values:

$$w = \sum_{i=1}^n w_{0,i} \frac{(q_i - q_{mi})^2}{(q_{max,i} - q_{min,i})^2} \quad (4.4)$$

The minimum posture angles  $q_{mi}$  used in our method are listed in table 4.1.

## 4.2 Range of the robot

### 4.2.1 Approximation of the workspace

Before calculating a possible trajectory to the specified end-effector position, the possibility of reaching this position by the current configuration needs to be checked. To decide whether a certain position is reachable, the method uses an approximate calculation of the workspace. The workspace is modelled as

Table 4.1: Minimum posture (MP) values for the joint cost functions. The angles are defined in the reference frames connected to the human base model (see figure 2.2) and relative to the standard T-pose.

Block	BAU	Description	MP angle ( $^{\circ}$ )
Clavicle	7	Abduction/adduction of shoulder girdle	0
	8	Elevation/depression of shoulder girdle	0
Shoulder	9	Horizontal flexion/extension of shoulder	0
	10	Abduction/adduction of shoulder	70
	11	Inward/outward medial rotation	0
Elbow	12	Flexion/extension of elbow	80
Wrist	13	Pronation/supination of elbow	0
	14	Flexion/extension of wrist	0
	15	Abduction/adduction of wrist	0

a part of a hollow sphere whereof the origin coincides with the origin of the shoulder block base frame. The approximate workspace can then be described by using a maximum and minimum value for the three spherical coordinates specifying the sphere part. Figure 4.1 shows an example of a possible workspace of a right arm. All reachable points in the workspace are located between a minimum radius  $r_{min}$  and a maximum radius  $r_{max}$ . The polar angle  $\theta$  and azimuthal angle  $\phi$  are specified in a reference frame parallel to the standard reference frame, placed in the origin of the shoulder block. As for the radius, a maximum and minimum value is specified.

The six parameters specifying the workspace are calculated at the launch of the program.  $r_{max}$  corresponds to the maximum distance of the chain's end-effector with respect to the shoulder base frame (see figure 4.2). With other words, it is the length of the chain when placed in the T-pose minus the length of the clavicle links. Since the use of joints corresponding to the clavicle block is rare in today's robotics and in any case, the range of the corresponding joint angles is limited, resulting in a negligible contribution to the workspace compared to that of the shoulder block, the clavicle block is ignored in this calculation for simplicity reasons. A similar strategy is used for calculating the inner radius of the sphere;  $r_{min}$  is the minimal distance of the end-effector with respect to the shoulder base frame. This distance can be determined by selecting the angle for the elbow joint that results in a maximum flexion, next to the T-pose angles for the other joints, and calculating the distance between the shoulder base and hand end-effector (figure 4.3). To specify the minimum

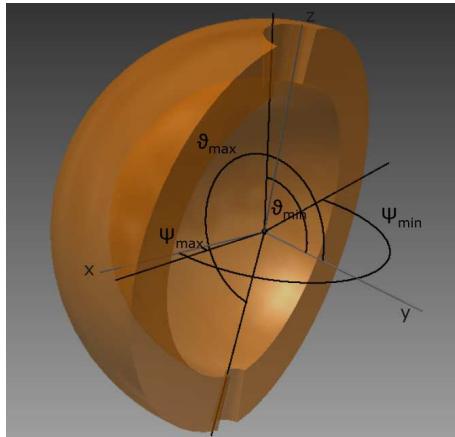


Figure 4.1: Example of an approximated workspace.

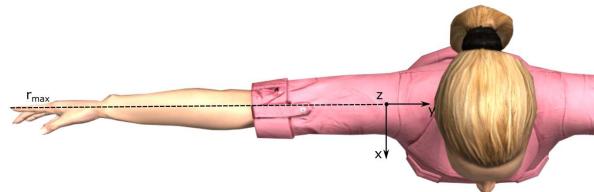


Figure 4.2: Determination of the maximum radius  $r_{max}$ : the maximum reachable distance of the end-effector, measured from the shoulder base frame origin.

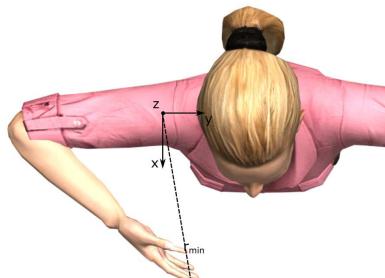


Figure 4.3: Determination of the minimum radius  $r_{min}$ : the elbow joint is placed in maximum flexion while the other joint angles correspond to the T-pose angles. The distance between the shoulder base frame and end-effector corresponds to  $r_{min}$ .

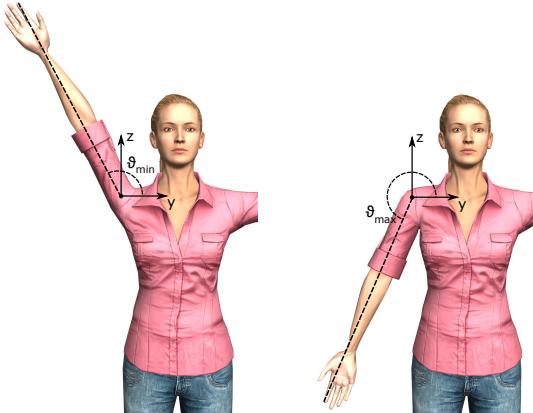


Figure 4.4: Calculation of the minimum and maximum value for  $\theta$ ; the arm is placed in respectively, maximum abduction and maximum adduction and the angle formed by the end-effector is calculated.

and maximum polar angle  $\theta$ , we respectively look at the effect of the maximum abduction and adduction of the shoulder joint on the position of the end-effector of the arm (figure 4.4). In a similar way, the minimum and maximum values for the azimuthal angle  $\phi$  is calculated by considering the maximum horizontal extention and flexion of the shoulder joint.

This approximation however includes a portion that is not included in the real workspace; when the shoulder approaches its maximum horizontal extension, not the whole area between the maximum and minimum radius can be reached. When observing the horizontal plane crossing the shoulder base frame, the points covered by a circle with the centre point located in the elbow base frame and radius equal to the length of the lower arm needs fall out the workspace (see figure 4.5). For most robots, the shoulder joint block is composed of two joints with an in-line axis, separated by a joint with an axis perpendicular to the link. In this case, the unreachable points in the 3D workspace are gathered by a sphere with an elbow-base centre point and a radius of the lower-arm length. Since this is the most common case, it is taken as a reference to calculate the approximate workspace. Therefore, next to the values for  $r$ ,  $\theta$  and  $\phi$ , also the length of the lower-arm is calculated and used in the determination of the range.

Figure 4.6 shows a  $xy-$  and  $xz-$  cross section of the workspace of NAO. The blue dots indicate the real workspace, while red dots indicate the calculated approximation. Uncovered blue dots in the left part of figure 4.6 result from not taking into account the configurations involving elbow flexion for maximum horizontal shoulder flexion/extension. The eliminated circle around the elbow is clearly visible in the right bottom corner of the  $xy-$  cross section (left of

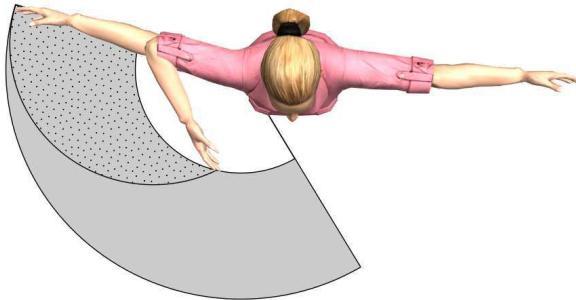


Figure 4.5: View of the calculated workspace in the horizontal plane crossing the shoulder base frame. The area covered by a circle with the centre point located in the elbow base frame and radius equal to the length of the lower arm (dotted surface) needs to be subtracted from the workspace (grey surface). N.b: this figure shows the calculated workspace used in the pointing-condition, where the end-effector is the finger.

figure 4.6). However, some blue dots are visible in this region. They origin from non-human like postures and do not contribute to proper natural trajectories. In the right part of figure 4.6, the posture of Nao reaching maximum extension is superimposed on the  $xz$ - cross section passing the shoulder base frame. For the specific configuration and joint limits of Nao, a small region of reachable points, highlighted by a black circle, is neglected by the approximate calculation. These are however points that are not of great interest for pointing and reaching gestures, since most probably, such a gesture is directed towards the space in front of the body. Other uncovered blue dots result from configurations involving elbow flexion for maximum flexion/extension and do not contribute to natural human postures.

#### 4.2.2 Evaluation of specified end-effector positions

Since only four variables are used to describe the approximate workspace, the method can rapidly evaluate if a certain end-effector position lies within the possible range of the robot. In order to do so, the parameters  $r$ ,  $\theta$  and  $\psi$  corresponding to the specified end-effector position need to be calculated. The radius  $r$  can easily be determined by calculating the norm of the vector starting at the shoulder base frame and ending in the specified point. By projecting this vector respectively in the  $yz$ -plane and the  $xy$ -plane, the angles  $\theta$  and  $\psi$  can be calculated. To check whether the point is in the range of the robot, these values are compared to the limit values of the approximate workspace. In case the desired position indeed lies within the hollow sphere-part, the method checks if the position is located inside the non-reachable sphere centred around

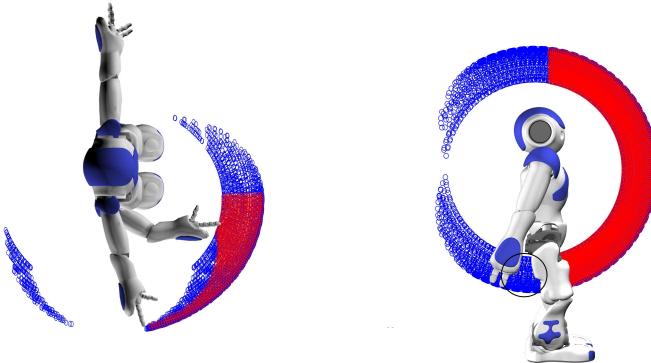


Figure 4.6: Calculating an approximate workspace. Blue dots indicate the real workspace, red dots the approximation. Left: xy-cross section passing the shoulder base frame. Uncovered blue dots in the top right result from not taking into account the configurations involving elbow flexion for maximum horizontal shoulder flexion. The same applies for the dots in the left bottom corner. The uncovered blue dots located in the circle around the elbow base (right bottom corner) origin from non-human like postures and therefore do not contribute to proper natural postures. Right: xz-cross section passing the shoulder base frame. A small region of reachable points, highlighted by a black circle, is neglected by the approximate calculation. Other uncovered blue dots result from configurations involving elbow flexion for maximum flexion/extension and do not contribute to natural human postures.

the elbow. In order to do this, the desired position is rewritten in the elbow base frame when positioned in maximum horizontal extension, and its norm is compared to the lower-arm length.

### 4.3 Trajectory generation

When the desired end-effector position is located within the workspace of the robot, a trajectory towards this position needs to be calculated. Different research has shown that, when asked to perform a point to point hand gesture, humans tend to move their hand along a straight path [118][119][120][121]. Therefore, a logical first trial for the path is a linear interpolation between the start and end position. However, even if the start and end point fall within the workspace, it is possible that a part of the trajectory falls outside the reachable range. For a human, this is for example the case when reaching a point close to the chest, starting from the T-pose. [122] reported that, when a subject was required to produce curved hand trajectories, a curve with low

curvature elements was tried to be approximated. To verify if a linear trajectory is possible, our developed method checks a set of points, distributed along the trajectory, to lie in the workspace. In case one of these points fall out of the range, a curved trajectory to reach the desired location is calculated. This trajectory consists of a circular arc connecting the start and end position of the end-effector, whereof the exact shape, i.e. the radius and mid-point corresponding to the circular arc, depends on in which amount the straight path is situated in the non-reachable zone. If only a small section of the straight path is not reachable by the configuration, a small correction with respect to the linear trajectory is possible. However, when a large portion of the straight path falls in the non-reachable zone, to be able to reach the desired end position, the corresponding arm chain needs to go around this zone, resulting in a path with higher curvature. Using this methodology, a trajectory with a minimal amount of curvature, as close as possible to the linear path, is calculated for the specified gesture. More information on how the trajectory is calculated in the gesture software is available in appendix B.

Figure 4.7 schematically summarises how a place-at gesture is calculated: firstly, the necessary end-effector position in the arm base frame is calculated, starting from the desired position specified by the user. After verifying the reachability of this point, a suitable trajectory is calculated. For every step in this trajectory, the joint angles can be determined by using the inverse kinematics algorithm discussed in section 4.1.

## 4.4 *Pointing* condition

The pointing condition is used in situations whereby the robot is aimed to point *towards* a certain specified position. In this case, no direct constraint is imposed on the end-effector; a series of configurations with a specific combination of end-effector position and orientation can fulfil the pointing constraint. In the pointing condition, the end-effector is the index finger, in contrast to the hand itself, as used in the place-at condition. In our software, the index finger is positioned in line with the hand and the length of the finger is added to the length of the last wrist-link in the corresponding DH-parameter. When pointing to an object, the index finger is directed towards the object. This implies that for a certain position of the end-effector, the orientation is chosen along the connection line between the object and the last wrist joint. Or with other words, the extension of the end-effector needs to pass the selected target position. This is illustrated in figure 4.8. To calculate all the possible postures that fulfil the pointing constraint, the end-effector is gradually virtually extended and the pointing position is imposed on the virtual end-effector. As such, the problem of finding the postures fulfilling the pointing constraint for the real configuration is reduced to finding a posture fulfilling a place-at constraint for a series of

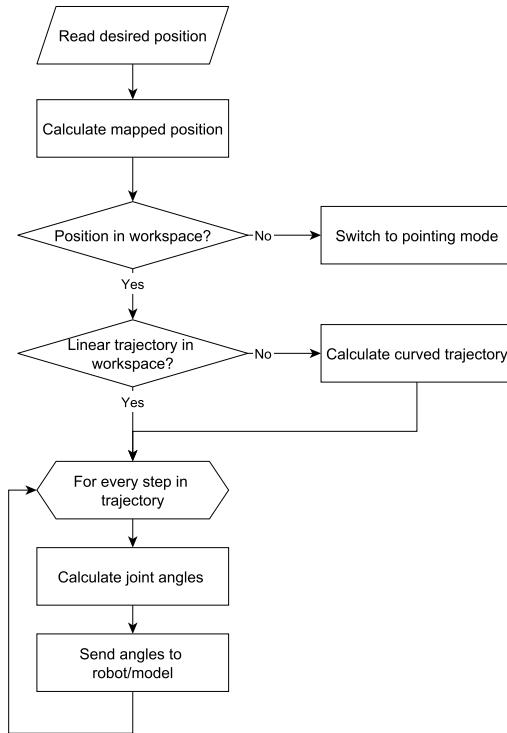


Figure 4.7: Schematic representation of the work flow for the place-at condition.

configurations, having a virtual end-effector with consecutive different lengths. The minimum and maximum amount of possible virtual extension depends on the robot's configuration and joint angle limits. For every virtual length, the optimal configuration is calculated using the algorithm discussed in section 4.1. The previously described cost function finally selects the optimal result by comparing the total cost of every configuration from the resulting collection of postures. Figure 4.9 gives a schematic representation of this process. When the optimal posture is selected, a trajectory towards the final (real) end-effector position is calculated and the joint angles for each step of the trajectory can be determined.

## 4.5 Gazing

The method supports the possibility of gazing towards a chosen position, or along with a deictic gesture imposed on the right or left arm. The position,

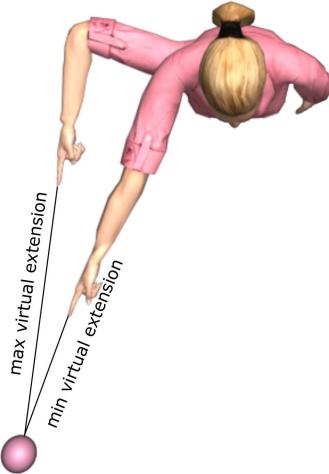


Figure 4.8: When pointing to an object, the index finger is directed towards the object. This implies that for a certain position of the end-effector, the extension of the end-effector needs to pass the selected target position. To calculate all the possible postures that fulfil the pointing constraint, the end-effector is gradually virtually extended and the pointing position is imposed on the virtual end-effector. As such, the problem of finding the postures fulfilling the pointing constraint for the real configuration is reduced to finding a posture fulfilling a place-at constraint for a series of configurations, having a virtual end-effector with consecutive different lengths.

specified in the standard reference frame by the user, should be converted to the necessary orientation the head should adopt to correctly gaze towards the desired direction. In order to do so, the position is first rewritten in the DH-end frame of the head chain, situated at the level of the eyes. This in contrast to the previous two conditions, where the position was converted to the base frame of the chain of interest. The vector, going from the origin of the end frame of the head chain towards the specified gazing position, serves as a basis to define a new coordinate frame, which will indicate the desired head orientation. More specifically, the vector indicates the direction of the  $x$ -axis. Since we don't aim for abduction/adduction of the neck, the  $y$ -axis is chosen to stay parallel to the standard  $xy$ -plane. By choosing the  $z$ -axis perpendicular to the previous two axes and assuring the vector is oriented upwards, the coordinate frame, and therefore, the necessary head orientation is unambiguously determined. The rotation matrix corresponding to this orientation is used as a constraint for calculating the correct joint angles using the the block mode, as explained in the previous chapter (chapter 3).

A simplified diagram of the complete work flow of the program is visualized

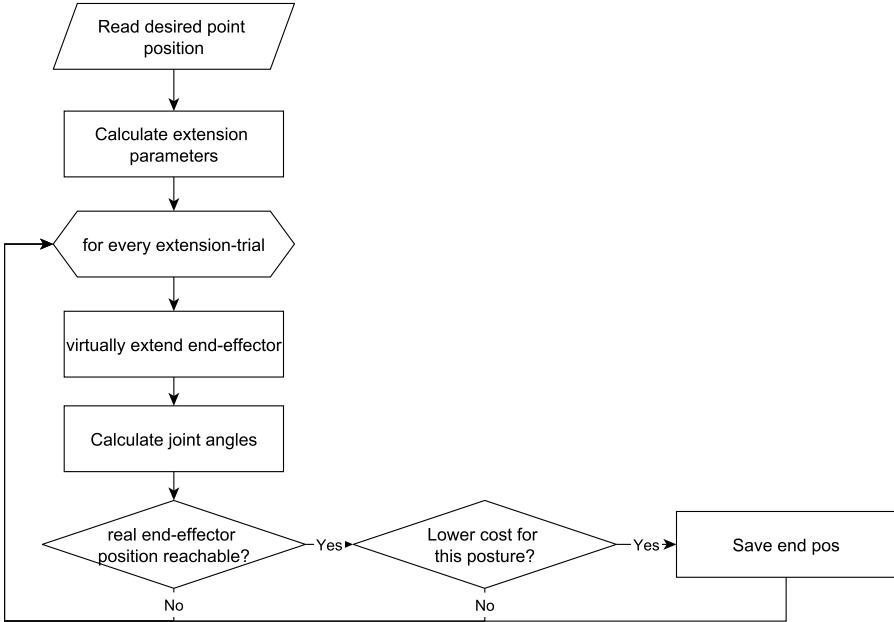


Figure 4.9: Schematic representation of the work flow for the pointing condition.

in figure 4.10. The method firstly verifies which mode the user would like to use. When using the block mode, the orientation information for the desired gesture is loaded from the database and mapped to the selected configuration, whereafter the corresponding joint angles are calculated for every intermediate posture. When using the end-effector mode, for every available chain, the method checks whether the user has inputted a desired position. In case a gazing position was specified, the corresponding head orientation is calculated. The calculation of the joint angles is further processed by the block mode. In case a position was specified for an arm chain, the method verifies which condition is enabled. When a pointing gesture is desired, the optimal end posture according to the principle of minimal deviation from the neutral posture is firstly determined, whereafter a suitable trajectory towards this posture is calculated. In case of a place-at condition, the trajectory to the mapped end-effector position is calculated immediately, provided that the position is situated in the workspace of the robot. If the position is not reachable by the robot, the pointing-condition will be enabled and a pointing gesture towards the position is calculated.

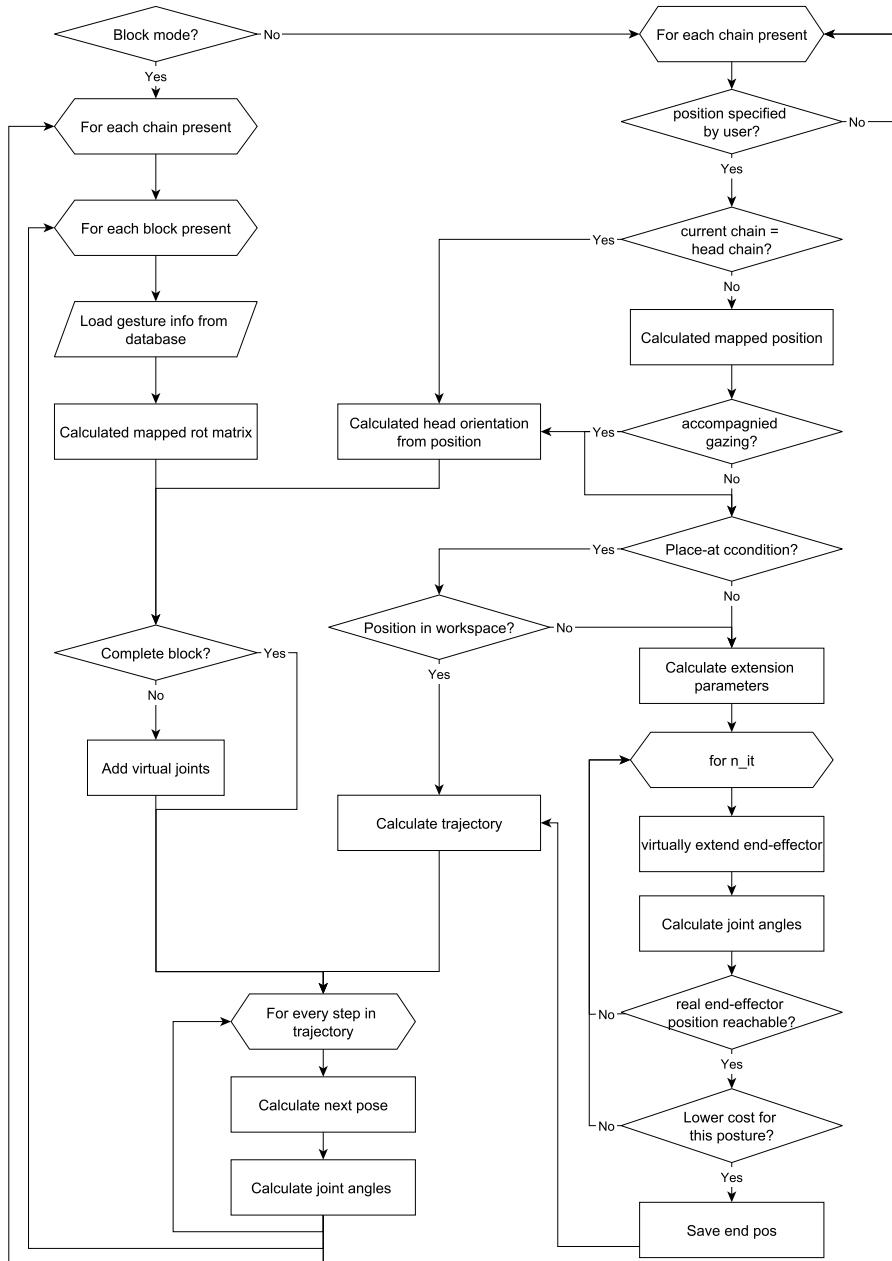


Figure 4.10: Simplified work flow of the complete method.

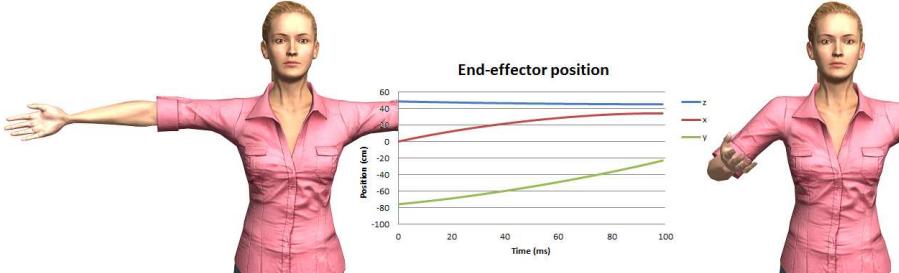


Figure 4.11: Calculated trajectory for a place-at task for the right hand with a start position of  $(0, 76, 48)$  cm and end position of  $(34, -23, 45)$  cm. Left: start pose. Right: end pose. Middle: plot of the calculated end-effector position with respect to the time.

## 4.6 Results

### 4.6.1 Results for the Place-at condition

The method was validated on different configurations. An example of the calculated trajectory for a place-at task for a 9 DOF arm is shown in figure 4.11. The arm consists of a 2 DOF clavicle, 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist. The initial pose (left side of the figure) corresponds to an end-effector position of  $(0, 76, 48)$  cm. The middle figure visualizes the calculated end-effector position with respect to the time when reaching for an end-effector position of  $(34, -23, 45)$  cm. The resulting end posture is shown at the right side of the figure. Figure 4.12 shows an  $xy$ -view of the same trajectory (blue line), superposed on the  $xy$ -cross section of the right arm workspace for the place-at condition (grey zone). As mentioned in section 4.3, a first attempt for the trajectory is a straight path. In this example however, the straight line between the start and end-effector position passes a non-reachable zone. Therefore, a curved trajectory was used to reach the desired end-effector position.

### 4.6.2 Results for the Pointing condition

As discussed in section 4.4, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function. In this section, we discuss a pointing gesture to the position  $(60, -20, 30)$  cm performed by the robot NAO [3] with the T-pose as the starting posture. Figure 4.13 shows the calculated end posture for the different iteration steps. The end-effector is virtually extended between a minimum and maximum value. The minimal

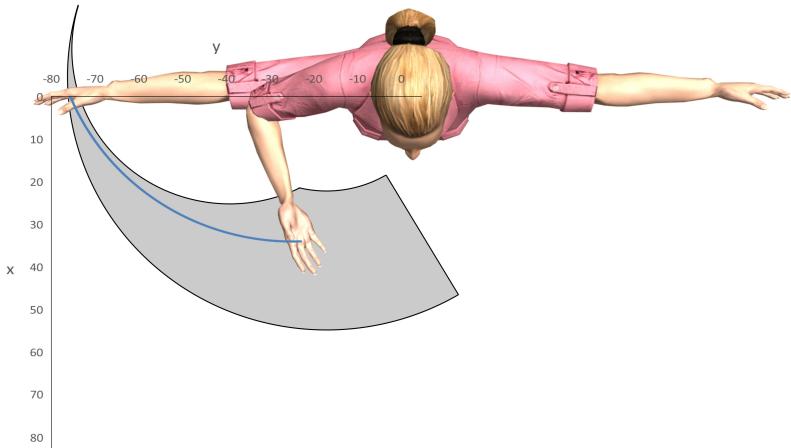


Figure 4.12: Top view of the calculated trajectory for a place-at task for the right hand with a start position of  $(0, 76, 48)$  cm and end position of  $(34, -23, 45)$  cm, superposed on the  $xy$ -cross section of the right arm workspace.

extension corresponds to the difference of the norm of the vector going from the shoulder base frame to the specified pointing position and the maximum length of the arm. The maximum extension, on the other hand, is the difference between the norm of this vector and the minimum length. Figure 4.13a shows the calculated end posture for the minimum virtual extension whereby the pointing position is visualized by a sphere. Figure 4.13d visualizes the end posture for the maximum virtual extension, while figure 4.13b and figure 4.13c correspond to two intermediate values of extension. The cost function selected posture (d) as the optimal end posture.

#### 4.6.3 Place-at condition imposed on different configurations

Table 4.2 shows the calculated end posture for a place-at gesture at  $(34, -34, 38)$  cm for four different configurations. The first column shows the joint configuration, while the second column shows the calculated posture for that configuration. The desired end-effector position is visualized by a sphere. In the top row, a 9 DOF human arm is shown, while configuration 2 shows the ASIMO robot [1]. For both ASIMO and the human model, the targeted end-effector position was reachable, and a suitable end posture could be calculated, as shown in the second column. Configuration 3 is that of the NAO robot [3]. NAO is considerably smaller than the previous models, and as a result, the maximum

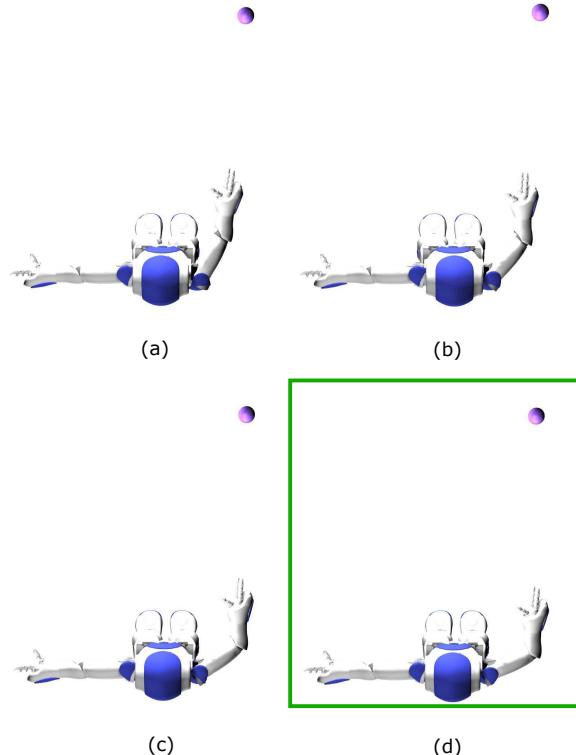
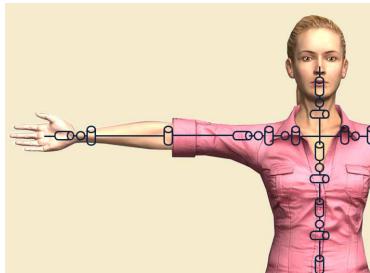
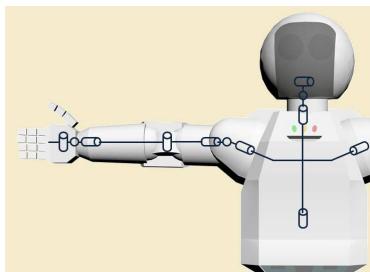
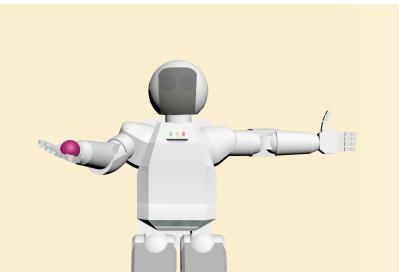
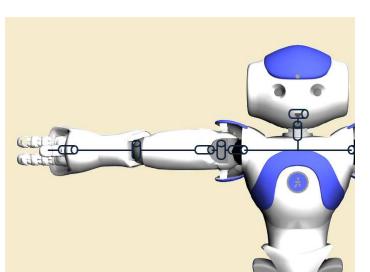
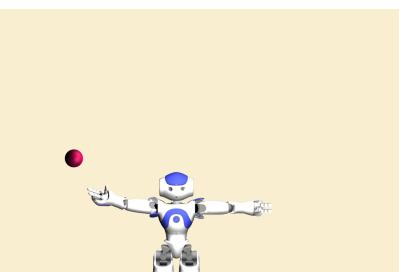


Figure 4.13: In the pointing condition, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function.(a) Minimum virtual extension. (d) Maximum virtual extension. (b) and (c) Intermediate values of extension. The cost function selected posture (d) as the optimal end posture.

Table 4.2: Results of the method for different arm configurations. The first column shows the joint configuration, while the second column shows the end posture for a place-at gesture at  $(34, -34, 38)$  cm.

	Configuration	Calculated posture
full config		
ASIMO		
NAO		

reachable distance is smaller. The desired position is located out of the range of the robot. Therefore, the pointing condition is activated, and a suitable posture for a pointing gesture towards the specified point is calculated.

## 4.7 Conclusions

In this chapter, the working principles and calculation aspects of the end-effector mode were covered. This mode is used for gestures whereby the position of the end-effector is crucial, like for deictic gestures, while the block mode, discussed in the previous chapter, was developed to calculate gestures whereby the overall pose is important, like for emotional expressions. To overcome the correspondence problem, the framework of the method is constructed independently of any configuration, and mappings are only calculated at run-time, based on the morphological information of a robot chosen by the user. The end-effector mode allows calculating postures for two different conditions; the place-at condition, whereby the user specifies the position of the hand, and the pointing condition, whereby the user specifies a pointing position towards the robot should point. The end-effector mode was validated on a set of configurations, including those of the robots NAO, ASIMO and Justin. The following chapter covers how the two working modes of the method can be mixed to generate blended gestures and how an initially neutral deictic gesture can be modulated into an affective gesture.

# 5 | Generating blended gestures and affective functional behaviors

Since for different types of gestures, different features are important, our method was designed to work in two modes. The *block mode* is used to calculate gestures whereby the overall arm placement is crucial, like for emotional expressions. The *end effector mode*, on the other hand, is developed for end-effector depending gestures, i.e. gestures whereby the position of the end-effector is important, like for manipulation and pointing. The working principles and results of the block and end-effector mode were presented in detail in the previous chapters. This chapter covers how the two developed modes are combined to generate blended deictic gestures and emotional expressions, and how information about the current emotional condition can be used to modify functional behaviors, calculated by the end-effector mode, into affective motions. The new implementations were validated on virtual models with different configurations, including those of the robots NAO and Justin.

This chapter is based on the following publication:

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefever, and Bram Vanderborght. Generic method for generating blended gestures and affective functional behaviors for social robots. *Autonomous Robots*, 1–12, 2017.

## 5.1 Blended gestures

### 5.1.1 Priority levels

In natural communication, humans use and combine different types of gestures. By combining the two modes of our method, it is possible to generate blended emotional expressions and deictic gestures. In order to do so, priority levels for each chain are assigned to both gesture types and a *mode mixer* was designed. If the mode mixer is turned off, all gestures are treated separately; starting a new gesture entails a previously started gesture to be aborted. By enabling the mode mixer, different gestures are blended by considering for every chain, only the end-effector condition(s) corresponding to the gesture with the highest priority level. The priority levels are defined using a number of rules:

- For an emotional expression, the priority level for each chain is set on level 1
- A deictic gesture has a higher priority than an emotional expression: the chain corresponding to the pointing/reaching arm receives a higher priority level; level 2
- Similarly, gazing has a higher priority than an emotional expression: the head chain receives a higher priority level; level 2

This is summarized in table 5.1.

For every separate chain, the highest priority level present determines which gesture needs to be considered for that chain. The corresponding calculation principles (see table 5.1) are enabled, and the required constraints are loaded for the different chains: orientational information for every block composing the chain for the block mode, or the desired end-effector position for the complete chain for the end-effector mode.

When, for example, an emotional expression is performed in combination with a left handed deictic gesture, the left arm chain has a level 1 priority for the emotional expression but a level 2 for the deictic gesture, as can be seen from table 5.1. Therefore, for that chain, the pointing position is imposed and the end-effector mode will calculate the corresponding joint angles. For the other chains, only priority level 1 is present. Therefore, the block mode will calculate the joint angles for all blocks in the remaining chains.

Figure 5.1 schematically summarizes how the mode mixer and the priority levels determine the imposed constraints, while figure 5.2 visualizes the work flow of one iteration, depending on the priority levels.

Table 5.1: Priority levels

	Emotional expression	Deictic gesture Left arm	Gazing Right arm
<b>priority level</b>			
Body	1		
Head	1		2
Arm right	1		2
Arm left	1	2	
<b>Calc. principle</b>	block mode	end-effector mode	block mode

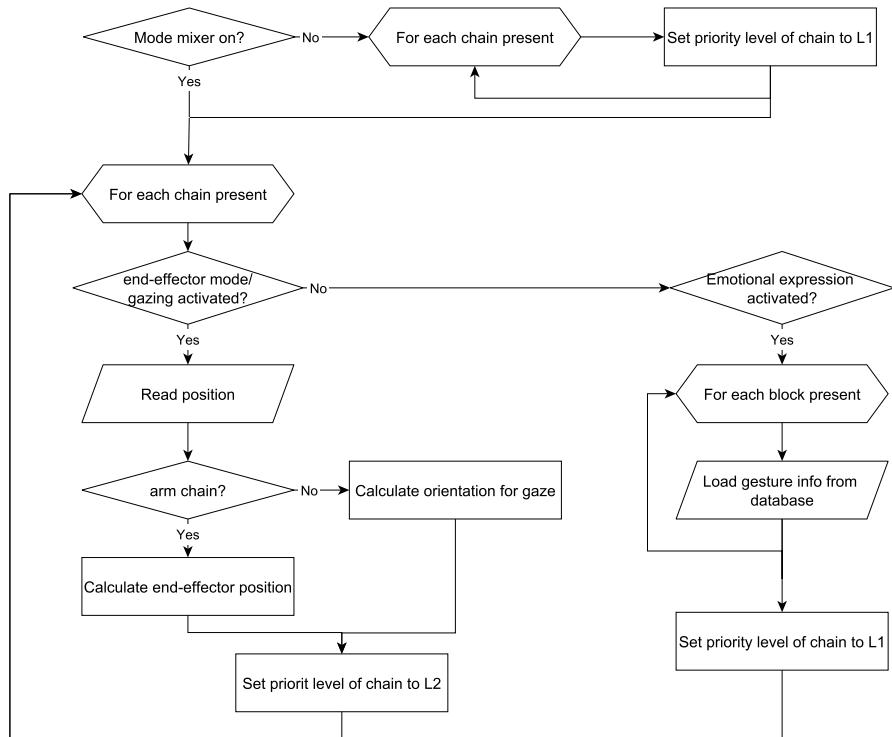


Figure 5.1: Schematic representation of how the end-effector constraints are determined by the motion mixer and the priority levels.

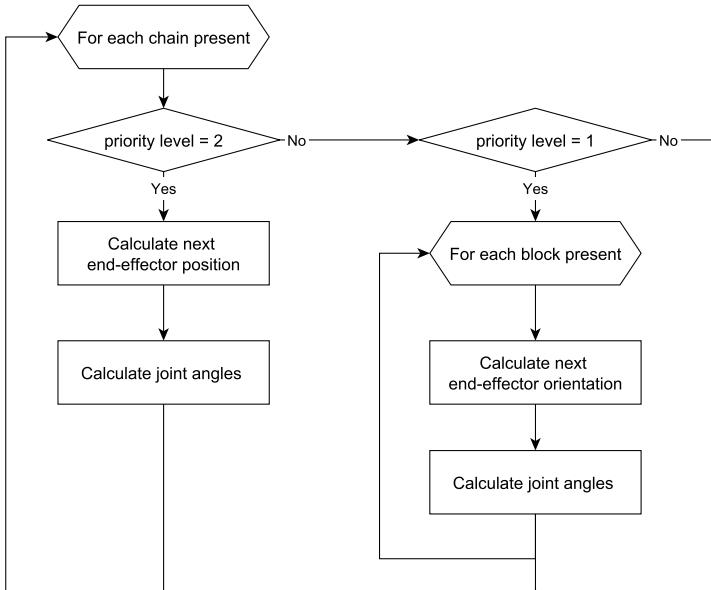


Figure 5.2: Work flow of one iteration, depending on the priority levels

### 5.1.2 Examples of blended gestures

Figure 5.3 illustrates the calculation of a blended gesture for both the robots NAO and Justin. The left part of the figure shows the joint configuration of the robots. NAO has an actuated head, and a left and right arm of 5 DOF. Justin's arms, on the other hand, contain 7 DOF, with a remarkable different configuration. In addition, Justin has an actuated body consisting of 3 DOF. The middle of figure 5.3 displays the end posture for the emotional expression of *fear*, calculated by the block mode. For the right part of figure 5.3, the mode mixer was enabled and a combination of gestures was demanded. Next to the expression of *fear*, a pointing gesture with the right arm was desired, accompanied by gazing towards the pointing location. As explained above, the priority levels determine which calculation principle is activated for every chain, and which corresponding end-effector conditions need to be used. For the emotional expression, all present chains have priority level 1. However, the priority of the pointing gesture for the right arm is higher since it has level 2. Therefore, for the right arm chain, the end-effector mode is activated, whereby the end-effector condition is determined by the desired pointing position. For all the other chains present, the block mode is activated. Since the priority of gazing towards a specified position for the head overrules that of the emotional expression, the necessary rotation matrix to obtain the desired gazing direction

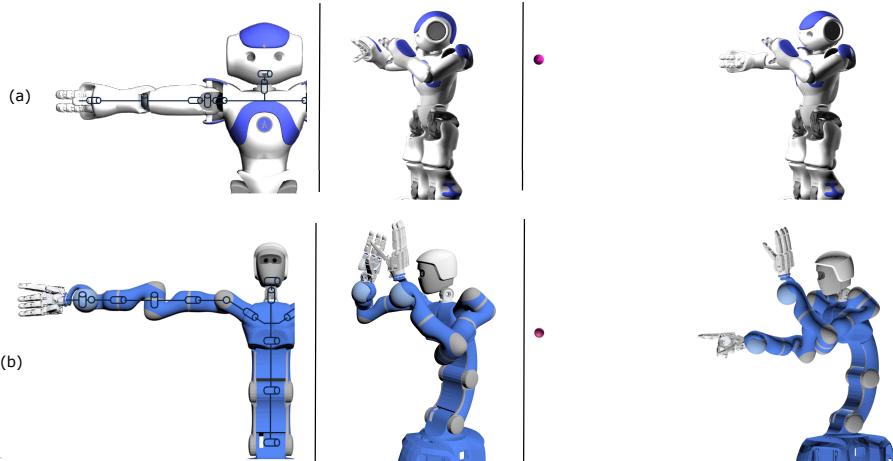


Figure 5.3: Example illustrating the calculation of a blended gesture for (a) NAO (b) Justin. Left: Joint configuration of the robot. Middle: Calculated end posture for the emotional expression of *fear*. Right: Calculated end posture for a combination of a pointing gesture with the right arm, and the emotional expression of *fear*.

is imposed. For the left arm chain, and body chain in case of Justin, the mapped rotation matrices, calculated using data from the gesture database, are imposed as end-effector condition for every present block in the corresponding chains.

## 5.2 Affective functional behaviors

### 5.2.1 Expressivity models

In some situations, it is desirable to express an emotional condition in a different manner than by using explicit bodily expressions as calculated by the block mode. It is possible, for example, that both arms are involved in a functional behavior, and therefore not available for performing an emotional expression. On the other hand, the recognizability of an emotional expression can decrease severely when one arm is used for a deictic gesture. In such cases it can be useful to express an emotional state through an ongoing functional behavior by modulating it, using a certain set of characteristic performance parameters. In literature, different expressivity models have been developed to reach that goal. Amaya et al. [123] proposed a model to generate an emotional animation from neutral motions by calculating an emotional transform based on the difference in speed and spatial amplitude of a neutral and emotional motion. In [124], six parameters, namely spatial extent, temporal extent, fluidity, power, overall

activation and repetition were used to modify behavior animations for the virtual agent Greta. Yamaguchi et al. [125] found that the amplitude, position and speed are relevant parameters in modifying basic motions to express joy, sadness, angeriness and fear, while Lin et al. [126] found that the stiffness, speed and spacial extent of the motion can effectively generate emotional animations from an initial neutral motion. Xu et al. [127] proposed a method for bodily mood expression, whereby a set of pose and motion parameters modulate the appearance of an ongoing functional behavior. Results indicated that the spatial extent parameters, including hand-height and amplitude, head position and the motion speed are the most important parameters for readable mood expressions [128]. Since in all these discussed expressivity models, the motion speed and the amplitude are important recurring factors, we decided to focus on these modification parameters in our method.

### 5.2.2 Generating affective gestures by influencing the motion speed

In both [128] and [125], it was experimentally confirmed that the motion speed influences the perceived level of both valence and arousal; a fast motion is associated with a high arousal and valence, while a slow motion is attributed to low arousal and valence values. By considering the emotions as vectors in the two dimensional emotion space of valance and arousal, based on the circumplex model of affect [129], we obtained an appropriate speed scaling factor for each emotion (see figure 5.4). When calculating a deictic gesture with the end-effector mode of our method, a suitable trajectory between the initial posture and the end posture is generated by calculating intermediate key frames. The timing between two consecutive frames is fixed, but the amount of frames, and therefore the total duration of the gesture is determined by the speed scaling factor. As such, affectional content is added to the deictic gesture by influencing the motion speed.

### 5.2.3 Generating affective postures using the nullspace

The second modification parameter, the amplitude of the motion, refers to the spatial extent; the amount of space occupied by the body. Xu et al. [128] found that this parameter is only related to the valence; open postures with a high amplitude are coupled with affective states with high valence, while closed, low amplitude postures are related to states with a low valence. Also this relation is represented on the circumplex model of affect in figure 5.4. As discussed in section 4.1, the necessary joint angles to reach a desired posture are calculated by the inverse kinematics algorithm of equation B.24 with as cost function  $w$ , a slightly adapted form of the joint range availability criterion (equation 4.4).

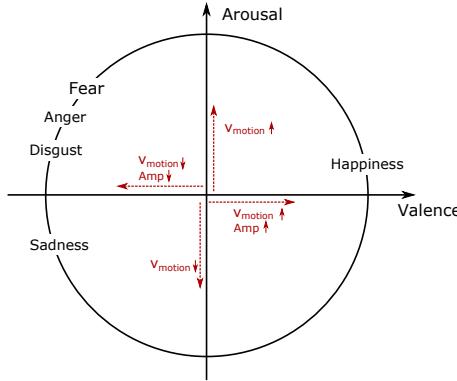


Figure 5.4: Dependency of the modification factors *motion speed* ( $v_{motion}$ ) and *Amplitude* ( $Amp$ ) on the valence and arousal value, depicted on the circumplex model of affect ([129]).

In that way, an optimal humanlike posture is calculated by keeping the joints  $q_i$  close to a selected set of minimum posture angles  $q_{mi}$ . Instead of using the fixed minimum posture angles, it is possible to express them as a function of the current valence level. Hence, the resulting calculated posture becomes dependent of the current affective state. The Body Action Units mostly influencing the openness of a posture are BAU 10 and 13; the units corresponding to the abduction/adduction of the shoulder and the flexion/extension of the elbow joint (see table 2.2). For the joints corresponding to these BAU's, a linear function of the valence is provided instead of the fixed minimum posture angle as used before. When scaling the valence level  $val$  for each emotion as read on the circumplex model of affect (see figure 5.4) between 0 and 1, the following linear function can be used to select the current appropriate value for the minimum posture angle, which we now call the *affective posture angle*  $q_{ai}$ :

$$q_{ai} = q_{ai,min} + val * (q_{ai,max} - q_{ai,min}) \quad (5.1)$$

The minimum value  $q_{ai,min}$  of the affective posture angle corresponds to the value associated to the minimum valence value, i.e. a value generating a closed posture with low amplitude. The angle value is defined in the corresponding reference frame connected to the human base model, and relatively to the T-pose as visualized in figure 2.2. Therefore, for BAU 10, a value of  $90^\circ$  is a suitable choice, since it corresponds to a posture whereby the upper arm is touching the flank of the body. Regarding BAU 13, a small amplitude posture is reached when keeping the forearm as close as possible to the upper arm. A value of  $170^\circ$  is therefore an appropriate choice. Similarly, the maximum value  $q_{ai,max}$  of the affective posture angle corresponds to the value associated to

the maximum valence value; the value generating an open posture with high amplitude. This should be a posture whereby both the elbow and wrist are located far away from the body. A suitable choice is therefore  $0^\circ$  for BAU 10, and  $80^\circ$  for BAU 13.

#### 5.2.4 Example: deictic gesture during different states of affect

Figure 5.5 illustrates the results of the two subsections discussed above. A right-arm reaching gesture during different states of affect was calculated for two different configurations; a human virtual model with a 9 DOF arm, depicted in figure 5.5a , and the robot Justin, depicted in figure 5.5b.

The right top of both sub figures shows the corresponding joint configuration of the used models, while the main of the figures visualizes a set of calculated postures for every affective state on a time line. As discussed in subsection 5.2.2, the total timing of the gesture is influenced by the speed factor, of which the value is determined by the current affective state. Since the motion speed increases with both valence and arousal, a high value is obtained for the happy state, resulting in a short total timing of the gesture of  $0.75\text{ s}$ . For the same pointing gesture performed during a sad state, a low speed factor and thus long duration ( $1.5\text{ s}$ ) is calculated, while for the fearful state, the values are located somewhere in between (duration of  $1.0\text{ s}$ ).

The influence of the amplitude modification factor is visible when comparing the end posture for each affective state. Since the amplitude of the posture increases with higher valence values, an open posture is calculated for the happy state, whereby the elbow is located far away from the body. For the sad state, the elbow is placed close to the body, generating a closed posture as expected. Since the valence values for fear and sadness are close to each other, as can be seen from figure 5.4, the difference in posture during the the corresponding states is small (a difference of approximate  $10^\circ$  for BAU 10). Here, the total timing of the gesture is the main differentiator.

### 5.3 Conclusions

In this chapter we discussed how the two working modes of the method can be combined to generate blended emotional expressions and deictic gestures. To achieve this, a mode mixer was developed, and for every mode, priority levels were assigned to each chain. The priority levels decide which end-effector constraints need to be considered for each chain. In that way, when gestures with different priority levels are selected with the mode mixer enabled, the imposed end-effector conditions originating from the different gestures result

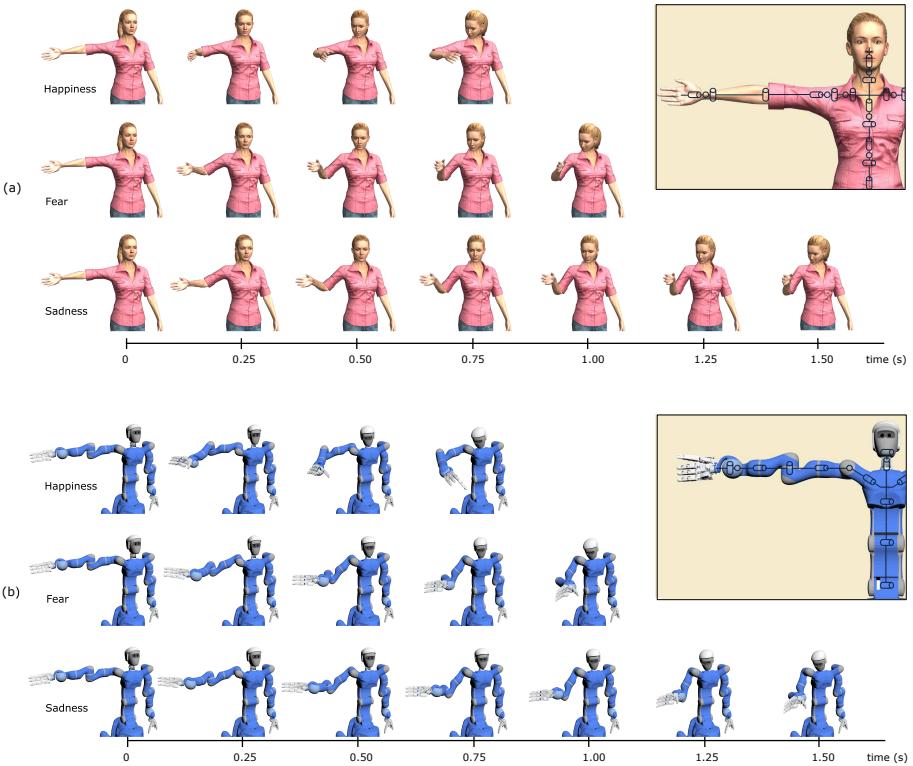


Figure 5.5: Example illustrating expressing affect during a functional behavior for (a) a human virtual model (b) the robot Justin. A reaching gesture was calculated during different affective states: happiness, fear and sadness. Right top: corresponding joint configuration. Main figure: Time line showing a set of postures for every affective state, illustrating the effect of the motion speed modification factor on the calculated gesture. The effect of the amplitude modification factor is visible when comparing the end postures for every mood.

in a blended posture. In some cases, it can be interesting to express an emotional condition not by using explicit bodily expressions as calculated by the block mode, but through an ongoing functional behavior. We implemented the possibility of modulating a pointing or reaching gesture into an affective gesture by influencing the motion speed and amplitude of the posture. The new functionalities of the method were illustrated using several virtual models, including those of Justin and NAO. The next chapter covers the validation of the method on physical robots.

# 6 | Generating gestures for physical robots

To overcome the difficulties in transferring joint trajectories to different robots, we proposed the use of a generic method to calculate gestures for social robots. The developed software allows the calculation of different types of gestures, including emotional expressions and deictic gestures, as well as combinations of both types and mood expressions through functional behaviors. In the previous chapters, the different modalities were discussed and validated on the virtual model of different robots. In this chapter, the innovations made to the method to successfully use it on physical robots are discussed. This includes the implementation of an inverse kinematics algorithm with a joint angle limitation module. The selection of the necessary optimal parameters for our method is illustrated through an example. Furthermore, a joint speed limitation module was added to the method to guarantee a smooth performance of the calculated joint trajectories. For the validation, a test scenario including different types of gestures was generated for a set of robots with different morphologies, namely NAO, Pepper and Romeo.

## 6.1 Adjusting the method for physical robots

### 6.1.1 Joint angle limits

When using the method on physical robots instead of on virtual models, mechanical constraints become important issues to consider. A first factor that can entail major implications on the calculated gestures are the joint limits of the considered robot. A joint angle limitation module needs to be implemented in the algorithm to enable the physical robot to perform the calculated gesture without violating the physical constraints. Different strategies have been used in literature to implement joint limitation in existing algorithms. A well known method to avoid mechanical joint limits for redundant manipulators is the Gradient Projection Method (GPM) introduced by Liégeois [130]. Here, null space motion is used to guide the joint angles away from their limits. When applied on a closed loop inverse kinematics (CLIK) algorithm, this is similar as what we use to guide the configuration into a natural (section 4.1.2) or affective posture (section 5.2.3). The only difference is the choice of the cost function  $w(q)$ . To guide the calculated joint angles away from their boundaries,  $w(q)$  is specified as a function of the distance from the joint limits. In case of a redundant manipulator, this method guides the solution to the center of the joint range, away from the limits, but does not guarantee that no limit is crossed. Furthermore, since our gesture method is aimed to work for any robot configuration, including non-redundant ones and in both the block- end end-effector mode, this is not a satisfying solution. We opted to work with an algorithm proposed by Drexler and Harmati [131]. Their methodology guarantees no violation of the joint limits by transforming the joint variables  $q_i$  to a set of fictive joint variables  $z_i$ . The transformation for every joint should be continuous, monotonously increasing and open on the interval between the lower and upper limit value so that can be written:

$$q_i = \beta_i(z_i) \quad (6.1)$$

whereby the domain of  $\beta$  equals  $[-\infty, \infty]$ , whereas its range is  $[q_{min,i}, q_{max,i}]$ . A proposed transformation is the tangent function, whereby a linear mapping scales the range to the appropriate limit values:

$$q_i = \frac{q_{max,i} - q_{min,i}}{\pi} \tan^{-1}(z_i) + \frac{q_{max,i} + q_{min,i}}{2} \quad (6.2)$$

By expressing the kinematic equations in terms of the fictive variables  $z_i$ , and calculating the real joint values  $q_i$  by equation 6.1, the resulting values will always stay between the imposed boundaries. Their proposed algorithm works as follows: in a first step, the joint velocities are calculated in a conventional

way, in our case by using equation B.24, whereafter they are transferred to the transformed joint space:

$$\dot{z} = d\beta^{-1}(z)\dot{q} \quad (6.3)$$

with  $d\beta$  the diagonal matrix formed by  $d\beta_i = \frac{\partial \beta_i(z_i)}{\partial z_i}$ . Then, the transformed variables  $z$  are calculated by integrating  $\dot{z}$ . In our case, this is done by using the implemented Runge-Kutta algorithm. Finally, the joint angles  $q$  can be acquired by using equation 6.1. Using this technique, the joint limits cannot be violated. However, when reaching a boundary, the derivative of the corresponding function  $\beta_i$  approaches 0, causing the problem to get ill-conditioned. To invert the matrix  $d\beta$ , one proposed method is to use the Pseudoinverse  $d\beta^\dagger$  based on singular value decomposition with truncation at low singular values:

$$d\beta_{ii}^\dagger = \begin{cases} \frac{1}{\frac{\partial \beta_i}{\partial z_i}} & \text{if } \frac{\partial \beta_i}{\partial z_i} \geq \epsilon \\ 0 & \text{else} \end{cases} \quad (6.4)$$

To regain manipulability in such a situation, a secondary task vector  $y$  in the transformed joint space is introduced, that aims to drive the joint away from the boundary. So instead of using equation 6.3 to calculate the transformed joint derivatives, an extended formula is used:

$$\dot{z} = d\beta^\dagger(z)\dot{q} + \left( I - d\beta^\dagger(z)J_A^\dagger J_A d\beta(z) \right) y \quad (6.5)$$

The two terms in this equation are in what follows denoted as respectively  $\dot{z}_1$  and  $\dot{z}_2$ :

$$\dot{z} = \dot{z}_1 + \dot{z}_2$$

$$\text{with } \begin{cases} \dot{z}_1 = d\beta^\dagger(z)\dot{q} \\ \dot{z}_2 = \left( I - d\beta^\dagger(z)J_A^\dagger J_A d\beta(z) \right) y \end{cases} \quad (6.6)$$

#### 6.1.1.1 Determination of the optimal task vector $y$ for our method

In [131], a linear function of  $z$  is proposed for the task vector  $y$ , which is only activated in case of low singular values:

$$y_i = \begin{cases} -m_i z_i & \text{if } d\beta_i < \epsilon \\ 0 & \text{else} \end{cases} \quad (6.7)$$

with  $m_i$  a suitable weight factor for joint i. By defining the task vector using the same parameter  $\epsilon$  as used as truncation bound for the calculation of the

pseudoinverse  $d\beta^\dagger$ , the second part of equation 6.5 ( $\dot{z}_2$ ), which is responsible for guiding the joint away from its limits, is only activated when the first part ( $\dot{z}_1$ ), responsible for guiding the joint towards the necessary value to reach the desired end-effector pose, equals zero and vice versa.

The illustrative example in [131] proposes the following values for the constants:  $m_i = 1$ ,  $\epsilon = 10^{-10}$ . For our first attempts, the same parameters were implemented in our method. This however did not give optimal results for our method. Since  $\epsilon$  is very small, the joint angles are allowed to approximate the joint limits very closely, generating very high values for  $z$ . Depending on the trajectory, this can result in big jumps of joint angle values when the responsible term for the calculation of  $\dot{z}$  switches from the second to the first part of equation 6.5. An illustrating example of this is shown in the following figures. Consider the joint configuration shown in figure 6.1. For this example we only consider the right arm chain, which consists of a clavicle block composed of 2 joints, a shoulder and wrist block, both consisting of 3 joints, and an elbow block, composed of 1 joint. The desired trajectory is as follows; the model's starting posture is the T-pose. From there, the emotional expression of *sadness* is activated, a gesture with a total duration of 1.5 seconds. Immediately after reaching the final posture for this gesture, the emotional expression of *happiness* is imposed. This gesture has a duration of 0.75 seconds. Both gestures are calculated by the block mode. The end-effector orientations for every block necessary to reach the end posture of both expressions are calculated by combining information from the database with the morphological specifications entered by the user, as explained in chapter 3. The path between the start and end posture is determined by interpolating between the corresponding orientations for the total duration of the gesture. For every key frame, the necessary joint angles to reach the desired posture can then be calculated using the joint-constrained inverse kinematics algorithm, in combination with Runga Kutta. The trajectory of the first clavicle joint is visualized in figure 6.2a, while figure 6.2b shows the calculated trajectory of the corresponding transformed joint variable  $z$ , together with the values of  $\dot{z}_1$  and  $\dot{z}_2$ . From the starting point until  $t = 1.27$  s, the first part of equation 6.5,  $\dot{z}_1$ , pushes the joint angle value  $q$  from its initial value towards the upper joint limit. When reaching the joint limit, the values for  $\dot{z}_1$  become very large since  $d\beta$  approaches zero, which results in a corresponding large value for  $z$ . The point where  $\frac{\partial \beta_1}{\partial z_1} = \epsilon$  is depicted by a red dot on figure 6.2b. There, the pseudo-inverse  $d\beta^\dagger$  is set to zero, causing the result  $\dot{z}$  no longer be determined by term  $\dot{z}_1$ , but by term  $\dot{z}_2$ . The algorithm successfully keeps the joint at its boundary, while gradually decreasing  $z$ . At  $t = 1.5$  s, the emotional expression of *happiness* is activated. The term  $\dot{z}_2$  continues in lowering  $z$  until  $\frac{\partial \beta_1}{\partial z_1}$  again equals  $\epsilon$ . At this point, denoted by a purple dot,  $\dot{z}_1$  is again activated, trying to guide the joint angle to a value necessary to reach the desired end-effector orientation. Here, the algorithm fails; because of the current high value of  $z$ , and therefore high value of  $\beta^\dagger$ ,  $\dot{z}_1$  immediately

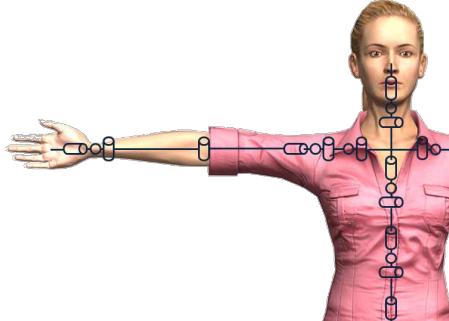


Figure 6.1: Joint configuration of the model used for the example in figure 6.2, 6.3 and 6.4. Relevant for this example is the 9 DOF right arm, consisting of a clavicle block (2 joints), a shoulder block (3 joints), an elbow block (1 joint) and a wrist block (3 joints).

drives  $z$  to a high negative value, projecting the joint angle  $q$  from its upper boundary to its lower boundary. This results in a direct switch of activation from  $z_1$  to  $z_2$  to prevent the joint angle  $q$  from crossing its lower limit value. The algorithm doesn't manage to lower the value of  $z$  enough to reactivate  $z_1$ . The desired end-effector orientation of the clavicle block is therefore not reached.

In order to solve this problem, a first possibility is to decrease the value of  $\epsilon$ . Figure 6.3a shows the calculated joint trajectory for the first clavicle joint for exactly the same configuration, desired gestures and parameters except for  $\epsilon$ , which is now set to  $10^{-3}$ . The corresponding transformed joint trajectories and the contributions of  $z_1$  and  $z_2$  are visualized in figure 6.3b. The initial trajectory of  $z$  is similar as for the previous example, however, the since  $\epsilon$  is smaller, the term  $z_2$  is activated considerably sooner. This point is again denoted by a red dot. The joint angles  $q$  are forced to keep a bigger distance from the upper boundary, and the corresponding  $z$ -values will stay significantly smaller. For the resulting duration of the *sadness*-gesture, the joint values are kept around this value by switching between the two contribution terms of  $\dot{z}$ ; firstly,  $z_2$  will attempt to lower the value of  $z$ , until the  $\frac{\partial \beta_1}{\partial z_1}$  again equals  $\epsilon$ . Then,  $z_1$  will continue to try to guide the joint to a value corresponding to the desired end-effector orientation, which results in a slight increase of  $z$ . This alternation continues until the second gesture, the emotional expression of *happiness*, is activated. The point where the  $z$ -value is low enough to activate  $z_1$  again lies at time  $t = 1.54$  s and is denoted by a purple dot on figure 6.3. From there, the algorithm guides the joint angle towards a lower value, corresponding to the necessary value to fullfil the current end-effector constraint. In contrast to the previous example, the desired end orientation could be reached by the

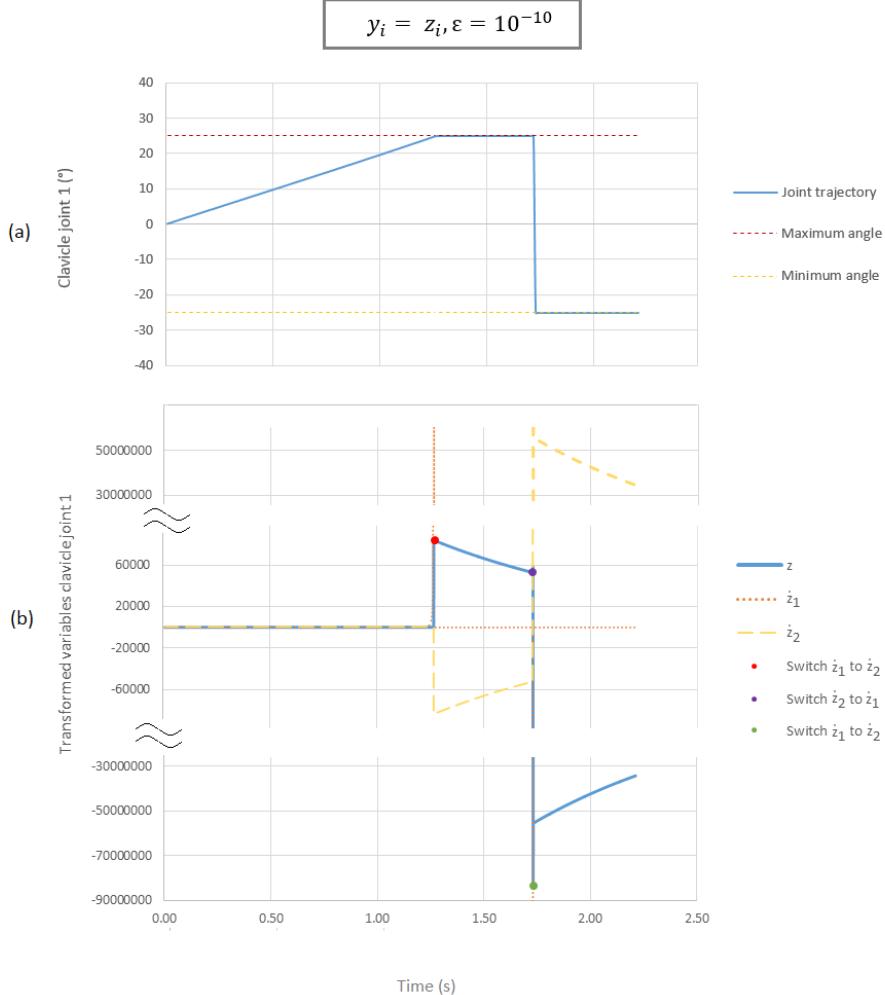


Figure 6.2: Calculated trajectory for the first clavicle joint of the configuration visualized in figure 6.1, for the execution of the emotional expression for *sadness*, followed by that for *happiness*.  $y_i = -z_i$  and  $\epsilon = 10^{-10}$ . (a) Trajectory for the real joint angle  $q$ . (b) Trajectory for the transformed joint variable  $z$ , accompanied by the contributions of  $\dot{z}_1$  and  $\dot{z}_2$ .

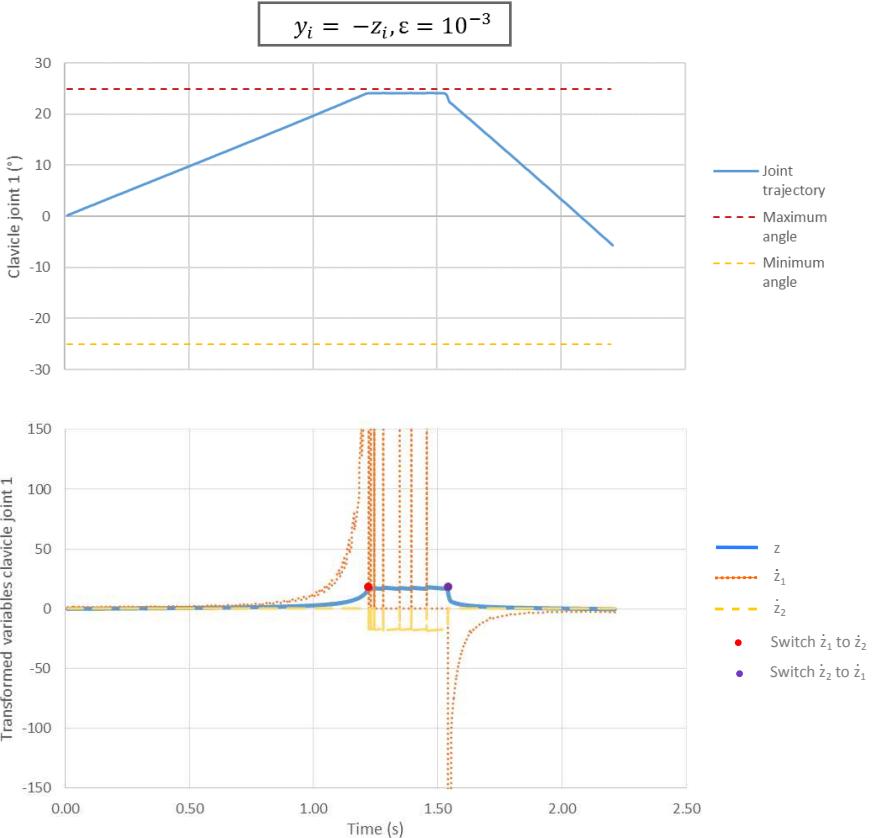


Figure 6.3: Calculated trajectories for the first clavicle joint of the configuration visualized in figure 6.1, for the execution of the emotional expression for *sadness*, followed by that for *happiness*.  $y_i = -z_i$  and  $\epsilon = 10^{-3}$ . (a) Trajectory for the real joint angle  $q$ . (b) Trajectory for the transformed joint variable  $z$ , accompanied by the contributions of  $\dot{z}_1$  and  $\dot{z}_2$ .

algorithm in this case, resulting in a correct solution for the joint angles.

One minor feature that could still be improved is the reaction time necessary to respond to a change in end-effector constraint when locked around a joint limit. As can be noted from figure 6.3, 40 ms are necessary for the algorithm to lower the  $z-$  value under the threshold to activate  $\dot{z}_1$  after starting the expression of *happiness* at  $t = 1.5$  s. Since a tangent function (equation 6.2) was used to serve as transformation  $\beta$ , its differential is proportional to the inverse of  $z^2$ :

$$\frac{\partial \beta_i(z_i)}{\partial z_i} = \frac{q_{max,i} - q_{min,i}}{\pi} \frac{1}{1 + z_i^2} \quad (6.8)$$

$\dot{z}_1$  is therefore proportional to  $z^2$ , while  $\dot{z}_2$  is only proportional to  $z$ , when selecting equation 6.7 for the task vector  $y$ . When, as in the previous example, an alternation between  $\dot{z}_1$  and  $\dot{z}_2$  exists to keep the joint angle close to its boundary, more iterations are necessary to guide  $z$  to its threshold to switch from  $\dot{z}_2$  to  $\dot{z}_1$  then vice versa. This is clearly visible in the region between the two dots in figure 6.3. To solve this issue, the following function can be used for  $y$  instead of equation 6.7:

$$y_i = \begin{cases} -\text{sign}(z_i)k_i z_i^2 & \text{if } d\beta_i < \epsilon \\ 0 & \text{else} \end{cases} \quad (6.9)$$

In figure 6.4, the calculated trajectories are plotted for the same example as before, but with using the alternative function for the task vector. Since both  $\dot{z}_1$  and  $\dot{z}_2$  are now proportional to  $z^2$ , the alternations in contributing factors for  $\dot{z}$  follow each other in a similar time span. Therefore, when activating the emotional expression for *happiness*, a reasonable shorter time is necessary to push the joint angle away from its limit and guide them towards the necessary value corresponding to the desired end-effector orientation.

### 6.1.2 Joint speed limits

A second important limitation factor to take into account when working with physical robots are the joint speeds. To ensure the speeds to stay within their boundaries, a saturation on the joint speed calculated by equation B.24,  $\dot{q}_{calc}$ , is included in the algorithm:

$$\dot{q} = \begin{cases} \dot{q}_{calc} & \text{if } -\dot{q}_{max} \leq \dot{q}_{calc} \leq \dot{q}_{max} \\ \text{sign}(\dot{q}_{calc})\dot{q}_{max} & \text{else} \end{cases} \quad (6.10)$$

As discussed in section 3.2, the time span in which an emotional expression should be finished is specified in the database. For pointing and reaching, the total duration of the gesture is dependent of the current affective state (see section 5.2.2). However, when limiting the joint speeds  $\dot{q}$ , it is possible that the desired end pose cannot be reached in the specified time span  $t_{end}$ . In order to give the algorithm the possibility of reaching the desired posture, if necessary, the reference time span is extended until the calculated joint angles have converged. For time steps exceeding the reference time span, the desired end-effector pose is kept to its desired final value:

$$x_d(t_k) = x_{d,t_{end}} \quad \text{if } \begin{cases} t_k > t_{end} \\ \text{Abs}(x_{e,t_k} - x_{e,t_{k-1}}) > \text{error} \end{cases} \quad (6.11)$$

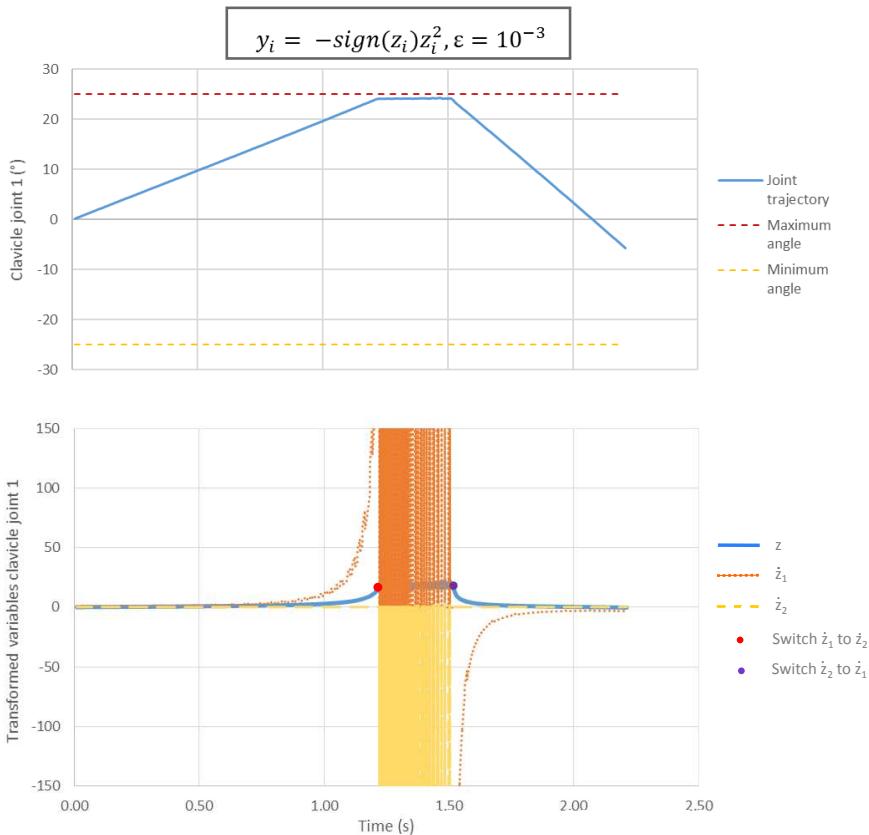


Figure 6.4: Calculated trajectory for the first clavicle joint of the configuration visualized in figure 6.1, for the execution of the emotional expression for *sadness*, followed by that for *happiness*.  $y_i = -\text{sign}(z_i)z_i^2$  and  $\epsilon = 10^{-3}$ . (a) Trajectory for the real joint angle  $q$ . (b) Trajectory for the transformed joint variable  $z$ , accompanied by the contributions of  $\dot{z}_1$  and  $\dot{z}_2$ .

To illustrate the effect of the joint speed limitation in the algorithm, a right-handed reaching gesture for the robot Romeo was calculated. Table 6.1 contains the necessary robot specifications that serve as input for the method. The top left shows the joint configuration of the robot. Relevant for this example is the right arm chain, consisting of a 3 DOF shoulder block, 1 DOF elbow and 3 DOF wrist. In the bottom right, the speed limits for the arm joints are listed. When calculating the specified gesture with the joint speed limitation disabled, the limit for the first wrist joint was crossed. Figure 6.5a shows the calculated speeds  $\dot{q}$  for that joint, together with its boundaries, while figure 6.5b shows the corresponding end-effector trajectory. The full line denotes the calculated end-effector position  $x_{i,e}$ , while the dotted line shows the desired end-effector position  $x_{i,d}$ , calculated by the trajectory function (see section 4.3) and imposed on the inverse kinematics algorithm. Except for the  $x$ - coordinate at the very start of the gesture, the calculated end-effector position follows the desired trajectory perfectly.

Figure 6.6 visualizes the same quantities for the same reaching gesture, but now calculated with the joint speed limitation enabled. Figure 6.6a shows how the speeds for the first wrist joint are kept within the imposed boundaries. As for the case without limitation of the speeds, the desired trajectory is followed very closely as can be noted from figure 6.6b. For this gesture example, the algorithm succeeds in calculating a solution for the desired gesture in the imposed time span, so no time extension is necessary.

## 6.2 Experimental results on physical robots

To illustrate the capabilities of our developed method, a set of gestures was created for different physical robots. To provide context to the gestures, they were integrated into a little story told by the robot. To highlight the flexibility and usability of our method, we opted to work with a set of configurations with significant differences; from over-actuated arms to under-actuated, and all having different joint configurations and link lengths. In a previous stage, the method was already validated on the virtual model of, amongst others, a highly actuated human model with 9 DOF arms, and the robots ASIMO [104] and Justin [2], both having 6 DOF arms, but however with considerably different morphology. For this validation on physical robots we worked with the robots Romeo [132], Pepper [133] and Nao [3]. All three robots have a different morphology. The specifications for Romeo are grouped in table 6.1. The left top shows Romeo's joint configuration. The robot has a 1 DOF actuated body, a 3 DOF head, and an over-actuated right and left arm consisting of 7 DOF. The joints of the arm chain are grouped into the different blocks, which results in a 3 DOF shoulder and wrist, and a 1 DOF elbow block. To calculate the DH-parameters, firstly a DH-frame is assigned to each joint. The

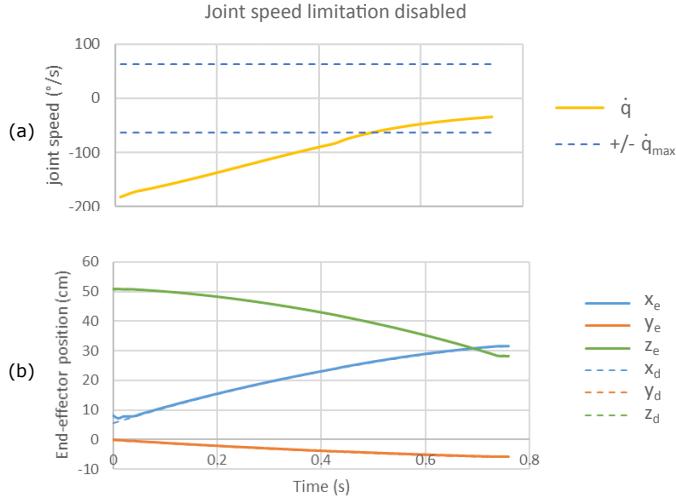


Figure 6.5: Output for a right-handed reaching gesture calculated for the robot Romeo without using the joint speed limitation algorithm. (a) Calculated speed for the first wrist joint, together with its boundaries. (b) End-effector trajectory of the wrist block. The full line denotes the calculated end-effector position  $x_{i,e}$ , while the dotted line shows the desired end-effector position  $x_{i,d}$ .

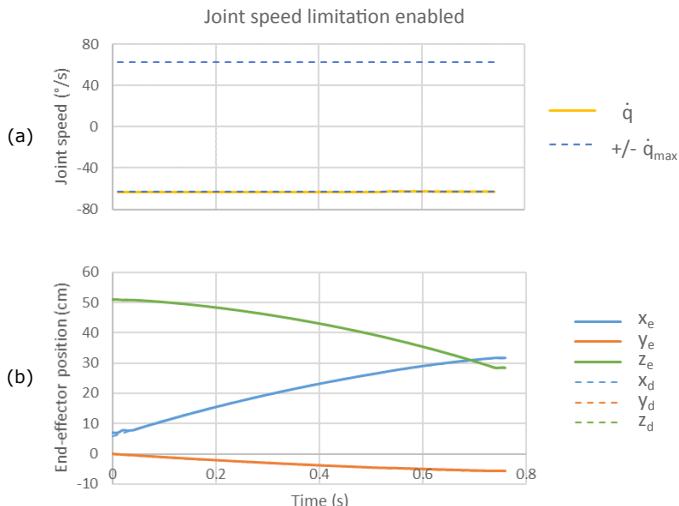


Figure 6.6: Output for a right-handed reaching gesture calculated for the robot Romeo using the joint speed limitation algorithm. (a) Calculated speed for the first wrist joint, together with its boundaries. (b) End-effector trajectory of the wrist block. The full line denotes the calculated end-effector position  $x_{i,e}$ , while the dotted line shows the desired end-effector position  $x_{i,d}$ .

frames corresponding to the first joint of each chain, the chain base frames, are visualized in the middle top of table 6.1, together with the standard reference frame  $x_syz_s$  placed in the pelvis of the robot. The orientation and position of these chain base frames with respect to the standard reference frame are necessary inputs for the program and are specified under the form of homogeneous transformation matrices. The bottom part of table 6.1 lists the remaining specifications that are used as input for the method, namely the DH-parameters, the joint angle limits and the joint speed limits. For the body block, three sets of DH-parameters are specified; the *Body*-set corresponds to the DH-parameters calculated for the body joint with as end-effector frame, the base frame of the head chain. For the *Body left*- and *Body right*-set, the base reference frame of respectively the right and left arm are used. This is used to determine the current orientation of the base reference frames of the head and arm chains in case of body motion. To make the connection between the body joints and the arm base frames in determining the DH-parameters, an extra, non-actuated joint was added. Table 6.2 lists the same specifications for the robot Pepper. This robot consists of a 2 DOF head, a 1 DOF body and a 5 DOF left and right arm. When grouping the joints into the different blocks, this results in a 3 DOF shoulder block, and a 1 DOF elbow and wrist. The specifications for the robot NAO are grouped in table 6.3. Unlike the two previous robots, NAO does not feature an actuated joint in the body. The robot does have a 2 DOF actuated head, and a right and left arm consisting of 5 DOF. Grouping the arm joints in blocks results in a 3 DOF shoulder, and a 1 DOF elbow and wrist block.

The test scenario was designed to group a number of different emotional expressions, calculated by the block mode, and both pointing and reaching gestures, calculated by the end-effector mode. The robot tells a story about how it helped a lost boy in the supermarket finding back his mother. A number of calculated postures for all three robots are listed in figure 6.7. The type of gesture, the used calculation mode (B M: Block mode or EE M: End-effector mode) and the context are added below each posture. Taking in consideration the differences in joint angle range for the different robots, for some gestures, other end-effector positions were chosen to guarantee a successful calculation of the trajectory. The video's of the complete gesture sequence for each robot were grouped on the Probo-website<sup>1</sup>. This validation was performed when the joint speed limitation was not yet implemented in the method. For the robots NAO and Pepper, no speed-related problems occurred. However, since Romeo has more strict speed limits (see table 6.1), a number of calculated gestures violated these limits. The calculated joint trajectories were rescaled in time to be able to be performed by the robot. As a result, the resulting gestures, and therefore the total duration of the test scenario is considerably longer for

---

<sup>1</sup><http://probo.vub.ac.be/GestureMethod/PhysicalRobots.htm>

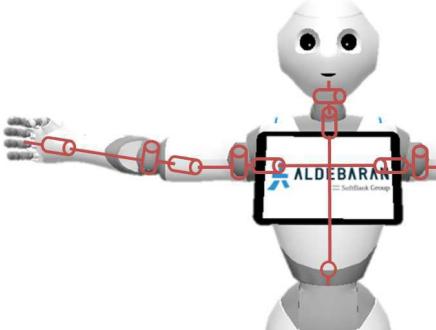
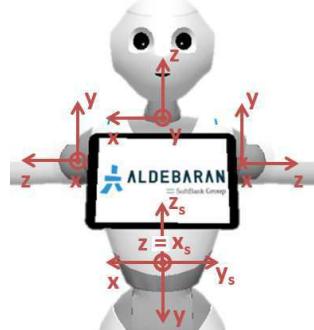
Romeo then for the other two robots. Afterwards, the joint speed limitation was added to the method and the same scenario was ran by the method to re-calculate the joint trajectories for Romeo. Since the physical robot was not any more available, the calculated gestures were visualized on the virtual model of the robot. The corresponding video was added to the same web page as those of the physical robots. Furthermore, a video grouping the four gesture sets was provided to visualize the timing of the calculated gestures for the different cases. When using the joint speed limitation algorithm, for most gestures of the test scenario, an alternative trajectory could be calculated for the Romeo configuration whereby the desired time span was not violated. Only for the first gesture, associated with the text *Hello, I'm Romeo* and calculated using the place-at condition of the end-effector mode, the time span was slightly extended. Figure 6.8 visualizes the timing of this gesture for Romeo, how it was implemented on the real robot and how it is optimally calculated using the joint speed limitation, visualized on the virtual model, together with the desired timing, obtained for the robot NAO. At  $t = 0s$ , the robot stands in a neutral pose. The desired duration of the gesture is  $1.5s$ . As already mentioned above, suitable joint trajectories to fulfil this timing constraint could be calculated for the NAO robot. For the validation on the physical model of Romeo, joint trajectories were calculated without the joint speed limitation. The resulting trajectories were rescaled in time to be able to be performed by the robot, which resulted in a total gesture duration of  $2.67s$ . When using the joint speed limitation algorithm, a duration of only  $1.67s$  was necessary to reach the imposed end-effector position. Since for all other gestures of the scenario, a suitable trajectory could be calculated within the desired time constraints, the overall timing of the resulting test scenario calculated using the joint speed limitation algorithm, is similar to that obtained for the robots NAO and Pepper, while the one used on the physical model of Romeo is unnecessary long.

Table 6.1: Specifications for the robot Romeo.

Joint configuration	DH-base frames		
Body	Head	Right	Left
$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -51 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.91 & 0.42 & 0 & 8 \\ 0.07 & -0.16 & 0.99 & -43 \\ 0.42 & -0.89 & -0.17 & -9.7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.91 & -0.42 & 0 & 8 \\ -0.07 & -0.16 & -0.99 & 43 \\ 0.42 & 0.89 & -0.17 & -9.7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

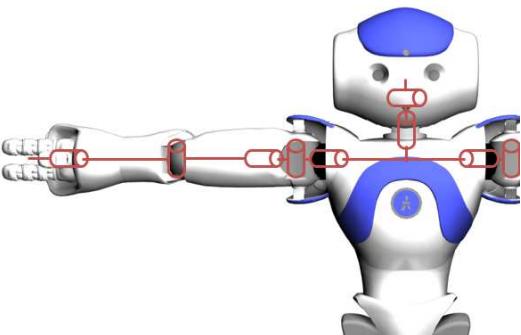
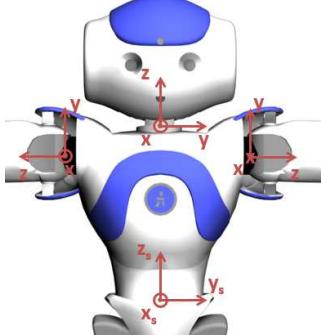
Joint	$\alpha$ (°)	DH-parameters			Joint limits		$\dot{q}_{max}$ (rad/s)
		a (cm)	d (cm)	$\theta$ (°)	min (°)	max (°)	
Head 1	-90	0	9.5	-90	-180	0	4
Head 2	-90	0	0	-90	-110	-50	1.9
Head 3	0	9.3	0	0	-20	20	1.5
Body	0	0	51	0	-45	45	1.5
Body right	90	0	41	-90	-135	-45	1.5
	0	0	19	0			
Body left	90	0	41	90	-135	-45	1.5
	0	0	19	0			
Shoulder 1	-90	0	0	0	-127	80	2.2
Shoulder 2	90	0	0	0	0	95	4
Shoulder 3	-90	0	21.5	0	-120	120	3.7
Elbow	90	0	0	0	0	90	4
Wrist 1	90	0	19	90	-30	210	1.1
Wrist 2	90	0	0	90	65	115	2.6
Wrist 3	0	11	0	0	-55	55	3.8

Table 6.2: Specifications for the robot Pepper.

Joint configuration		DH-base frames	
Body	Head	Right	Left
			
$base\ R_{stand}$			
$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -38.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 7.2 \\ 0 & 0 & 1 & -24 \\ 0 & -1 & 0 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 0 & -7.2 \\ 0 & 0 & 1 & -24.4 \\ 0 & 1 & 0 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Joint	$\alpha$ ( $^{\circ}$ )	DH-parameters			Joint limits		$\dot{q}_{max}$ ( $rad/s$ )
		a (cm)	d (cm)	$\theta$ ( $^{\circ}$ )	min ( $^{\circ}$ )	max ( $^{\circ}$ )	
Head 1	90	0	38.8	90	-29.5	209.5	7.3
Head 2	0	5	9	-90	-130.5	-53.5	9.2
Body	90	24.4	0	0	-29.5	29.5	2.3
Body right	-90	24.4	0	-90	-119.5	-60.5	2.3
	0	-7.2	15	-90			
Body left	-90	24.4	0	-90	-119.5	-60.5	2.3
	180	7.2	-15	90			
Shoulder 1	-90	0	0	0	-119.5	119.5	7.3
Shoulder 2	81	0	0	0.5	0.5	89.5	9.2
Shoulder 3	-90	-1.5	18	0	-119.5	119.5	7.3
Elbow	90	0	0	0.5	0.5	89.5	9.2
Wrist	0	3	22	0	-104.5	104.5	17.4

Table 6.3: Specifications for the robot NAO.

Joint configuration	DH-base frames
	
$base\ R_{stand}$	
Head	Right
Left	
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -21.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & -1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
	$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & 1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Joint	DH-parameters			$\theta$ (°)	Joint limits		$\dot{q}_{max}$ (rad/s)
	$\alpha$ (°)	a (cm)	d (cm)		min (°)	max (°)	
Head 1	90	0	0	0	-119	119	8.2
Head 2	0	5	0	-90	-126	-61	7.1
Shoulder 1	-90	0	0	0	-119	119	8.2
Shoulder 2	90	0	0	15	15	100	7.1
Shoulder 3	-90	0	10	0	-119	119	8.2
Elbow	90	0	0	2	2	88	7.1
Wrist	0	0	17	0	-193	14	19

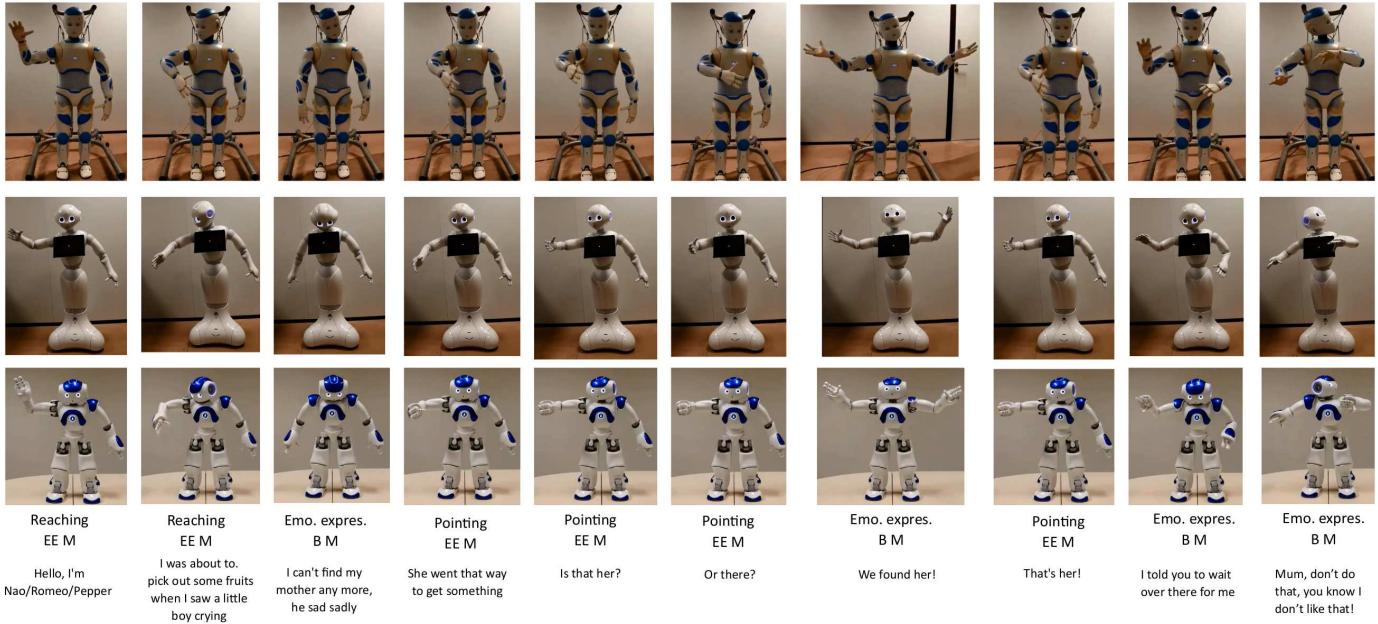


Figure 6.7: Postures captured from the gestures calculated by the method for the robots Romeo, Pepper and NAO. Below every posture, the type of gesture, the used calculation mode (B M: Block mode or EE M: End-effector mode) and the context in the story are added.

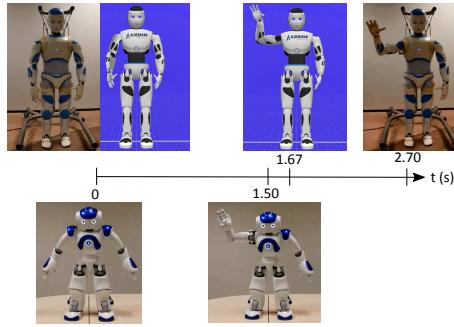


Figure 6.8: Timing of the first gesture of the test scenario for Romeo, how it was implemented on the real robot without using the joint speed limitation and by rescaling the resulting joint trajectories, how it is optimally calculated using the joint speed limitation algorithm, visualized on the virtual model, and the result obtained for the robot NAO, having the desired gesture duration.

### 6.3 Conclusions

To ease the sharing of gestures between different robot morphologies, we proposed the use of a generic gesture method. In the previous chapters, the different modalities of the method were validated on the virtual model of several robots. To guarantee a good performance of the calculated gestures on physical robots, a number of adjustments were made to the method. Firstly, the implemented closed loop inverse kinematics algorithm was extended with a joint angle limitation module. To guarantee the joint angles to stay within their boundaries for every robot configuration, both redundant and non-redundant ones, we opted to work with an algorithm proposed by Drexler and Harmati [131]. Through an illustrative example, the optimal parameters for this algorithm are discussed. Furthermore, a joint speed limitation module was implemented to keep the speeds within their specified limits. To highlight the flexibility and usability of the method, it was used on a set of robots with significant differences in morphology. For the validation on physical robots, gestures were calculated for the robots NAO, Pepper and Romeo. All three robots have different joint configurations, going from over-actuated to under-actuated arms, different link lengths, and differences in joint angle and speed limits. The test scenario was designed to combine different types of gestures, both emotional expressions calculated by the block mode of the method, as well as pointing and reaching gestures, calculated by the end-effector mode. The necessary inputs for the method were discussed and joint trajectories were successfully generated for the three robots.

PART I. DEVELOPMENT OF A GENERIC GESTURE METHOD FOR SOCIAL ROBOTS

# Part II |

Designing social robots using  
a generic gesture method



# 7

# Studying design aspects using a generic gesture method

Since social robots are meant to interact and communicate with humans in a natural way, while operating in our daily environment, their design should be adapted to this. Although many social robots are for that reason more or less based on the human model, the exact morphology of the robot depends on its specific application. In this chapter, we discuss how our developed gesture method, introduced in part I, can be useful in the design of new social robots. The software was designed to cope with the correspondence problem when generating gestures for different robot platforms, resulting in a generic framework that is evaluated at runtime using a small set of morphological information. As such, gestures can be calculated for a desired robot configuration with a minimal effort of the programmer. By generating a set of gestures for different morphologies, the importance of specific joints and their influence on a series of postures can be studied. The gesture method proves its usefulness in the design process of social robots by providing an impression of the necessary amount of complexity needed for a specific task, and giving interesting insights in the required joint angle range.

This chapter is based on the following publication:

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Studying design aspects for social robots using a generic gesture method. *International Journal of Social Robotics, In review.*

## 7.1 Introduction

The design of social robots is a challenging task. In contrast to classical industrial robots, which can be considered more as tools, social robots are aimed at interacting with people in an interpersonal manner [17]. For humans and robots to be able to work closely together, it is important to ensure a natural, intuitive interaction. Therefore, social robots need to be able to communicate using both verbal and nonverbal signs. This, together with the aim of social robots to be used in our daily lives, implying they need to be adapted to our environments and tools, typically results in robot designs that are more or less based on a model of the human body. A number of developed social robots therefore feature 7 DOF arms, consisting of a 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist. This is the case for, amongst others, ASIMO (figure 7.1a) [1], ARMAR-III (figure 7.1b) [134], WABIAN-2 (figure 7.1c) [135] and iCub (figure 7.1d)[103].

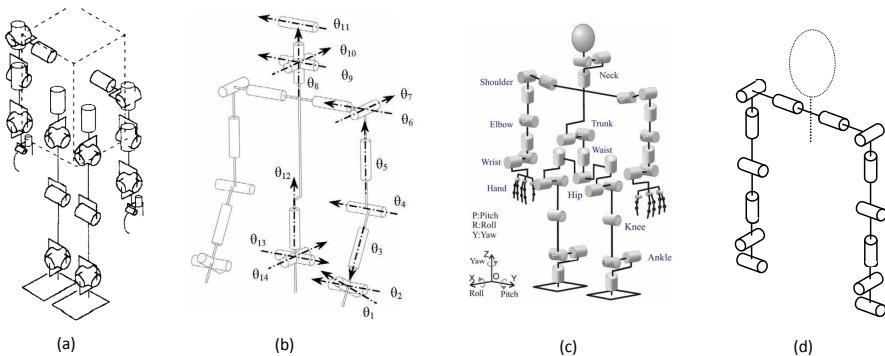


Figure 7.1: Visualization of the joint configuration of a number of robots featuring 7 DOF arms. (a) ASIMO [1]. (b) ARMAR-III [134]. (c) WABIAN-2 [135]. (d) iCub.

When designed for a certain application, some robots may have a different morphology to optimally fulfil the desired tasks. An interesting arm configuration of 7 DOF is that of the robot Pyrène (figure 7.2a) of the TALOS series, aimed for applications in industrial settings. To guarantee a maximum manipulability in the front of the robot, the first shoulder joint is oriented along the yaw axis, instead of the pitch axis like the previously named robots. An additional advantage of this placement is the compactness of the robot when the arms are folded to the front, allowing it to pass narrow sections in its working space [141]. The robot WE-4RII [61], on the other hand, was developed to study human-like emotion. Humans feature a scapula joint, allowing us

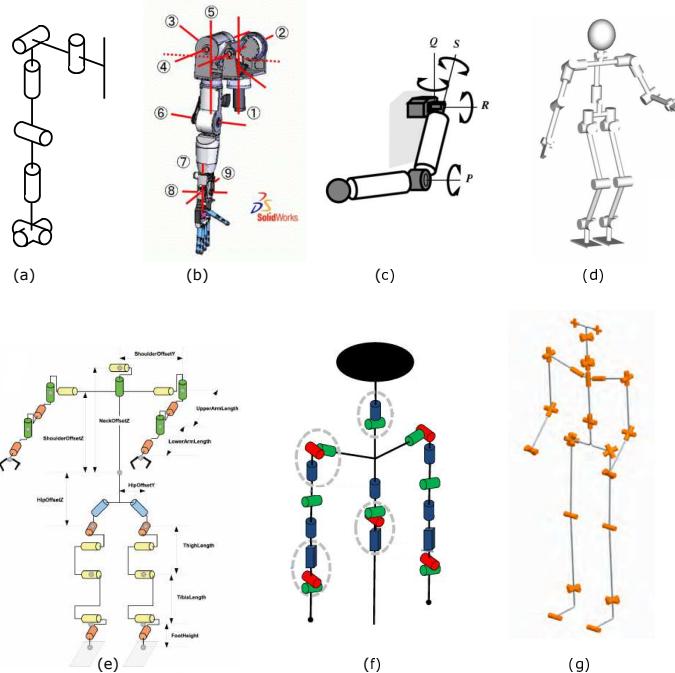


Figure 7.2: Visualization of the joint configuration of a number of robots with different arm morphologies. (a) Pyrène. (b) WE-4RII [136]. (c) Robovie [137]. (d) KHR-3 [138]. (e) NAO [3]. (f) R1 [139]. (g) ARMAR-IV [140].

to shrug or square the shoulders; motions that convey important information concerning our current internal state. Therefore, next to a 3 DOF shoulder part, 1 DOF elbow part and 3 DOF wrist part, the robot was designed with an additional 2 DOF base shoulder part, enabling the robot to create more human-like emotional expressions (figure 7.2b). Robovie's [137] design is more minimalistic. Its applications were mostly focussed on object indication and route direction-giving, and thus, mostly deictic gestures were aimed to be used. The robot features 4 DOF arms, consisting of a 3 DOF shoulder part and a 1 DOF elbow (figure 7.2c). Other robots with less articulated arms are KHR-3 [138] and NAO [3]. Both robots feature a 3 DOF shoulder and 1 DOF elbow. But while NAO's wrist only consists a roll-joint (figure 7.2d), KHR-3's wrist consists of a yaw and pitch joint (figure 7.2e). An important requirement of the recently developed robot R1 was the possibility of manipulating and carrying objects. Given the advantages of a parallel mechanism, namely the high payload and structural stiffness, combined with the lightness of the platform, this solution was used for the wrist actuation. The parallel mechanism consists of a base, three linear actuators and a platform, allowing for the flexion/extension

and adduction/abduction of the wrist joint, as well as a translational motion. For the shoulder, a traditional collocation of 3 serial joints is used. Together with the joints provided for the elbow flexion and internal rotation of the forearm, this results in a robotic arm of 8 DOF (figure 7.2f)[139]. Another example of a robot featuring 8 DOF arms is ARMAR-IV. To achieve more dexterity in bimanual manipulation, the typical 7 DOF arm was extended with the sternoclavicular joint (figure 7.2g) [140].

It is difficult to investigate if the chosen arm morphology is indeed the best possible solution for the robots discussed above. Existing robots differ from each other in a large range of aspects; they have, for example, different joint angle limits, different relative link lengths, a different collocation of joints, and thus, a different overall outer appearance. Because of the wide range of differences, it is difficult to isolate the influence of one specific design parameter. To give insights in the effect of different design aspects on the performance of specified motions and help in making substantiated trade-off's in the design process of new robots, we propose a methodology based on the calculation of gestures for different morphologies and their visualization on one single virtual model. In part I of this thesis, we proposed a solution for the correspondence problem by designing a generic method to generate gestures for social robots. The framework of the software was constructed using a human base model, representing the rotational possibilities of a human. Since most humanoid robots are based on the human body, but in general less actuated, the human base model comprises the majority of available social robots, and thus, the software can be used for this set of robots. At runtime, the generic framework is evaluated using a minimal amount of morphological data, inputted by the user. As such, gestures can be generated for different morphologies with a minimal effort of the programmer, which makes the method interesting as a tool to study the influence of different design aspects of social robots on a set of predefined behaviors.

## 7.2 Methodology

When building a new robot, the designers experience a huge amount of design freedom. The global appearance of the robot will influence the users' expectancies and acceptance towards the robot. The number of joints, their placement and their range have a direct influence of the dexterity and manoeuvrability of the robot. A more articulated robot will have a broader functionality, but will also result in a higher complexity, and, not unimportant, a higher cost. With our proposed design methodology, we provide a tool to give insights in different design aspects, which can help to make substantiated trade-off's in the design process.

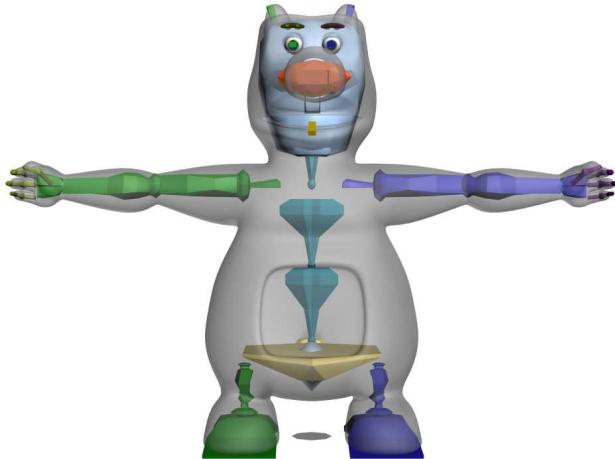


Figure 7.3: Virtual model of the Probo character, constructed using the 3DS MAX Biped.

For social robots designed for a certain application, a set of essential gestures can be selected. By generating this set of gestures for different configurations and visualizing them on a single virtual model, the importance of specific joints and their influence on the performance of the mapped gestures can be investigated. Different collocations of joints can be tested, and also the influence of different joint speed and joint angle limits can be studied. The latter can be interesting for a design using actuators whereof the range is limited. This is, for example, the case for servomotors, but also for a range of series elastic actuators (SEA) and variable stiffness actuators (VSA) [142][143][144]. Using this methodology, an optimal placement of the actuator's neutral point can be determined, which can in some cases cancel the need of an additional gearbox.

In what follows, we will illustrate this idea by using the virtual model of the robot Probo. Probo was designed by the Vrije Universiteit Brussel to study human-robot interaction with children. The first prototype of the robot focused on the use of facial expressions for a natural interaction, resulting in a 19 DOF actuated head [145][59]. Now, in a second iteration, we aim to build a new version of the robot, extended with actuated arms. More information about this can be found in the next chapters (chapter 8 and chapter 9).

The virtual model of the robot was constructed in Autodesk 3DS MAX and rigged as a 3DS MAX Biped (see figure 7.3). The Biped is a standard feature in this software and represents a customizable skeleton consisting of several bones, linked together by following the human anatomy.

Different joint configurations can be assigned to the virtual model, by asso-

ciating the different joints to the available bones of the Biped and specifying the corresponding rotation axes. For every configuration, different gestures can then be calculated as follows:

- **step 1:** calculate the Denavit-Hartenberg parameters of the configuration, specify the joint limits and offer them as input for the method.
- **step 2:** use the method to calculate a set of desired gestures.
- **step 3:** the calculated gestures can be visualized by loading the calculated joint trajectories into the virtual model using 3DS MAX.

Figure 7.4 gives a schematic representation of the workflow of this methodology. In what follows, the three main steps of the process are discussed in more detail.

### Step 1

In the first step, the chosen configuration is quantitatively described by identifying the corresponding Denavit-Hartenberg parameters [146]. Table 7.1 illustrates this for a randomly chosen 5 DOF arm and 2 DOF head configuration. The first column visualizes the chosen joint configuration, superposed on the robot's virtual model. The different arm joints are grouped into a shoulder, elbow and wrist block and DH-axes are assigned to the different joints. The remaining columns list the names of the available joints in the configuration, together with their corresponding DH-parameters. Next to the DH-parameters, a set of joint angles and joint speed limits can be specified. The parameters are stored in text-files and saved in a dedicated directory, from where they are loaded as inputs by the gesture method at runtime.

### Step 2

In the next step, a set of gestures can be calculated. The graphical user interface of the gesture software allows to easily chose between a number of emotional expressions, or to specify a desired position for a pointing or grasping gesture. The method's output is a series of data files containing the calculated joint trajectories.

### Step 3

The last step consists of visualizing the calculated gestures. This can be done in Autodesk 3DS MAX by rotating the Biped's bones according to the calculated joint angle values. A correct mapping between the chosen joint configuration

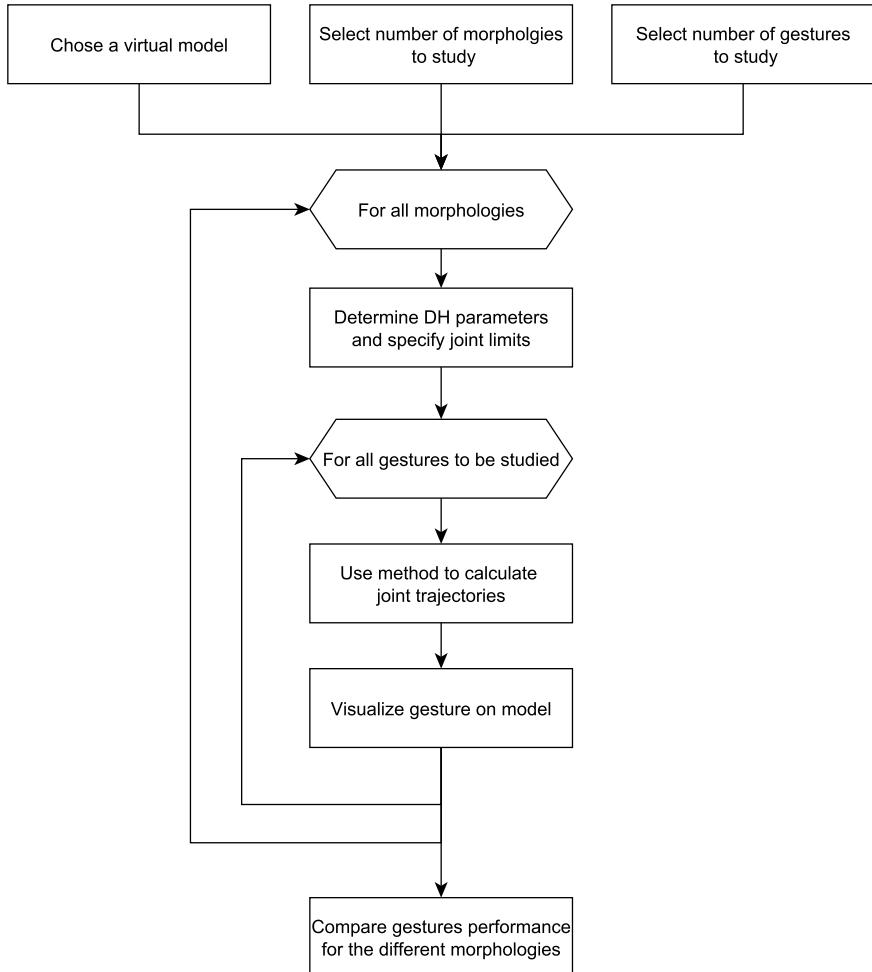


Figure 7.4: Flowchart visualizing the process of the proposed design methodology.

Table 7.1: In a first step, the configuration's Denavit-Hartenberg (DH) parameters are calculated. This tabel shows the DH parameters for a 5 DOF arm and 2 DOF head assigned to the virtual model of the robot Probo. First column: visualisation of the chosen joint configuration. Second column: list of the available joints in the configuration (head joints ranked from bottom tot top, arm joints ranked from right to left). The remaining columns list the corresponding Denavit-Hartenberg parameters.

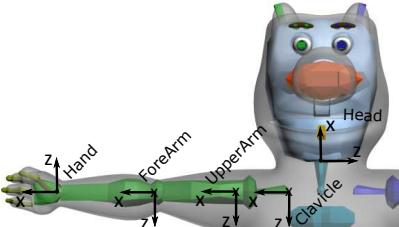
Configuration	Joint	DH-parameters			
		$\alpha$ (°)	a (cm)	d (cm)	$\theta$ (°)
	Head 1	-90	0	0	0
	Head 2	0	18.8	0	-90
	Sh 1	-90	0	0	0
	Sh 2	90	0	0	0
	Sh 3	-90	0	17.7	0
	Elbow	90	0	0	0
	Wrist 1	0	0	24.5	0

and the Biped's specifications provides the necessary rotation axes. The mapping for the joint configuration discussed above is illustrated in table 7.2. The first column shows the virtual model with the Biped bones and their corresponding reference frames. The second and third columns respectively list the different joints of the chosen configuration and the available bones in the 3DS MAX Biped. The last column shows the correct Biped rotation axes that need to be used in combination with the calculated joint trajectories to visualise the desired gestures in 3DS MAX.

### 7.2.1 Effects of joint configuration on a set of emotional expressions

To illustrate the use of the developed gesture method to study the effects of differences in joint configuration, the methodology discussed above was followed for a series of configurations. Table 7.3 shows a set of results for 7 different

Table 7.2: For visualizing the gestures, a correct mapping between the chosen joint configuration and the specifications of the 3DS MAX Biped needs to be calculated, in order to correctly identify the necessary rotation axes. First column: reference frames of the Biped's Bones. Second column: joints of the chosen configuration. Third column: available bones of the 3DS MAX Biped. Fourth column: Biped axes corresponding to the joints' rotation axes.

3DS MAX model	Joint	3DS MAX joint linking Bone      rotation axis
	Head 1	Head      x
	Head 2	Head      z
	Sh 1	UpperArm      x
	Sh 2	UpperArm      -z
	Sh 3	UpperArm      x
	Elbow	ForeArm      -z
	Wrist 1	Hand      x

morphologies, with arm configurations ranging from 9 to 4 DOF. Different emotional expressions were calculated for each of the 7 morphologies, whereof the end postures are shown in table 7.3. From left to right, the end posture, and thus the most expressive posture for *anger*, *disgust*, *happiness* and *sadness* is shown. Configuration 1 consists of a 9 DOF arm, composed of a 2 DOF clavicle, 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist. The head also consists of 3 DOF. For every arm and head block, the maximum amount of joints is present in this configuration. Therefore, it can be seen as a *complete* configuration and thus, the calculated end postures can serve as a reference for comparing the end postures calculated for other configurations. Next to a different collocation of the joints, the main difference of configuration 2 is the missing clavicle block. When observing the end postures, it can be noted that for this virtual model and the chosen gestures, the influence of the clavicle block is negligible. The third and fourth column show two configurations with different 2 DOF wrists. The influence of the specific missing joints can be observed from the small differences in the placement of the hand in the calculated postures. Evidently, these differences become larger when eliminating more joints, as in configuration 5 and 6, where only 1 joint is present in the wrist. Configuration

Table 7.3: End postures for different emotional expressions, calculated for 7 different joint configurations.

Joint configuration	Anger	Disgust	Happiness	Sadness
(1)				
(2)				
(3)				



5 only features the flexion/extension joint in the wrist, while in configuration 6 only the pronation/supination is present. While the end postures calculated for configuration 6 resemble the reference postures well, those for configuration 5 differ significantly, which demonstrates the importance of the pronation/supination motion for this type of gestures. The last column shows an arm configuration with only 4 DOF, consisting of a 2 DOF shoulder, 1 DOF elbow and 1 DOF wrist. By choosing the collocation of the shoulder joints properly, i.e. by providing an internal rotation, followed by a joint causing an adduction/abduction motion, the calculated postures resemble the reference postures relatively well. Only for the emotional expression of *disgust*, the placement of the hand significantly differs from that in the reference posture, however, the intended emotional state can still be recognized. Therefore, this configuration can be interesting for generating emotional expressions when, for example, cost or low complexity is highly important. Of course, for other types of gestures, like accurate pointing gestures or manipulation purposes, this configuration is less interesting because of the reduced dexterity.

Using the gesture method in this way is mainly interesting for giving insights in the influence of joints on different gestures. It gives an impression of the necessary amount of complexity needed for a specified task. For robots intended to perform a variety of different tasks, not specifically described in advance, a more complex robot of course gives more freedom regarding the motions to be performed. But for a robot designed for a specific task or application, such as, for example, a robot companion for children in the hospital based on emotional interaction (like the Huggable [147]), or a route direction-giving robot, a careful consideration of different morphologies can help in finding an optimal design for a desired degree of complexity, cost and expressibility.

### 7.2.2 Effects of joint angle limits on a set of emotional expressions

Another interesting aspect that can be studied is the influence of joint angle limits. In designs using motors with a limited range, an optimal placement of the neutral position can diminish the complexity of the design by reducing the transmission. Table 7.4 and figure 7.5 illustrate the use of the gesture method for this application, for configuration 6 shown in table 7.3, while considering the same gestures as above. The first row of table 7.4, set (a), shows the end postures calculated using a wide joint limit range, namely  $-180^\circ$  to  $180^\circ$  for all joints, except for the elbow joint, which only goes to  $0^\circ$ . These joint limits are represented by dots on the graphs shown in figure 7.5a. Here, the first column represents the right arm, while the second represents the left. The calculated joint angles, necessary to reach the desired end posture for the gestures are visualized in the same graph. From these results, a first constriction of the

joint angle range can be tested. For the joints in the right arm, the boundaries are set such that they include the values calculated in the first trial (a), except for the first shoulder joint. Here, the range is limited from  $-100^\circ$  to  $20^\circ$ , constraining the initial value of  $30^\circ$  for the expression of *anger* and  $-133^\circ$  for that of *disgust*. The new limits, together with the corresponding calculated joint angles are visualized in the second row of figure 7.5. The effect of the new constraints can be noted from the difference in placement of the forearm in the second row of table 7.4. For the left arm, a similar joint range was chosen. For the 3 DOF shoulder block, this resulted in a different joint angle set to reach a quasi-identical arm placement. The calculation of the angles for the elbow and wrist joint was not influenced by the restriction in angle limits.

A second trade-off that can be made is a similar constriction of the third shoulder joint. By setting the limits from  $0^\circ$  to  $120^\circ$ , instead of to  $150^\circ$  as in trial (b), the calculated values will fall in the range of most servomotors, which can be a practical advantage. The results for these limits are shown in the third column of table 7.4, and the corresponding calculated angles are visualized in figure 7.5c. This second constriction only has an effect on the right arm, since for the left arm, all calculated angles for trial (b) already fell in this range. For the right arm, again a difference in placement can be noted for the expressions of *anger* and *disgust*. While a similar restriction can highly simplify the design, the effect on the predetermined set of gestures is minimal. It can therefore be interesting to perform similar studies in the design process, in order to predict and anticipate on the expected joint angle range.

## 7.3 Conclusion

In this chapter, we proposed the use of the developed gesture method, introduced in part I, as a tool in the design process of social robots. The framework of the software is constructed very generic, and is only evaluated at runtime using a minimal set of morphological information specified by the user. Therefore, gestures can be created for numerous configurations with a minimal effort, which makes the method interesting for design purposes as well. For robots designed to achieve predefined tasks, a number of essential gestures can be nominated in advance. Generating these gestures for different morphologies can give interesting insights in the necessary design complexity for the desired task, which can be useful to make a correct trade off in cost, simplicity and performance. An example of the influence of both the collocation of the different joints, and the joint angle range was illustrated using the model of the robot Probo. Since also the joint speed limits are a necessary input for the gesture method, also this influence can be studied. As such, the gesture method can be a practical tool in the design process of social robots and help in generating an optimal design regarding cost, complexity and expressibility.

Table 7.4: To study the effects of the joint angle range, emotional expressions for *anger*, *disgust*, *happiness* and *sadness* were created for configuration 6 visualized in table 7.3, using different sets of joint angle limits.

Set	Anger	Disgust	Happiness	Sadness
(a)				
(b)				
(c)				

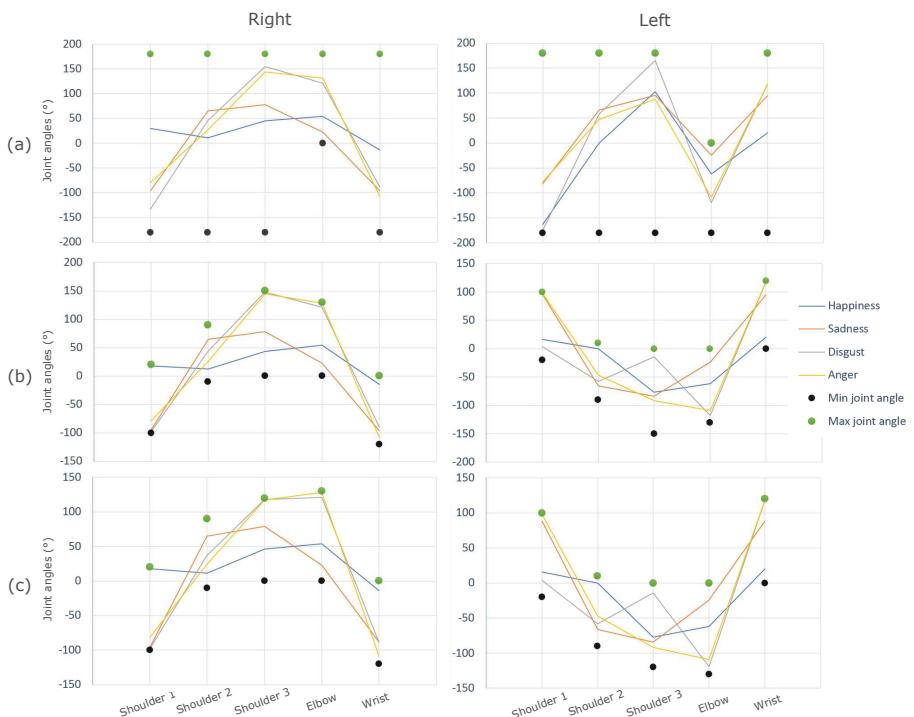


Figure 7.5: Calculated joint angles for the expression of *anger*, *disgust*, *happiness* and *sadness* for configuration 6 visualized in table 7.3, for different sets of joint angle limits.

PART II. DESIGNING SOCIAL ROBOTS USING A GENERIC GESTURE METHOD

# Part III |

## Elvis, a new social robot



# 8

# Design of Elvis; a new social robot

This chapter discusses the design of Elvis, a new social robot that was developed in the frame of the Probo-project. This project aims to study cognitive human-robot interaction and the possibilities of robot assisted therapy (RAT), with a special focus on children. The initial version of the robot consists of an actuated head with 19 DOF, and uses facial expression in combination with speech to establish an intuitive communication with the user. Probo was used as facilitator in several robot assisted therapies for children diagnosed with autism spectrum disorder (ASD). The new prototype of the robot, named Elvis, is designed based on the experiences and conclusions drawn from these experiments. The main novelty in this development stage is the implementation of actuated arms. To achieve an optimal morphology for the targeted applications of the probo-project, an a priori study using the developed generic gesture method was performed on both the joint configuration and the joint range. This resulted in three interesting configurations that were investigated in more detail and finally, physically developed.

This chapter is based on the following publications:

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Development of Elvis, a new social robot for affective and functional gesturing. *IEEE Robotics and Automation letters, In review.*

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing. *International Journal of Robotics Research, In review.*

## 8.1 The Probo-project

Probo is a huggable social robot, developed to study cognitive human-robot interaction with children. The project initially focused on hospitalized children; since a stay in the hospital can be a frightening experience for children, the robot Probo would be their robotic friend, and could comfort and inform them about their stay in the hospital, possible interventions and the use of medication.

The first prototype of the robot was released in 2009. To avoid users from having specific expectations towards the behavior of the robot, Probo was designed as an imaginary animal, based on the ancient mammoths. Because of the trunk, it is very recognizable and distinct from other social robots. Human social cues and communication skills are used to guarantee an intuitive communication and natural interaction between a child and the robot, whereby for the first prototype, the focus was laid on the use of facial expressions as non-verbal communication modality. The robot has an actuated head consisting of 19 DOF, including movable eyes, eyelids, eyebrows and ears and a movable neck, trunk and mouth. Since the robot was designed to work in close interaction with children, special attention was paid to the huggable and safe nature of the robot. Compliant actuators are used for all driven joints, in combination with the use of soft and flexible materials [148][145]. The complete mechanical system is covered with a fabric jacket to provide a soft and huggable appearance. Figure 8.1a shows the uncovered head of the robot, while figure 8.1b shows the covered prototype in interaction with a child.

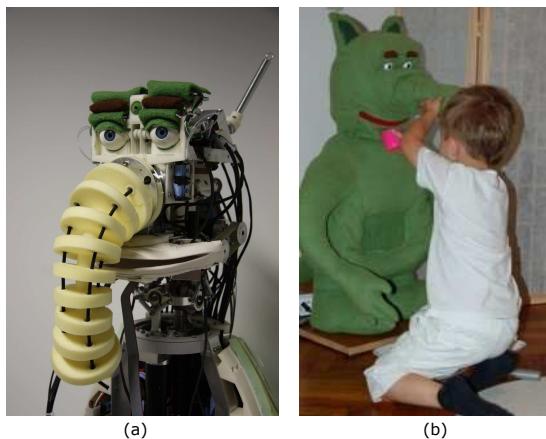


Figure 8.1: The social robot Probo. (a) Uncovered prototype. (b) Probo in interaction with a child.

To control the robot, a software architecture consisting of several subsystems, including a perceptual, emotional and motor system, was developed. The generated facial expressions were validated through several recognition tests [59]. Figure 8.2 visualizes the facial expressions for the basic emotions *happiness*, *surprise*, *sadness*, *anger*, *fear* and *disgust*.

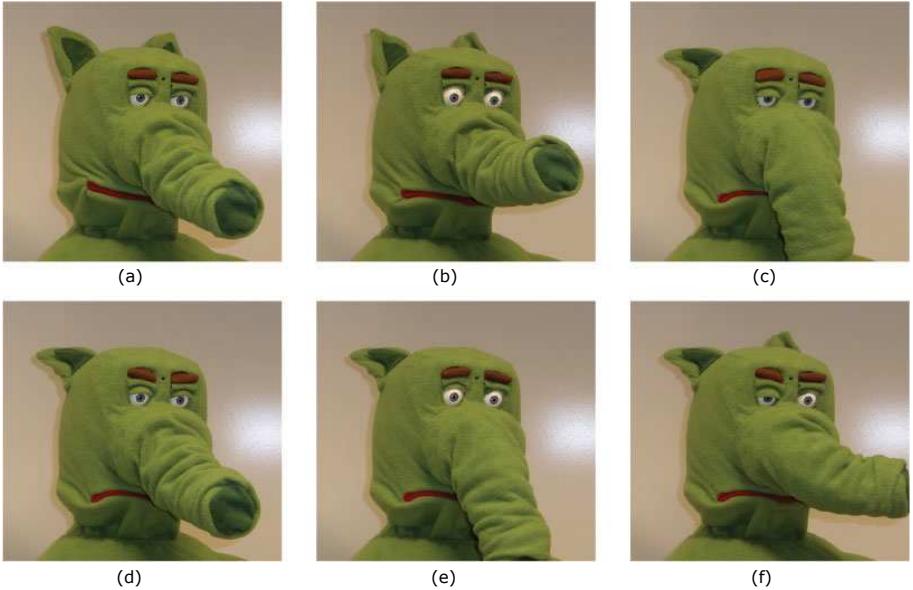


Figure 8.2: The 6 basic emotions, expressed by the robot Probo. (a) Happiness. (b) Surprise. (c) Sadness. (d) Anger. (e) Fear. (f) Disgust.

After the development, the research team had the opportunity to evaluate the robot's use in RAT for children with autism spectrum disorder, which resulted in a switch from hospitalized children as main target group towards this population. In a first study, the effectiveness of employing Probo as facilitator in an existing intervention, called Social Story telling, was investigated [149][150]. In addition, the team investigated whether the robot can be useful in teaching children with ASD to identify situation-based emotions [151]. In a later stage, two interaction studies investigated if children with ASD behave differently when interacting with a robot or human during a play task, by analysing the performance of the task, the frequency of certain social and asocial behaviors and the children's ability to elicit interaction with an accompanying adult and collaborative behavior [29][30].

By performing these experiments and observing the interactions of children with Probo, several aspects that could improve the robot and its applicability

for RAT could be noted;

- The robot's big trunk covers a large portion of the mouth. Since children with ASD mostly focus on the mouth to read facial expressions, this was sometimes perceived as disturbing. Therefore, a smaller trunk would benefit the recognizability of the displayed emotions.
- When the touch screen in the belly was used, the child's attention was completely focused towards it, while the robot's face was practically ignored. This entails that the robot is not any more considered as an embodied agent, but rather as a standard screen.
- The variety of therapies that could be investigated would significantly increase if the robot would be equipped with actuated arms. Examples are interventions stimulating joint attention and imitation games. Of course, the use of gestures would increase the aliveness of the robot in a large extent, and benefit the interaction in all respects.
- The fact that it is positioned upright can raise the expectation that the robot can walk. Therefore, a possible improvement can be to make the robot sit.

## 8.2 Elvis

Based on these considerations, a new version of the robot was developed. Since the outer appearance of the robot significantly changed from the initial Probo character, we decided to change the name as well and called the new developed robot *Elvis*. Regarding its identity, Elvis can be seen as Probo's little brother. In this development stage, we mainly focused on the design of the actuated arm system. The head of the robot only contains a neck module of 3 DOF, designed by colleague Albert De Beir. In a later stage, the arm system will be combined with a complete re-design of the head, resulting in a fully actuated system.

## 8.3 Joint configurations

To select a suitable joint configuration for the arms to be designed, the developed gesture method, discussed in part I, was used to study the effect of several morphologies on a set of gestures. Since the robot is aimed to interact with children on an emotional level, the ability of expressing affective states is a first important constraint. Therefore, the gesture study was performed by evaluating the performance of a set of emotional expressions for 7 different arm

configurations. The end postures of the calculated gestures were discussed in the previous chapter (chapter 7) and listed in table 7.3. To ease the discussion below, the list was repeated as table 8.1 in this chapter. Three joint configurations appeared to be interesting to consider in more detail. Configuration 6 proved to have a good performance of the calculated gestures, for a relatively low complex arm chain of 5 DOF. Configuration 5 only differs from configuration 6 by the replacement of the joint responsible for pronation/supination of the forearm by one generating a flexion/extension motion. This configuration appeared to be less interesting to generate the emotional expressions of *anger*, *disgust* and *sadness*. However, for other types of gestures, such as a number of emblems, the flexion and extension of the wrist is important to realize the gesture. Examples are the hand movement for *stop*, or clapping the hands. By choosing the initial placement of the wrist differently, it can become an interesting morphology for generating emotional expressions too. Finally, configuration 7 also gave a reasonable performance versus complexity ratio. For a configuration of only 4 DOF, consisting of a 2 DOF shoulder, 1 DOF elbow and 1 DOF wrist, recognizable gestures could be generated.

To be able to study all three joint configuration physically, the arm system was designed semi-modular, allowing several joint modules to be switched to result in the different morphologies. The realisation of the three joint configurations are denoted, respectively, as Elvis-Ca, Elvis-Cb and Elvis-Cc.

### 8.3.1 Elvis-Ca

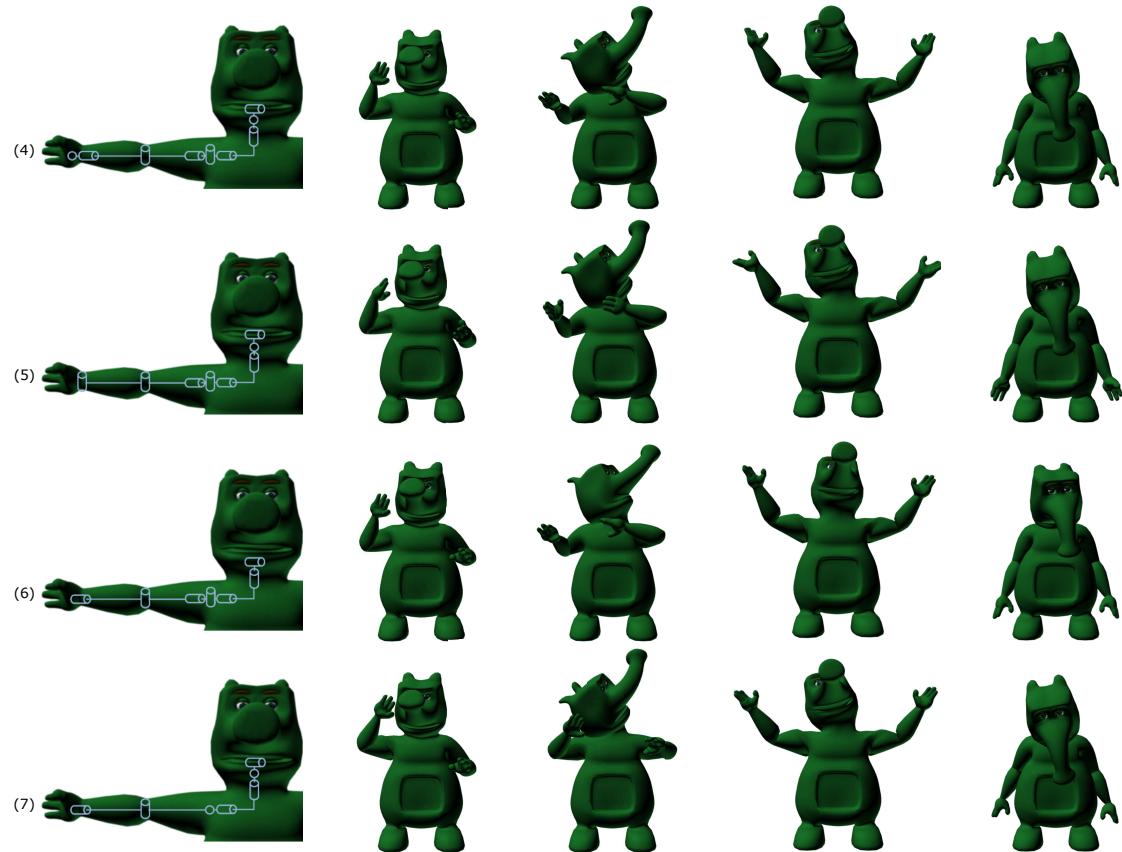
Figure 8.3a shows the joint configuration of the arm used for Elvis-Ca, superposed on the virtual model of the original version of Probo. The configuration consists of a 3 DOF shoulder block, 1 DOF elbow and 1 DOF wrist, containing the joint responsible for the pronation/supination of the forearm. This specific collocation of joints equals that of the robot NAO. As for most available servos, the range of the used motors is limited to 120°. By studying the joint trajectories of the generated gestures, an optimal placement of the neutral position of the motor can be achieved. In the previous chapter, more specific in section 7.2.2, the developed gesture method was used to study the influence of different sets of joint angle limits on the calculated end posture for a series of emotional expressions. For every joint, it was possible to select an optimal neutral position of the motors, resulting in a good performance of the calculated gestures when only using a joint range of 120°. As such, no additional transmission is needed to increase the joint range.

### 8.3.2 Elvis-Cb

Elvis-Cb is based on joint configuration 5 of the gesture study. The resulting

Table 8.1: End postures for different emotional expressions, calculated for 7 different joint configurations. (Iterated from table 7.3.)

Joint configuration	Anger	Disgust	Happiness	Sadness
(1)				
(2)				
(3)				



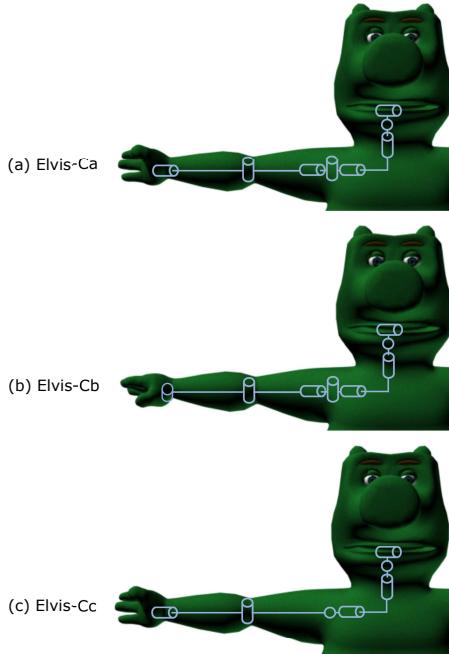


Figure 8.3: Joint configuration of the arm used for (a) Elvis-Ca, (b) Elvis-Cb, (c) Elvis-Cc.

end postures of the calculated emotional expressions, listed in table 8.1, suggest that the missing joint responsible for the pronation and supination of the forearm, is important for a good performance of the gestures. For a number of other gestures, however, the flexion and extension of the wrist is a necessary motion to naturally execute the arm movements. Therefore, it is worthwhile to take a deeper look into this joint configuration. For the results visualized in table 8.1, the joint configuration was linked to the virtual model when standing in T-pose, the hand palm facing out. By altering the hand orientation of the robot's model after coupling it to the configuration, a different robot appearance, and therefore, different end postures can be reached. Table 8.2 visualizes this for 5 different angles. In the first column, the model is placed in T-pose, while the other columns visualize the calculated end postures for, again, the emotional expression of *anger*, *disgust*, *happiness* and *sadness*. The first row shows the morphology used in the previous chapter; the model is placed in T-pose with the palm facing out. In the second column, a pronation of  $30^\circ$  is imposed on the right and left forearm of the virtual model after coupling it to the joint configuration. Therefore, there is a constant deviation of the hands' orientation with respect to the first row. The remaining rows similarly feature morphologies whereby the hand is rotated around the forearm's midline, with

Table 8.2: By altering the hand orientation of the robot's model after coupling it to the configuration of Elvis-Cb, a different robot appearance, and therefore, different end postures can be reached.

Hand ori	Anger	Disgust	Happiness	Sadness
0°				
30°				
45°				
60°				
90°				

an angle of, respectively,  $45^\circ$ ,  $60^\circ$  and  $90^\circ$ . The calculated end postures show that a pronation of  $60^\circ$  with respect to the original T-pose is a good option for the chosen emotional expressions. In addition, positioning the hand according to this angle allows a good performance of a number of emblem gestures, such as the *stop*-gesture, and clapping the hands (figure 8.4). Figure 8.3b visualizes this specific configuration, used for Elvis-Cb, superposed on the virtual model of Probo.

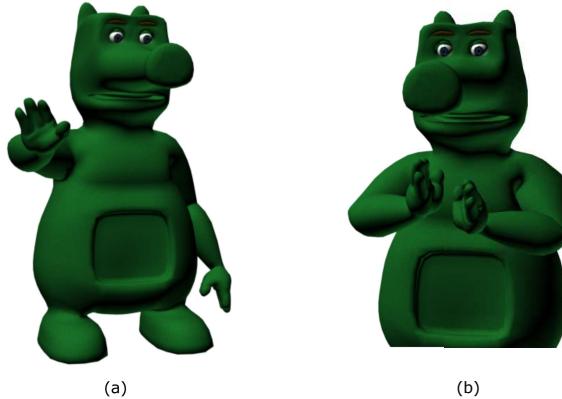


Figure 8.4: Positioning the hand according to a pronation of  $60^\circ$  with respect to the original T-pose after coupling the configuration Elvis-Cb to the virtual model, allows a good performance of a number of emblem gestures, such as (a) the *stop*-gesture, and (b) clapping the hands.

Since in the block mode of the gesture method, the joint angles are calculated separately for each block, and this joint configuration only differs from configuration Elvis-Ca by a different wrist, the necessary joint angle range to calculate the aforementioned emotional expressions corresponds to the one designated to the previous configuration, except for the wrist joint. Figure 8.5 visualizes the necessary joint angles to reach the end postures of the calculated emotional expressions. In section 7.2.2, a suitable neutral position for the shoulder and elbow servos could be found, ensuring a proper performance of the calculated gestures for a joint angle range of  $120^\circ$ . Observing figure 8.5 learns that for the wrist joint, for both the left and right arm, the calculated joint angles lie in a range of  $120^\circ$ . By choosing the maximum position of the right wrist's servo at  $35^\circ$  with respect to the T-pose, and the minimum at  $-85^\circ$ , the gestures can be calculated without approximations or the need of additional transmission ratios.

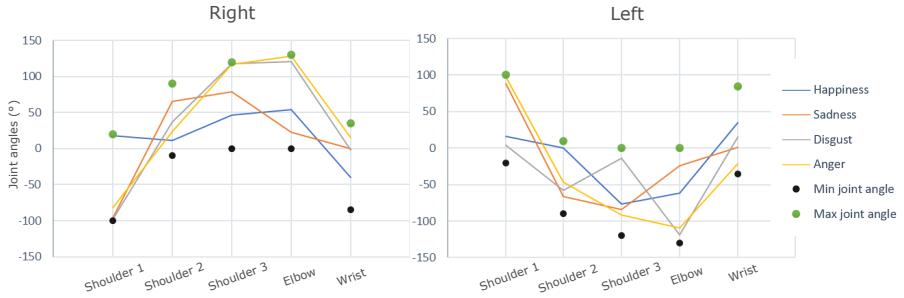


Figure 8.5: Calculated joint angles for the expression of *anger*, *disgust*, *happiness* and *fear*, together with the chosen joint angle limits, indicating the maximum and minimum reachable position of the servos.

### 8.3.3 Elvis-Cc

The last morphology, Elvis-Cc is based on configuration 7 of Table 8.1 and features an arm configuration consisting of only 4 DOF (figure 8.3c). In the state of the art, social robots predominantly feature shoulder modules with 3 DOF, which makes this specific configuration with a 2 DOF shoulder a unique test case. The gesture study performed in the previous chapter resulted in a reasonable performance of the postures, and therefore, this morphology can be interesting to realize in physical form.

## 8.4 Design principles

In contrast to the original version of Probo, which uses relatively expensive Maxon motors and custom made aluminium parts, Elvis was designed using low cost techniques. Firstly, for the actuation, only widely available servo motors were used. To achieve a compliant behavior, the use of servo savers was investigated. Servo savers are used in radio controlled (RC) cars to protect the motors in case of a collision. They are small elements that are placed between the driving servo and the output link and contain a spring intended to absorb a possible impact. Secondly, the arm system is, except for the bearings, completely 3D printed in Polyamide (PA). Next to the cost-related benefits of this technique, compared to the milling and turning of metal parts, the use of a plastic as main working material contributes to a lightweight design. Additionally, to keep the structure light, the shaping of the arm is achieved by the structural parts itself, instead of providing a structural framework that is covered by shields to shape the final look.

In what follows, the different joint modules are discussed in detail.

## 8.5 Design of the different modules

### 8.5.1 Shoulder

The complete shoulder, shown in figure 8.6, consists of three joints, whereof the first and third are twisting joints. The arm is connected to an aluminium frame by a casing that holds the bearings for the first twisting joint. The joint is actuated by a servo attached at the bottom of the casing, by means of a belt (figure 8.7a). The servo for the second joint, responsible for the flexion/extension and abduction/adduction motion, is located inside the casing, and positioned between the two bearings supporting the twisting motion. Again a belt is used to connect the servo and the output link (figure 8.7b). While the mechanics of the first two joints are merged to a certain extent, the last joint of the shoulder is realized as a separate module. As such, if desired, it can be replaced by a rigid structure to realize the configuration of Elvis-Cc. The module houses the servo, that directly drives the internal rotation of the upper arm, together with the bearing supporting this motion (figure 8.7c). The specifications of the used servos, together with those of the other modules are listed in table 8.3.

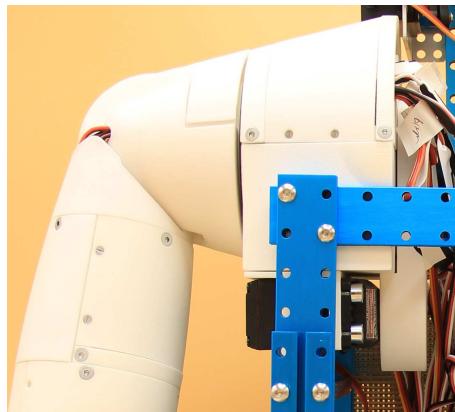


Figure 8.6: Complete shoulder of Elvis, consisting of three joints.

### 8.5.2 Elbow

The elbow module connects the output of the third shoulder joint to the robot's forearm. As for all other non-twisting joints, the output link is actuated by the servo using a belt. Figure 8.8a shows how the servo and the drive shaft are positioned in the design, while figure 8.8b shows the shielded module.

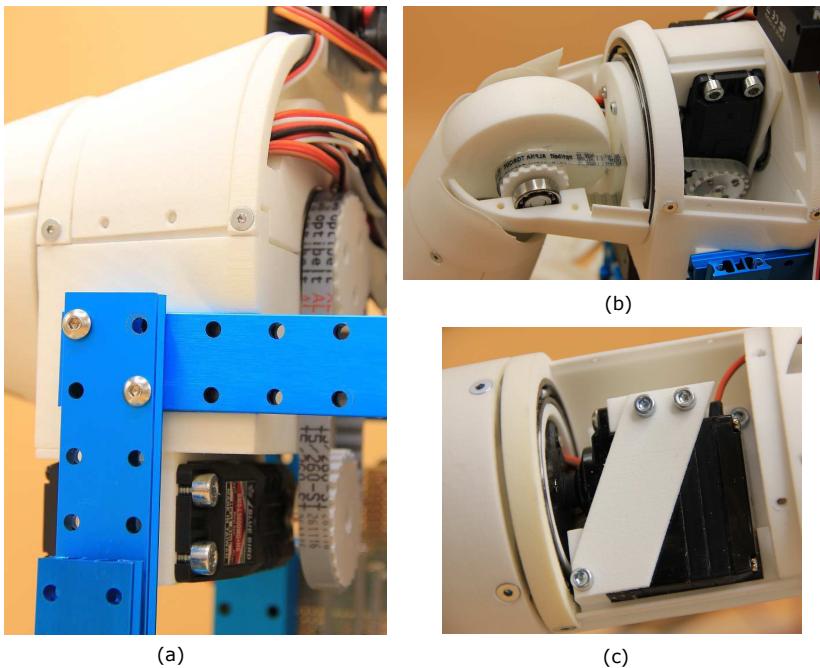


Figure 8.7: Shoulder design of Elvis; internal mechanics. (a) Shoulder 1: twisting joint. (b) Shoulder 2: flexion/extension and abduction/adduction motion. (c) Shoulder 3: internal rotation of upper arm.

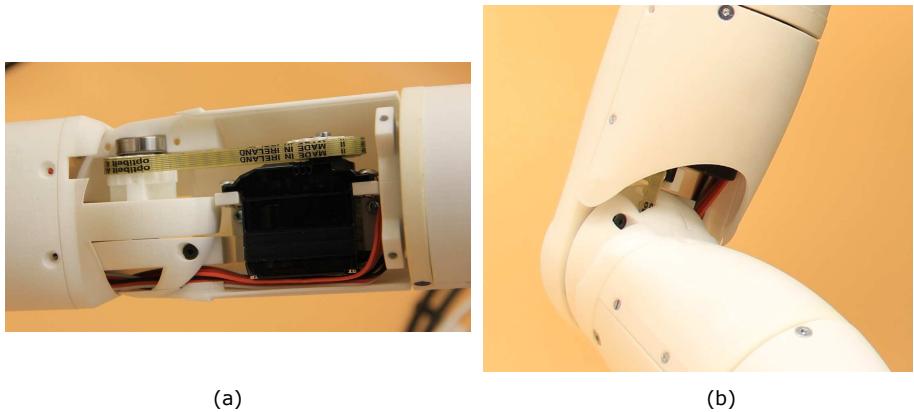


Figure 8.8: Elbow design of Elvis. (a) Internal mechanics, (b) Shielded.

Table 8.3: Specifications of the servos used in Elvis.

Joint module	joint	Servo name	Nom. torque Ncm	Nom. speed $s/60^\circ$
Shoulder	1	BMS-L530MG	243	0.12
	2	BMS-L530MG	243	0.12
	3	BMS-630MG	146	0.15
Elbow	BMS-630MG		146	0.15
Wrist	BAU 13	BMS-630MG	146	0.15
	BAU 14	BMS-380MAX	46	0.14
Fingers	BMS-380MAX		46	0.14

### 8.5.3 Wrist

As mentioned before, two different wrist modules were developed for Elvis. Elvis-Cb features the joint corresponding to BAU 14; the flexion and extension of the wrist. A drive shaft joins the hand module with the structure of the forearm, and is connected to the driving servo, situated in the lower part of the forearm, by means of a strong wire. The servos providing the motion of the fingers (see section 8.5.4) are located in the top part of the forearm (figure 8.9a).

The joint corresponding to BAU 13, the pronation and supination of the forearm is used in Elvis-Ca and Elvis-Cc. This module is interchangeable with the previous one, and uses a rectangular shaped rod to rigidly connect both parts instead of a drive shaft. The joint, situated in the middle of the forearm, is directly driven by the servo motor. The lower part is in this case used to host the finger-servos. Figure 8.9b shows the printed forearm featuring this joint.

### 8.5.4 Hand module

The design of the hand is based on the appearance as it is commonly used in cartoons; consisting of a thumb and 3 fingers. To allow the robot to accurately point towards a certain direction, the index finger is actuated separately. This is especially useful in, amongst others, therapies involving joint attention. In contrast, NAO's fingers are actuated as a whole, which makes it more difficult

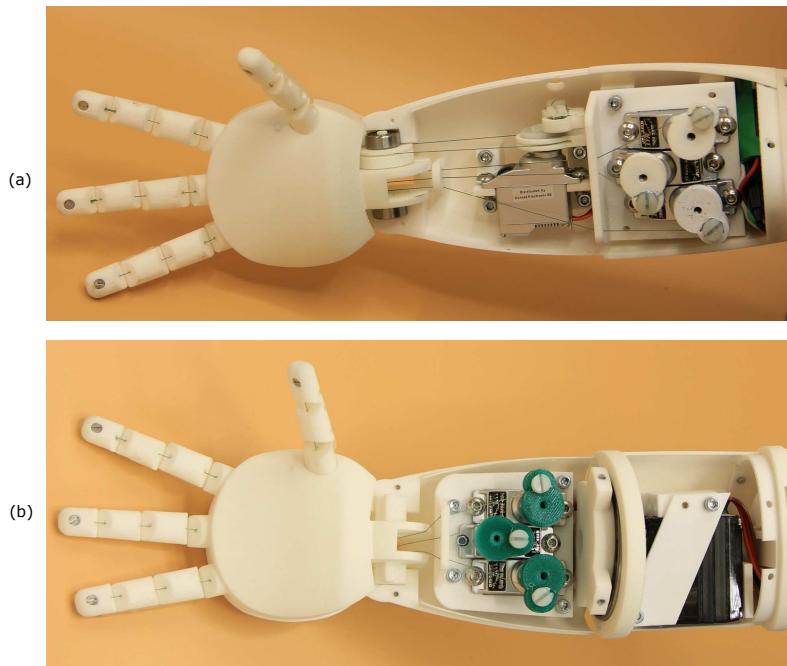


Figure 8.9: Forearm design of Elvis. (a) Forearm containing BAU 14, used for Elvis-Cb. (b) Forearm containing BAU 13, used for Elvis-Ca.

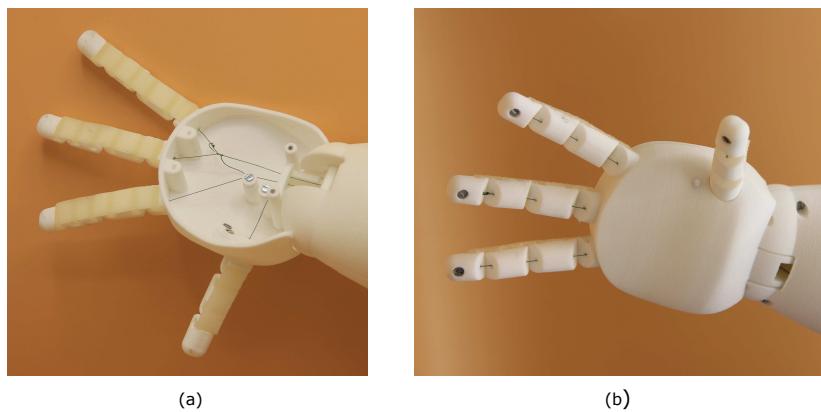


Figure 8.10: Hand design of Elvis. (a) Back of the hand, unshielded. (b) Hand palm.

to accurately point to a certain position. The thumb is actuated separately as well, which makes it possible to use, for example, the emblem *thumbs up*. The remaining two fingers are actuated together. The fingers are tendon driven; thin wires, originating from the fingertips are guided through the wrist, allowing to position the servos in the forearm. Figure 8.10 shows the hand module of Elvis.

## 8.6 Elvis's new look

The two arms, consisting of the different modules discussed above, were 3D printed and mounted on an aluminium frame, together with the neck module provided by colleague Albert De Beir. To obtain a characteristic huggable Probo-look, a special jacket from fabric was designed to cover the complete mechanical system. A soft, brown faux-fur was chosen, in combination with a blue fleece fabric for details in the ears, paws and trunk. This color combination contributes to the imaginary aspect of the character. As suggested by the experiences with the original version of Probo, Elvis was put in a sitting position, the touch screen was omitted and the trunk was shortened. To give the robot a more gentle and cute appearance, the head and abdomen were rounded, resulting in a chubby creature. Additionally, the shape and size of the ears were altered to the resemblance of elephant ears. The result is shown in figure 8.11.

## 8.7 Conclusions

In this chapter, the development of the social robot Elvis was discussed. The robot was designed within the frame of the Probo-project, which aims to study human-robot interaction with children and the possibilities of a natural communication using human verbal and non-verbal communication skills. Elvis was designed, based on the findings and conclusions drawn from several experiments with the robot Probo. Where the initial version of Probo focused on the use of facial expressions as non-verbal communication modality, Elvis was equipped with actuated arms to enable the use of gestures and body language. To guarantee an optimal morphology for the aimed purposes of the robot, the developed gesture software was used to perform an a-priori joint study. Because the expression of affective states is of major importance in this project, a selection of possible configurations was made by studying the performance of a set of emotional expressions for different morphologies using the developed gesture method. Three different configurations were selected to be studied in more detail, and later physically realised. The robot is designed semi-modular, allowing certain modules to be switched in order to obtain the three different morphologies. The design of the different modules was discussed, whereafter

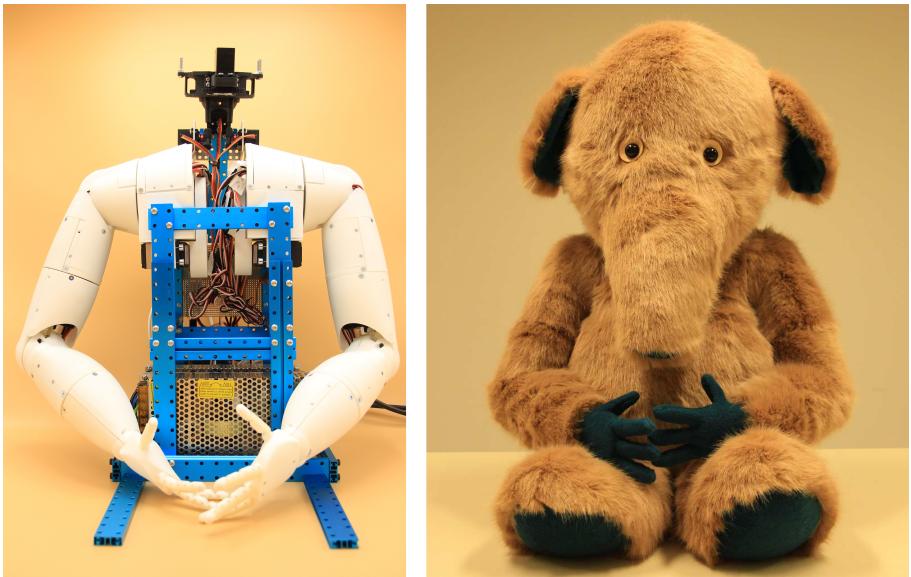


Figure 8.11: The social robot Elvis. Left: non-covered prototype, featuring configuration Elvis-Ca. Right: covered prototype.

the complete assembled prototype was presented. The next chapter focusses on generating gestures for the robot Elvis.

PART III. ELVIS, A NEW SOCIAL ROBOT

# 9

# Generating motions for Elvis

The previous chapter described how three different configurations where selected and physically realised for Elvis. Using the developed software discussed in part I of this thesis, gestures were generated for a series of morphologies, whereby the effect of different joint collocations was investigated. To obtain an optimal design, the joint range and the exact positioning of the servomotors was studied as well. By designing the arm system semi-modular, specific joint modules can be switched to result in the three different joint configurations. As a final step, this chapter presents a number of gestures generated for the three Elvis variants. The emotional expressions, used in the aforementioned gesture study, as well as a set of additional gestures, were generated by both the uncovered and the covered robot, and discussed in detail.

This chapter is based on the following publications:

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Development of Elvis, a new social robot for affective and functional gesturing. *IEEE Robotics and Automation letters, In review*.

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing. *International Journal of Robotics Research, In review*.

## 9.1 Controlling the robot

The 19 servomotors located in the arms and head are driven by an Arduino and two Adafruit 16-Channel 12-bit PWM/Servo Driver boards. The necessary software to control the robot's motion was provided by colleague Hoang-Long Cao and is schematically represented in figure 9.1. The system uses Naoqi, developed by Softbank Robotics, as a middleware to enable software to communicate with a virtual robot model, regarding motion, speech and sensors. The behavior of the virtual model is mapped with that of the physical robot through a bridge using a serial communication with the Arduino. As such, the robot can be controlled using software developed in Choreographe (Softbank Robotics) or other SDKs. For all three Elvis configurations, a set of gestures was calculated using the developed gesture method. The resulting joint trajectories were loaded in Choreographe, from where they, through the middleware, could be sent to the robot as animations. Additional gestures could be created using Choreographe's timeline, by manually putting the different joints of the virtual model in the desired position.

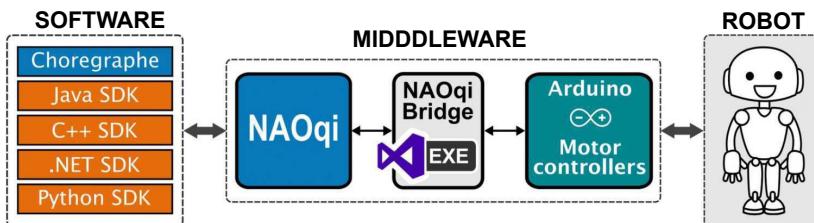


Figure 9.1: Schematic representation of the different software units used to control the robot.

## 9.2 Results

### 9.2.1 Uncovered model

Table 9.1 lists the end posture for a number of emotional expressions, generated by the uncovered model of the three different robot configurations Elvis-Ca, Elvis-Cb and Elvis-Cc. The video's of the gestures were grouped on the Probo-website<sup>1</sup>. For this initial validation, the elastic elements connected to the servos, the servo savers, were removed to eliminate any possible influences of the gravity and inertia on the performance of the gestures. The results are in line with the expectations raised by the gesture tests using the virtual model

<sup>1</sup><http://probo.vub.ac.be/GestureMethod/Elvis.htm>

of the initial Probo, discussed in the previous chapter (see table 8.1 and 8.2). Because Elvis-Cb misses the joint responsible for the internal rotation of the forearm, a difference in arm placement can be noted for this configuration. In the previous chapter, an optimal value for the constant hand deviation with respect to the original T-pose was determined, resulting in a pronation of 60° as optimal results. As predicted by the tests with the virtual model, the results in table 9.1 confirm that for the neutral pose and the emotional expression for *sadness*, only a slightly difference in hand placement can be noted, while the effect is the largest for the performance of *happiness*. Regarding Elvis-Cc, the effect of the 2 DOF shoulder module mostly manifests itself in the emotional expression of *disgust* and *anger*.

A number of other gestures are listed in table 9.2, including the T-pose and a pointing gesture, calculated by the gesture method as well. In addition, the table includes the *stop* gesture, *waving*, and the emblem *thumbs up*. These gestures were generated by manually steering the different joints in the desired position to reach a suitable posture. For the T-pose, the resulting posture is different for all three configurations. Configuration b has a constant inward medial rotation of the forearm, resulting in an altered T-pose. Because of the different joint configuration, the placement of the shoulder for Elvis-Cc differs from that of the other configurations. Large differences in posture can be noted for the *stop*-gesture. Both the Elvis-Ca and Elvis-Cc configuration lack the joint responsible for the flexion/extension of the wrist, which makes it difficult to create a recognizable posture for this gesture. The resulting postures show great similarities with the *hello*-gesture. For Elvis-Cb, the wrist allows to approximate the characteristic *stop*-gesture, with outstretched arm and bended wrist, the hand palm facing out. For the *hello* and *thumbs up* gesture, a suitable posture could be achieved for all configurations, however, for the latter, the range of possibilities to position the thumb upwards with a natural arm position was limited for Elvis-Cb, because of the missing rotational joint in the wrist. On the other hand, this was the only configuration for which a *clapping* motion could be generated. Pointing/placing motions could be generated for all configurations, but because of the lower articulated configuration, Elvis-Cc typically resulted in less natural configurations. Regarding the importance of the wrist, the flexion/extension motion in Elvis-Cb does not significantly contributes to the pointing capabilities of the robot, while the internal rotation of the forearm enables the hand to be well positioned for a potential grasping motion.

### 9.2.2 Covered model

The same gestures were created for the covered robot. The emotional expressions are listed in table 9.3, while the other gestures can be found in table 9.4. For the internal rotations, the fur was created as two separate pieces to guaran-

Table 9.1: Set of emotional expressions, generated by the uncovered prototype Elvis-Ca, Elvis-Cb and Elvis-Cc. The first column visualizes the robot's arm configuration.

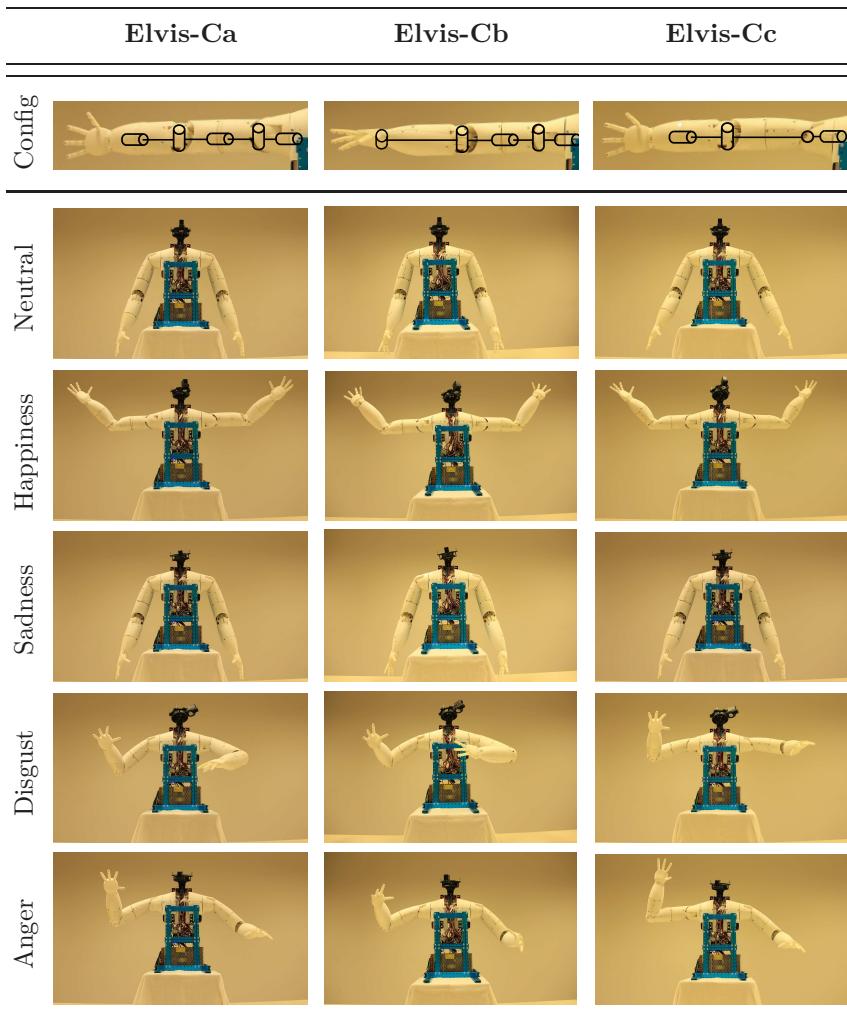


Table 9.2: Set of additional gestures, generated by the uncovered prototype Elvis-Ca, Elvis-Cb and Elvis-Cc. The first column visualizes the robot's arm configuration.

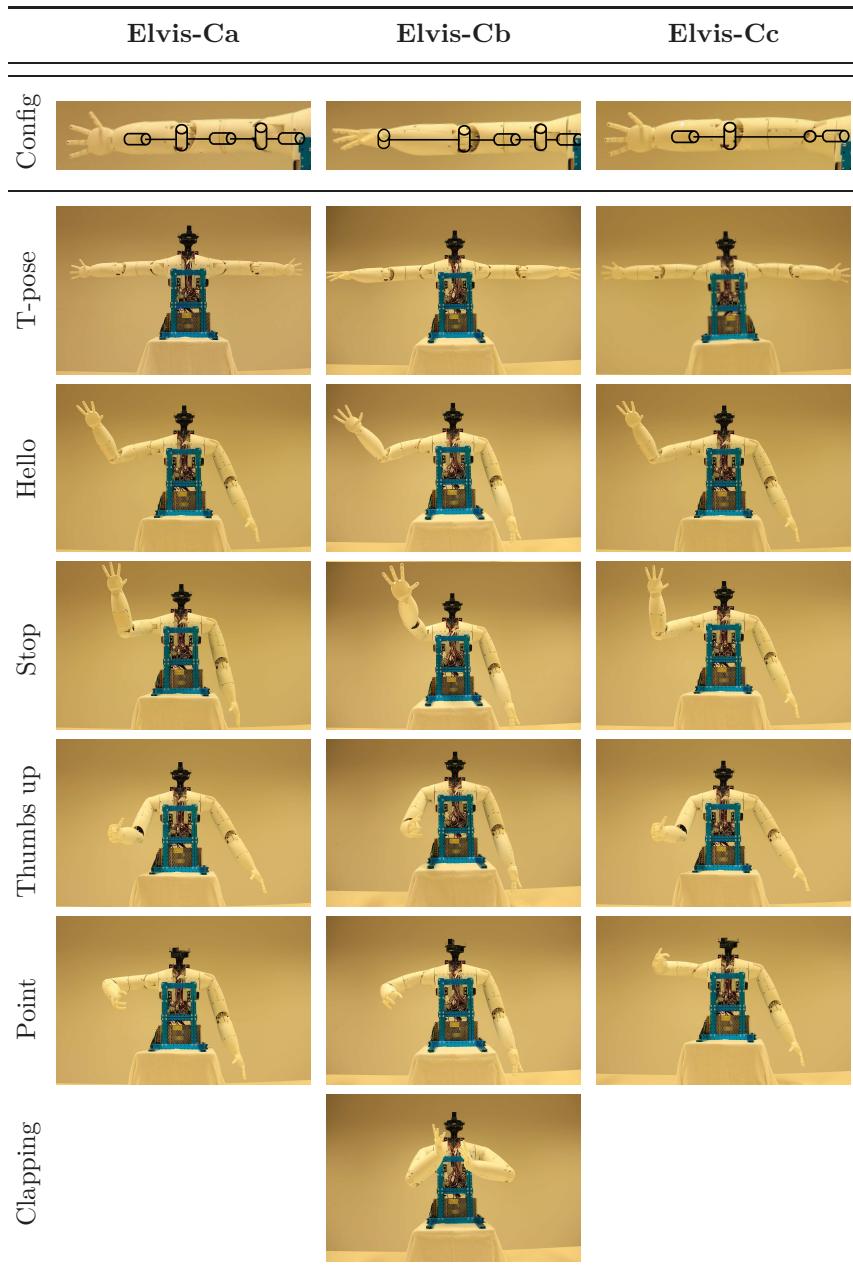


Table 9.3: Set of emotional expressions, generated by the covered prototype Elvis-Ca, Elvis-Cb and Elvis-Cc. The first column visualizes the robot's arm configuration.

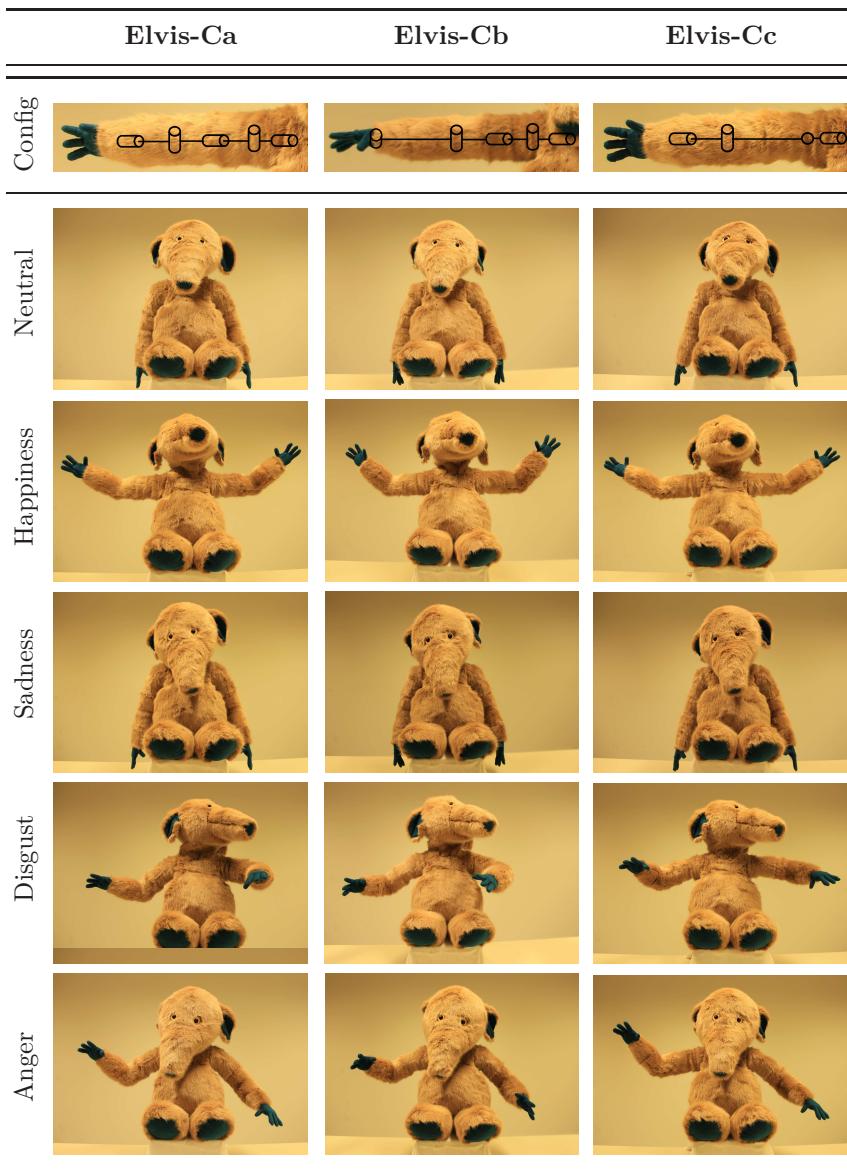


Table 9.4: Set of additional gestures, generated by the covered prototype Elvis-Ca, Elvis-Cb and Elvis-Cc. The first column visualizes the robot's arm configuration.

	Elvis-Ca	Elvis-Cb	Elvis-Cc
Config			
T-pose			
Hello			
Stop			
Thumbs up			
Place			

tee an optimal freedom of movement. For the elbow and middle shoulder joint, however, the fur consists of one single piece. Especially for the elbow joint, the tight cover resulted in a restriction of the motion and obstructed the servo to reach large angles. Instead of a joint range of  $120^\circ$  as for the uncovered model, only an angle of  $90^\circ$  could be reached. Higher angles could not be realised by the servo and resulted in a extreme heating of the motor because of the high currents. In general, the thick cover impedes the heat regulation of the robot, with the danger of overheating the servos.

The consequences of the diminished joint angle range for the elbow are clearly visible in the expression of *anger* and *disgust* (table 9.3), as well as in the *thumbs up* gesture for Elvis-Cb (table 9.4). The other gestures could be successfully generated by the covered robot. The fur masks a large portion of the differences in posture generated by the three different configurations. The *neutral* and *sad* postures are quasi identical for all three versions. The difference in hand position for the *happiness* gesture performed by Elvis-Cb, clearly visible for the uncovered model in table 9.1, is less notable for the covered model. Likewise, the differences in arm placement for emotional expressions for *disgust* and *anger* are less prominent. Also, the differences in shoulder placement for Elvis-Ca and Elvis-Cc are concealed by the fur, resulting in almost identical postures.

## 9.3 Conclusions

In this chapter, a number of results for the three Elvis configurations were listed. As a first set of gestures, a series of emotional expressions was calculated using the developed gesture method. For the uncovered robot, the variations in posture resulting from the differences in joint configurations agreed with what was expected from the precedent gesture study, discussed in chapter 8. A number of additional gestures were created by manually steering the joints in a desired position to reach an optimal posture. For a certain set of gestures, including the *stop* gesture and *clapping*, the presence of BAU 14 in Elvis-Cb contributed to the performance of the gesture. BAU 13, on the other hand, allows to orientate the hand independently of the placement of the elbow, which can, in specific, be interesting for grasping movements. For the covered robot, some issues were encountered. Whereas there is no problem for the uncovered model, the fur impedes the heat transfer of the servos, causing them to heat up significantly. Furthermore, the tight cover around the elbow joint obstructed the motion in a certain degree, resulting in a reduced joint angle range. For the generated emotional expressions, this mainly resulted in a lower performance of the expressions of *anger* and *disgust*. In general, the cover masks some of the variances in arm placement resulting from the differences in joint configuration. The configuration of Elvis-Cc, featuring only 4 DOF in each arm, performs surprisingly well for the tested gestures. Only for pointing and reaching gestures,

this morphology is less interesting because of the reduced dexterity, often leading to less natural postures. In the scope of the Probo-project, we believe that Elvis-Cb could be the optimal configuration, seen its performance of the tested emotional expressions, and the possibility of successfully generating additional gestures, such as *clapping* and the emblem *high five*, while keeping the design relatively low-complex with a total of 8 servos per arm.

PART III. ELVIS, A NEW SOCIAL ROBOT

# Part IV |

## Conclusions



# 10 | Conclusions

The way we communicate with technology is changing; from the use of a keyboard or touch screen, designers now aim to focus more on human-centered designs. Instead of asking the user to adapt its communication skills to the possibilities of the technological object, a more natural interaction is aimed for. Especially for robots that are aimed to operate in our daily lives, in our houses and workspaces, assisting us in a wide range of applications and, most importantly, collaborating with us in close interaction, it is crucial to guarantee intuitive communication possibilities. Therefore, social robots are designed with features supporting human communication skills and social cues. Several robots, capable of using speech, facial expressions or gestures have been developed. Different robots can feature different morphologies, and here, the correspondence problem comes into the picture. Traditional methods of implementing gestures use the robot's morphology to specify the motion patterns. Often, gestures are preprogrammed off-line for the specified configuration, stored in a database and replayed during interaction. Human video recordings or photographs can serve as a reference to create human-like postures, or the puppeteering technique can be used, whereby the physical robot is manually put in the desired position. Another commonly used technique to create motion patterns is by mapping motion capture data to the robot. Because the gestures are specified for a certain robot, the motion patterns cannot be used for other robots, which makes it difficult to share gestures, and more general, share control architectures in which gestures and motions are used. The aim of this thesis was therefore, to investigate how gestures can be generated in a generic way for different morphologies, to develop a software for this objective, and validate its different aspects.

## 10.1 A generic gesture method for social robots

### 10.1.1 Development of the gesture method

Seen their importance in both human-human and human-robot interaction, deictic gestures and emotional expressions were selected as focus group. While emotional expressions are explicit, full body actions representing an internal affective state and crucial features for creating socially accepted and fluent robot interactions, deictic gestures are, as pointing gestures, the most common type of illustrator gesture. To successfully generate deictic gestures, a correct positioning of the end-effector is crucial. However, to convey emotional expressions, the relative placement of the arm links, and therefore, the overall placement of the arm is important. To allow the gesture software to calculate both types of gestures, two working modes were developed; the *end-effector* mode for the first type of gestures, the *block* mode for the latter.

To ensure the generic aspect of the method, the framework of the software was constructed using a human base model, representing the rotational possibilities of a human. Since most social robots are built to the resemblance of a human, however less actuated, this base model comprises most of the available social robots. By evaluating the framework at runtime, using a limited set of parameters specifying the robot's configuration, inputted by the user, the software is therefore usable for the aforementioned set of robots.

A central feature of the method is a closed loop inverse kinematics algorithm that calculates the necessary joint angles for a selected gesture and robot configuration. The correct constraints to achieve the gesture are calculated and imposed by the corresponding working mode. To generate human-like postures in the end-effector mode, the available null-space motion is used to guide the joint angles to a predefined set of *minimum posture* angles, while generating affective deictic gestures is achieved by coupling these angles to the valence value of the current emotional state. As such, the openness of the posture is used as a first modification parameter to convey emotional content through an initially neutral behavior. Additionally, the speed at which the gesture is executed was coupled to both the valence and arousal value of the current affective state.

To verify whether a desired end-effector position is located in the robot's reach, a methodology to calculate an approximate workspace was implemented. By providing a mode mixer, the two developed working modes can be combined together to generate blended deictic gestures and emotional expressions. To guarantee a good performance of the gestures by physical robots, the inverse kinematics algorithm was equipped with a joint angle constraining feature, as well as a joint speed limitation.

### 10.1.2 Validation of the developed software

The different aspects of the method were validated on several robot models. The block mode proved to successfully generate emotional expressions for different configurations. For this validation, the morphologies of, amongst others, a fully actuated human model and the robots ASIMO, NAO and Justin were used. An online survey showed that the mapped gesture well resemble the target gestures.

In the end-effector mode, the implemented inverse kinematics algorithm was successfully extended with a cost function, responsible to guide the configuration into a natural or affective posture. Both grasping and pointing gestures could be generated for a series of configurations, and the effect of the modulation parameters clearly resulted into readable emotional states.

The implementation of the mode mixer and priority levels allowed to mix different types of gestures, to generate blended gestures. This feature was, similarly as for the aforementioned modalities, validated using the virtual model of different robots, including ASIMO, NAO and Justin.

The implemented joint limitation algorithm was tuned to guarantee optimal results for our software, and allowed to validate the complete developed gesture method on the physical model of several robots. The different possibilities of the method were assembled into one validation, by a scenario whereby the robot tells a little story, while using different types of gestures calculated by the method. The story was performed by NAO, Pepper and Romeo.

### 10.1.3 Limitations and possible improvements of the method

Because of the highly generic aspect of the method, a large number of variables and variable structures are passed through the different functions of the software. This makes the method computationally very heavy, and too slow to be used in realtime calculations.

The developed gesture method is presented in this thesis as a separate software module, that allows the calculation of joint trajectories for gestures chosen from a graphical user interface. In theory, the software could be incorporated in a cognitive behavior controller, whereby the gestures are initiated as a result to external and internal stimuli. However, because of the high computational workload it is not straightforward to integrally integrate it into such an architecture. The EU-project DREAM, whereof the Robotics and Multibody Mechanics Research Group is a partner, aims to develop a complete platform-independent cognitive architecture for robots used in therapies [36]. To ensure the generic aspect of the architecture, the behavior generation system selects abstract behaviors, based on the user's profile, interaction events and behavior databases. To realize the triggered gestures, while avoiding the system to

#### PART IV. CONCLUSIONS

become computationally expensive, the abstract behaviors are mapped to the chosen robot configuration by using stored joint trajectories that were generated a priori by the generic gesture method.

The gesture database, containing the key postures and time constraints of the emotional expressions, is currently limited to 5 basic emotions, used to evaluate the possibilities of the method. Emotional expressions corresponding to other affective states can be easily added to the library. Since humans use several postures and gestures to express one single affective state, different emotional expressions corresponding to one emotion can be incorporated as well, to allow for some variance of the gestures, contributing to a natural human-robot interaction. Finally, also other types of gestures can be calculated by the block mode. Since in the block mode, the orientation of the end-effector of all available blocks is imposed, all gestures whereby the overall pose of the arms and upper-body is important, can be calculated. Possible examples are the emblem for waving, gestures used in imitation games or dance motions.

The algorithm to search an optimal end-effector pose in the pointing condition of the end-effector mode could be improved. Since for a desired pointing position, multiple combinations of end-effector position and orientation are possible, the optimal combination is selected by a cost function, trying to minimize the deviation between the joint angles and a set of predefined *minimum posture* angles. To find the possible end-effector poses meeting the pointing constraint, the last link of the arm chain is gradually virtually extended, and the desired pointing position is imposed on the virtual end-effector. For every virtual extension, the necessary joint angles to reach the most natural pose can be calculated by the inverse kinematics algorithm by using the DH-parameters for the extended configuration, in the same manner as when joint angles are calculated for the normal configuration in a *place-at* condition. However, for the normal configuration, an approximate workspace is calculated and used to examine if a desired end-effector position is in reach of the robot. Calculating a similar workspace for each virtual, extended configuration would decrease the calculation speed significantly. Therefore, in the current version of the software, the necessary joint angles for the virtual configuration are calculated, whereafter they are imposed on the real configuration. The corresponding position of the real end-effector, calculated using direct kinematics, is then verified to lay in the robot's workspace. If this is not the case, the current solution is rejected, and the search of the most natural pose for the next virtual extension is started. At the end, the cost function selects the optimal solution from the resulting collection of postures. However, when a large number of calculated solutions are located outside the workspace, the resulting configurations in the collection to chose an optimal result can become scarce or even non-existing. A beneficial adjustment would be that, if a solution is not in reach of the robot, a new iteration is started with *the same* extension parameters, to look for the second best natural posture for this condition.

To generate affective gestures, the method uses the posture's openness and the motion speed as modification parameters to modulate an initially neutral behavior. To increase the recognizability of the conveyed state of affect, the effect of additional parameters, such as the fluidity and repetition of the motion, could be investigated.

In this stage, the use of the fingers is not implemented in the gesture software. Only in the pointing condition, the length of the index finger is used to position the end-effector at the fingertip, while in the place-at condition, the end-effector is located in the center of the hand palm. Incorporating the joints of the fingers into the human base model, and consequently into the framework of the method, would permit the software to calculate a broad collection of additional gestures. To calculate several emblems, such as the *thumbs up* gesture, a similar approach as for mixing emotional expressions and deictic gestures could be used; the desired position of hand could be handled by the end-effector mode, and combined with the necessary finger motion using the mode mixer. In addition, by incorporating finger motion, the quality of several gestures could be improved. Possible examples are clenching the left hand's fingers during a right-handed deictic gesture performed in an angry mood, or closing the fingers in a certain extend to minimize the posture when expressing fear.

## 10.2 Use of the method for design purposes

To calculate gestures for a certain robot using the developed software, the configuration is specified to the program by its Denavit-Hartenberg parameters and a maximum of four rotation matrices. In addition, to keep the joint trajectories between the physical boundaries, joint angle limits and speeds can be imposed. Once this information is passed to the software, gestures to be calculated can be chosen from the graphical user interface, or a position for a pointing or grasping gesture can be imposed. As such, different gestures can be calculated with a minimal effort. Since a selected set of gestures can be created for different morphologies, the influence of specific design aspects can be revealed by visualizing the calculated gestures on a single virtual model. Consequences of joint angle range can be studied, as well as the effect of the placement of a certain joint.

This methodology is mainly interesting to give insights in the required complexity for a well-defined task and for finding an optimal cost/efficiency trade-off. For robots intended to perform a variety of different tasks in different applications, a more complex robot of course gives more freedom regarding the motions to be performed. But for a robot designed for a specific task or application, a careful consideration of different morphologies can help in finding an optimal design for a desired degree of complexity, cost and expressibility.

## 10.3 Elvis

### 10.3.1 Development of the robot Elvis

To illustrate the above methodology, a new social robot, called Elvis, was developed. Elvis was developed in the frame of the Probo project and was based on the experiences of interactions with the first prototype of the robot. Since the robot is aimed to interact with children on an emotional level, the possibility of expressing affective states is an important requirement. Therefore, a gesture study was performed for a series of different configurations, using a set of emotional expressions. Based on the generated gestures, three configurations were selected to be studied in more detail. In addition, the gesture software was used to select an optimal, reduced joint range for all DOF. The three configurations, denoted as Elvis-Ca, -Cb and -Cc were physically realised by designing a semi-modular arm system, whereof different joint modules can be switched. Finally, gestures were created for all three robot versions by imposing the calculated joint trajectories.

### 10.3.2 Limitations and possible improvements of Elvis

The characteristic huggable appearance of Probo was passed to Elvis by covering the mechanical system with a customized fur jacket. For most joints, the cover was made in two separate pieces and independently attached to the input and output link, providing an optimal freedom of movement. For the elbow joint, however, the cover was made in one piece for aesthetic reasons. The tight cover obstructed the motion in a certain degree, causing the range of motion to be limited to 90° instead of 120°. Higher angles could not be reached by the servo motor because of the large friction. To work properly and guarantee the provided joint range, the elbow cover should be redesigned.

Another issue introduced by the cover is the impeded heat transfer. The warmth created by the servos is kept inside the cover, with the danger of overheating the motors.

For all joints, hobby servos are used. For these motors, the positional feedback is restricted to the servo's internal control circuit and not fed to the global control software. Therefore, when the desired position cannot be reached, because of, for example, a person blocking the motion or, as for the elbow joint, the fur obstructing the motion, the global control software is not aware of that, while the servo keeps pulling higher currents. This leads to seriously overheating, and even destruction of the servo. The current design of the robot is mainly suited to demonstrate the generation of gestures and the possibilities of the developed gesture software. The aforementioned problem could be partially solved by providing series elastic actuators (SEA) and thus reimplementing the

servo savers. Additionally, the global control software should be improved. To guarantee a safe interaction, especially with children, a global feedback concerning the servos' current position should be provided. Another possibility to allow a safer human-robot interaction is the use of a global impedance control loop instead of the currently used position control.

To improve the mechanical design and reduce the necessary motor torques, the use of gravity compensation techniques could be studied.

While the initial version of Probo focussed on facial expressions and therefore featured 19 DOF in the head, the main interest in the design of Elvis was the development of an actuated arm system to generate gestures. A next prototype should include both aspects to optimize the recognizability of the generated emotional expressions and generally contribute to a natural human-robot interaction.

PART IV. CONCLUSIONS

# A | List of publications

## Publications in International Journals

1. **Greet Van de Perre**, Michael Van Damme, Dirk Lefeber, and Bram Vanderborght. Development of a generic method to generate upper-body emotional expressions for different social robots. *Advanced Robotics*, 29(9):597–609, 2015.
2. **Greet Van de Perre**, Albert De Beir, Hoang-Long Cao, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Reaching and pointing gestures calculated by a generic gesture system for social robots. *Robotics and Autonomous Systems*, 83:32–43, 2016.
3. **Greet Van de Perre**, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Generic method for generating blended gestures and affective functional behaviors for social robots. *Autonomous Robots*, 1–12, 2017.
4. **Greet Van de Perre**, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Studying design aspects for social robots using a generic gesture method. *International Journal of Social Robotics*, *In review*.
5. **Greet Van de Perre**, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Development of Elvis, a new social robot for affective and functional gesturing. *IEEE Robotics and Automation letters*, *In review*.
6. **Greet Van de Perre**, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing. *International Journal of Robotics Research*, *In review*.

## APPENDIX A

7. Ramona Simut, Cristina Pop, Johan Vanderfaeillie, **Greet Van de Perre**, Bram Vanderborght, and Dirk Lefeber. "can you cure me?": Children with autism spectrum disorders playing a doctor game with a social robot. *Journal of School Health*, 3(3):1–9, 2016.
8. Ramona Simut, Johan Vanderfaeillie, Andreea Peca, **Greet Van de Perre**, and Bram Vanderborght. Children with autism spectrum disorders make a fruit salad with Probo, the social robot. an interaction study. *Journal of Autism and Developmental Disorders*, 46(1):113–126, 2016.
9. Albert De Beir, Hoang-Long Cao, Pablo Gómez Esteban, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. Enhancing emotional facial expressiveness on NAO: A case study using pluggable eyebrows. *International Journal of Social Robotics*, 2016:1–9, 2016.
10. Ramona Simut, **Greet Van de Perre**, Cristina Pop, Jelle Saldien, Johan Vanderfaeillie, Daniel D. David, Dirk Lefeber, and Bram Vanderborght. Probogotchi: a novel edutainment device as a bridge for interaction between a child with ASD and the typically developed sibling. *Journal of Evidence-Based Psychotherapies*, 16(1):91–112, 2016.
11. Pablo Gómez Esteban, **Greet Van de Perre**, Bram Vanderborght, Hoang-Long Cao, Albert De Beir, Paul Baxter, Tony Belpaeme, Erik Billing, Haibin Cai, Mark Coeckelbergh, Cristina Costescu, Daniel David, Zhaojie Ju, James Kennedy, Honghai Liu, Alexandre Mazel, Amit Pandey, Kathleen Richardson, Emmanuel Senft, Serge Thill, David Vernon, Hui Yu, and Tom Ziemke. How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. *Paladyn*, 8(1):18–38, 2017.
12. Hoang-Long Cao, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. A survey on behavior control architectures for social robots in healthcare interventions. *International Journal of Humanoid Robotics*, 14(4):1–24, 2017.
13. Hoang-Long Cao, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. A collaborative homeostatic-based behavior controller for social robots in human-robot interaction experiments. *International Journal of Social Robotics*, 10(2):1–16, 2017.
14. Hoang-Long Cao, **Greet Van de Perre**, James Kennedy, Emmanuel Senft, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, Tony Belpaeme, Dirk Lefeber, and Bram Vanderborght. A personalized and

platform-independent behavior control system for social robots in therapy: development and applications. *IEEE Transactions on Cognitive and Developmental Systems*, PP(99):1–13, 2018.

## Conference papers

1. **Greet Van de Perre**, Ramona Simut, Bram Vanderborght, Jelle Saldien, and Dirk Lefeber. About the design of the social robot Probo, facilitator for ASD therapies. In *9th National Congress on Theoretical and Applied Mechanics, Brussels, 9-10-11 May 2012*, pages 1–6, 2012.
2. Ramona Simut, **Greet Van de Perre**, Cristina Pop, Bram Vanderborght, Jelle Saldien, Alina S. Rusu, Sebastian Pintea, Johan Vanderfaillie, Dirk Lefeber, and Daniel D David. The huggable social robot Probo for social story telling for robot assisted therapy with ASD children. In *International Conference on Social Robotics, works-in-progress, 23-25.22.2011, Amsterdam, Netherlands*, pages 97–100, 2011.
3. Bram Vanderborght, Ramona Simut, **Greet Van de Perre**, Cristina Pop, Daniel D David, Johan Vanderfaillie, and Dirk Lefeber. The social robot Probo as interaction partner for autistic children. In *IROS13 workshop: Towards Social Humanoid Robots: What makes interaction human-like?*, pages 1–4, 2013.
4. Hoang Long Cao, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. Robee: A homeostatic-based social behavior controller for robots in human-robot interaction experiments. In *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 516–521, 2014.
5. Hoang Long Cao, **Greet Van de Perre**, Ramona Simut, Cristina Pop, Andreea Peca, Dirk Lefeber, and Bram Vanderborght. Enhancing my keepon robot: A simple and low-cost solution for robot platform in human-robot interaction studies. In *Proceedings of the 2014 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 555–560. IEEE, 2014.
6. Pablo Gómez Esteban, Hoang-Long Cao, Albert De Beir, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. A multilayer reactive system for robots interacting with children with autism. In *Proceedings of the Fifth International Symposium on New Frontiers in Human-Robot Interaction*, pages 1–4, 2016.

## Conference abstracts

1. **Greet Van de Perre**, Ramona Simut, Bram Vanderborght, Jelle Saldien, and Dirk Lefeber. A gesture system for social robots. In *HFR 2012, 5th workshop on Human-Friendly Robotics, Brussels, Belgium, October 18-19, 2012.*, 2012.
2. **Greet Van de Perre**, Michael Van Damme, Dirk Lefeber, and Bram Vanderborght. Development of a gesture method for social robots. In *International Summer School on Social Human-Robot Interaction*, 2013.
3. **Greet Van de Perre**, Michael Van Damme, Dirk Lefeber, and Bram Vanderborght. First validation of a generic method for emotional body posture generation for social robots. In *Conference on Human-Robot Interaction, lbr, 3-9 March 2014.*, pages 308–309, 2014.
4. **Greet Van de Perre**, Michael Van Damme, Dirk Lefeber, and Bram Vanderborght. Generic method for emotional gesture generation for social robots: first validation. In *Fifth EUCoglll Members Conference, Bochum "embodied communication", 19-20 March 2014.*, 2014.
5. **Greet Van de Perre**, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Generating gestures for different robot morphologies through one generic gesture system. In *International Summer School on Social Human-Robot Interaction*, 2017.
6. Ramona Simut, **Greet Van de Perre**, Cristina Pop, Bram Vanderborght, Jelle Saldien, Alina S. Rusu, Sebastian Pintea, Johan Vanderfaillie, Dirk Lefeber, and Daniel D David. Is the social robot Probo an added value for social story intervention for children with ASD? In *7th ACM/IEEE International Conference on Human Robot Interaction (HRI2012), Late-breaking reports (LBR), March 5-8, 2012, Boston, Massachusetts, USA*, pages 1–2. IEEE, 2012.
7. Albert De Beir, Hoang-Long Cao, Pablo Gómez Esteban, **Greet Van de Perre**, and Bram Vanderborght. Enhancing NAO expression of emotions using pluggable eyebrows. In *Proceedings of the 1st International Conference on Social Robots in Therapy and Education*. Windesheim Flevoland, 2015.
8. Hoang-Long Cao, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. Toward a platform-independent social behavior architecture for multiple therapeutic scenarios. In *Proceedings of the 1st International Conference on Social Robots in Therapy and Education*. Windesheim Flevoland, 2015.

9. Hoang-Long Cao, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, **Greet Van de Perre**, and Bram Vanderborght. A platform-independent robot control architecture for multiple therapeutic scenarios. In *Proceedings of the Fifth International Symposium on New Frontiers in Human-Robot Interaction*, volume 1602.05456, pages 1–5, 2016.

## Book chapters

1. Hoang-Long Cao, Cristina Pop, Ramona Simut, Albert De Beir, **Greet Van de Perre**, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Probolino: A portable low-cost social device for home-based autism therapy. In Adriana Tapus, Elisabeth André, Jean-Claude Martin, FranÃ§ois Ferland, and Mehdi Ammi, editors, *International Conference on Social Robotics*, Lecture Notes in Computer Science, pages 93–102. Springer, 2015.
2. Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Ramona Simut, **Greet Van de Perre**, Dirk Lefeber, and Bram Vanderborght. An end-user interface to generate homeostatic behavior for nao robot in robot-assisted social therapies. In *Advances in Computational Intelligence*, volume 10306, pages 609–619. Springer International Publishing, 2017.

APPENDIX A

# B | Calculation details

## B.1 Direct kinematics

### B.1.1 Denavit-Hartenberg convention

The Denavit-Hartenberg (DH) convention is a common approach for specifying the geometry of a kinematic chain. In a first step, a frame of reference is attached to each joint  $i$ . Hereby, the  $z_i$  axis is directed along the rotation axis of joint  $i + 1$ , while the  $x_i$ -axis is chosen along the common normal to the joint axes of joint  $i$  and  $i + 1$  (see figure B.1). To specify the transformation of the reference frame  $x_iy_iz_i$  with respect to the frame  $x_{i-1}y_{i-1}z_{i-1}$ , and therefore, the placement of joint  $i + 1$  with respect to that of the previous one, four parameters are used:

- $a_i$  is defined as the length of the common normal to the axes of joint  $i$  and  $i + 1$ .
- $\alpha_i$  is the twist angle between  $z_{i-1}$  and  $z_i$  around  $x_i$ .
- $d_i$  is the offset to the common normal along  $z_{i-1}$ .
- $\theta_i$  is the angle between  $x_{i-1}$  and  $x_i$  around  $z_{i-1}$ .

To specify the orientation of frame  $x_iy_iz_i$  with respect to the frame  $x_{i-1}y_{i-1}z_{i-1}$ , the homogeneous transformation between both frames is written using the DH-parameters. The roto-translation around and along  $z_{i-1}$  to go from the initial reference frame  $x_{i-1}y_{i-1}z_{i-1}$  to the intermediate frame  $x_{i'}y_{i'}z_{i'}$  can be written as:

$${}^{i-1}A_{i'} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

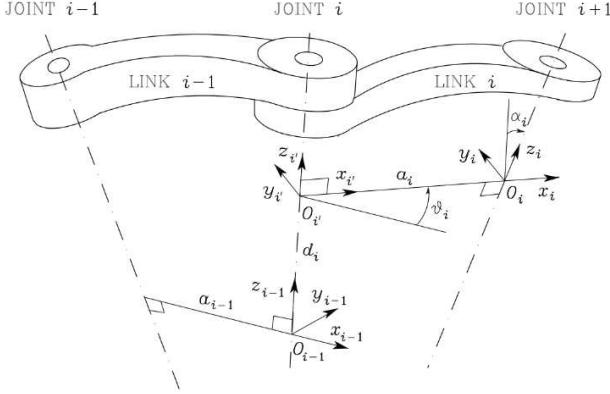


Figure B.1: Denavit-Hartenberg parameters: definition of the reference frames and parameters [101].

While the roto-translation around and along  $x_i$  to go from  $x_{i'}y_{i'}z_{i'}$  to  $x_iy_iz_i$  equals:

$${}^{i'}A_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) & 0 \\ 0 & \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

By combining both equations, an expression for  ${}^{i-1}A_i$ , the Denavit-Hartenberg transformation matrix for joint  $i$ , can be found:

$${}^{i-1}A_i(q_i) = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos\theta_i \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

Here,  $q_i$  is the parameter related to the joint value. For a revolute joint,  $q_i = \theta_i$ , while for a prismatic joint,  $q_i = d_i$ . The relation between the reference frame  $x_b y_b z_b$ , placed in the base of the manipulator, and the frame attached to the end-effector  $x_e y_e z_e$ , can be calculated by combining the Denavit-Hartenberg matrices for each joint [101]:

$${}^bA_e = {}^bT_0 {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n) {}^nT_e \quad (\text{B.4})$$

When we consider a kinematic chain composed of revolute joints only, all parameters except  $\theta_i$  are constant. Therefore, by specifying the DH-parameters

to the gesture software, equation B.4 can be calculated as a function of the current joint angles.

### B.1.2 Calculation of the end-effector pose

Since equation B.4 specifies the homogeneous transformation between the base and end-effector, the orientation  $\phi_e$  and position  $p_e$  of the end-effector with respect to the base frame, together denoted as the *pose*  $x_e$ , can be extracted from this matrix [101]:

$$x_e = \begin{bmatrix} p_e(q) \\ \phi_e(q) \end{bmatrix} \quad (\text{B.5})$$

$${}^b A_e = \begin{bmatrix} R & p_e \\ 0 & 1 \end{bmatrix} \quad (\text{B.6})$$

The position  $p_e$ , expressed in the Cartesian coordinates  $\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$  can be read

directly from the fourth column of matrix  ${}^0 A_n$ . To calculate the remainder of the pose, first an orientation representation needs to be chosen. For our software, the *zyx*-Euler angles are used. Therefore,  $R$  can also be written as:

$$\begin{aligned} R &= R_z(\psi)R_y(\beta)R_x(\phi) \\ &= \begin{bmatrix} \sin(\beta) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\psi)\cos(\beta) & \cos(\psi)\sin(\beta)\sin(\phi)-\sin(\psi)\cos(\phi) & \cos(\psi)\sin(\beta)\cos(\phi)+\sin(\psi)\sin(\phi) \\ \sin(\psi)\cos(\beta) & \sin(\psi)\sin(\beta)\sin(\phi)+\cos(\psi)\cos(\phi) & \sin(\psi)\sin(\beta)\cos(\phi)-\cos(\psi)\sin(\phi) \\ -\sin(\beta) & \cos(\beta)\sin(\phi) & \cos(\beta)\cos(\phi) \end{bmatrix} \quad (\text{B.7}) \end{aligned}$$

The Euler angles  $\psi$ ,  $\beta$  and  $\phi$  can then be found as follows:

$$\psi = \text{atan}\left(\frac{r_{21}}{r_{11}}\right) \quad (\text{B.8})$$

$$\beta = \text{asin}(-r_{31}) \quad (\text{B.9})$$

$$\phi = \text{atan}\left(\frac{r_{32}}{r_{33}}\right) \quad (\text{B.10})$$

## B.2 Interpolation between postures

In the gesture software, the necessary joint trajectories to reach a desired gesture are calculated. For the block mode, the gesture database consists of the specification of the desired end-effector orientations, together with the specification of the duration of the gesture  $t_{gesture}$ . For a gesture calculated by the end-effector mode, the duration of the gesture depends on the chosen affective state. To generate a trajectory towards the desired posture, multiple intermediate postures are calculated. The software uses a fixed time-step  $\Delta t$ , and by combining this with the desired timing of the gesture, the number of intermediate postures  $n_{i,p}$  can be obtained:

$$n_{i,p} = \frac{t_{gesture}}{\Delta t} \quad (\text{B.11})$$

### B.2.1 Interpolation in the block mode

For the block mode, the  $n_{i,p}$  intermediate postures are found using a linear interpolation between the current and desired orientation.

For a rotation with amplitude  $\gamma$  around the rotation axis specified by the unity vector  $1_k = (n_x, n_y, n_z)$ , the rotation matrix can be written as:

$$R = \begin{bmatrix} (1 - \cos\gamma)n_x^2 + \cos\gamma & (1 - \cos\gamma)n_xn_y - \sin\gamma n_z & (1 - \cos\gamma)n_xn_z + \sin\gamma n_y \\ (1 - \cos\gamma)n_y n_x + \sin\gamma n_z & (1 - \cos\gamma)n_y^2 + \cos\gamma & (1 - \cos\gamma)n_y n_z - \sin\gamma n_x \\ (1 - \cos\gamma)n_z n_x - \sin\gamma n_y & (1 - \cos\gamma)n_z n_y + \sin\gamma n_x & (1 - \cos\gamma)n_z^2 + \cos\gamma \end{bmatrix} \quad (\text{B.12})$$

By setting this expression equal to the  $zyx$ -Euler angles rotation matrix specifying the desired orientation with respect to the current orientation, calculated using the specifications of the database, the corresponding rotation axis and amplitude can be found as follows:

$$\gamma = \text{atan} \left[ \frac{\sqrt{\frac{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}{2}}}{\frac{r_{11} + r_{22} + r_{33} - 1}{2}} \right] \quad (\text{B.13})$$

$$1_k = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{r_{32} - r_{23}}{2\sin\gamma} \\ \frac{r_{13} - r_{31}}{2\sin\gamma} \\ \frac{r_{21} - r_{12}}{2\sin\gamma} \end{bmatrix} \quad (\text{B.14})$$

An interpolation between the current and desired orientation is then established by a linear progress of the rotation amplitude. The necessary rotation matrix at time  $t_i$  is calculated using equation B.12 for the calculated rotation axis and the amplitude  $\gamma(t_i)$ :

$$\gamma(t_i) = \frac{\gamma_{start}}{t_{gesture}} t_i \quad (\text{B.15})$$

### B.2.2 Interpolation in the end-effector mode

In the end-effector mode, intermediate postures are calculated by an interpolation between the initial end-effector position  $p_{e_{start}}$  and the desired end-effector position  $p_{e_d}$ . A first attempt for the trajectory is a linear interpolation:

$$p_e(t_i) = \frac{(p_{e_d} - p_{e_{start}})}{t_{gesture}} t_i + p_{e_{start}} \quad (\text{B.16})$$

If a part of linear trajectory falls outside the workspace of the robot, an alternative, curved trajectory is calculated. This trajectory will lie in the plane defined by the initial end-effector position  $p_{e_{start}}$ , the desired end-effector position  $p_{e_d}$ , and the initial position of the elbow  $p_{elbow_{start}}$ . The latter can be found in the fourth column of the corresponding DH-matrix. The equation of the plane can be written as follows:

$$ax + by + cz = 1 \quad (\text{B.17})$$

whereby  $a$ ,  $b$  and  $c$  can be determined by solving the system of equations obtained by imposing the coordinates of the three known points lying in this plane, namely  $p_{e_{start}}$ ,  $p_{e_d}$ , and  $p_{elbow_{start}}$ , using Cramer's rule.

A new reference frame  $x_c y_c z_c$  can then be determined, whereby the  $x_c$ -axis is chosen along the connection line between the initial and desired end-effector position and the  $z_c$ -axis perpendicular on the plane:

$$x_c = p_{e_d} - p_{e_{start}} \quad (\text{B.18})$$

$$z_c = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (\text{B.19})$$

$$y_c = z_c \times x_c \quad (\text{B.20})$$

The curved trajectory is the circular arc, defined by the points  $p_{e_{start}}$ ,  $p_{e_d}$  and a third *way point*  $p_w$ . This way point determines the exact shape and curvature of the trajectory. The initial position of this point lies on the connection line

between the start and end-effector pose. Its exact position depends on the amount at which the linear trajectory falls out of the workspace. The position of this point is expressed in the  $x_{c'}y_{c'}z_{c'}$ -frame, which is the reference frame parallel to  $x_c y_c z_c$ , but placed in the base of the kinematic chain:

$$x_{c'}y_{c'}z_{c'}p_w = \begin{bmatrix} x_{c'}y_{c'}z_{c'}x_w \\ x_{c'}y_{c'}z_{c'}y_w \\ x_{c'}y_{c'}z_{c'}z_w \end{bmatrix} \quad (\text{B.21})$$

To obtain a curved trajectory, the point  $p_w$  is shifted along the  $y_{c'}$ -axis according to the following equation:

$$x_{c'}y_{c'}z_{c'}y_w = \text{sign}(y_{e_{start}}) \sqrt{\text{armlength}^2 - x_{c'}y_{c'}z_{c'}x_w^2 - x_{c'}y_{c'}z_{c'}z_w^2} \quad (\text{B.22})$$

The equation of the circle can be written as:

$$x^2 + y^2 + mx + ny + l = 0 \quad (\text{B.23})$$

whereby the value of  $m$ ,  $n$  and  $l$  can be determined by solving the system of equations obtained by imposing the coordinates of the three points  $x_{c'}y_{c'}z_{c'}p_{e_{start}}$ ,  $x_{c'}y_{c'}z_{c'}p_{e_d}$  and  $x_{c'}y_{c'}z_{c'}p_w$ .

The intermediate desired end-effector position at time  $t_i$  can then be obtained as follows. In a first step, the desired position obtained by linear interpolation using equation B.16 is converted to the  $x_{c'}y_{c'}z_{c'}$  frame. From the  $x_{c'}$ -coordinate, the corresponding point on the circle can be determined by solving the quadratic equation B.23. After transferring it back to the chain's base frame, this position is finally used as the desired intermediate end-effector position at time  $t_i$ .

### B.3 Inverse kinematics

For an imposed end-effector pose, the corresponding joint angles need to be calculated. This is done using a closed loop inverse kinematics algorithm. In a first step, the derivative of the joint angles is calculated using the following equation (see section 3.3.1 and 4.1):

$$\dot{q} = J_A^\dagger(q) (\dot{x}_d + K(x_d - x_e)) + (I - J_A^\dagger(q) J_A(q)) \dot{q}_0 \quad (\text{B.24})$$

Here,  $x_d$  is the desired end-effector pose. For the block mode, only the orientation is imposed and therefore,  $x_d$  is reduced to  $\phi_d$ , specifying the desired  $zyx$ -Euler angles. For the end-effector mode, on the other hand, the end-effector

position is imposed and therefore,  $x_d = p_d$ .  $x_e$  is the current end-effector pose, which can be calculated based from the DH-matrix as demonstrated in the previous section.  $\dot{x}_d$  can be calculated by dividing the offset to the desired pose by the time at which it should be reached, which is the time-step  $\Delta t$ .

$J_A^\dagger(q)$  is the Moore-Penrose pseudo inverse of the analytical Jacobian  $J_A(q)$ . The analytical Jacobian is obtained by differentiating the direct kinematics function with respect to the joint variables:

$$J_A(q) = \frac{\partial x_e}{\partial q} \quad (\text{B.25})$$

with

$$x_e = \begin{bmatrix} p_e(q) \\ \phi_e(q) \end{bmatrix} \quad (\text{B.26})$$

Since in this block mode only the orientation is used,  $J_A(q)$  is here reduced to its rotational part only, while for the end-effector mode, only the translational part is used.

To calculate an expression for the analytical Jacobian  $J_A$ , first the geometric Jacobian  $J$  is determined.

### B.3.1 Geometric Jacobian

The geometric Jacobian describes the relation between the joint velocities and the corresponding end-effector linear velocity  $\dot{p}_e$  and angular velocity  $\omega_e$  [101]:

$$\begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} \dot{q} = J(q)\dot{q} \quad (\text{B.27})$$

The expression for the geometric Jacobian can be composed using the chains' DH-matrices.

The translational part of the Jacobian can be calculated as follows:

$$J_v = \begin{bmatrix} J_{v_1} & \dots & J_{v_n} \end{bmatrix} \quad (\text{B.28})$$

with

$$J_{v_i} = {}^0 z_{i-1} x({}^0 p_n - {}^0 p_{i-1}) \quad (\text{B.29})$$

The angular velocity part of the Jacobian, on the other hand, can be calculated using following expression:

$$J_\omega = [ \ J_{\omega_1} \ \dots \ J_{\omega_n} ] \quad (\text{B.30})$$

with

$$J_{\omega_i} = {}^0 z_{i-1} \quad (\text{B.31})$$

Here,  ${}^0 z_{i-1}$  represents the direction of the  $z$ -axis of the reference frame attached to joint  $i$ , with respect to the reference frame attached to the first joint of the kinematic chain. This vector is available in the third column of the DH-matrix specifying the transformation between both reference frames, namely  ${}^0 A_{i-1}$ .  ${}^0 p_{i-1}$  represents the position of the origin of reference frame attached to joint  $i$  with respect to that attached to the first joint, and can be obtained from the fourth column of the DH-matrix  ${}^0 A_{i-1}$ . Similarly,  ${}^0 p_n$  can be obtained from the fourth column of  ${}^0 A_n$

### B.3.2 Calculation of the analytical Jacobian

The analytical Jacobian differs from the geometric Jacobian in the rotational part, since the angular velocity of the end-effector with respect to the base frame,  $\omega_e$ , is not equal to the derivative of the rotation part of the pose  $\dot{\phi}_e$ . The relation between  $\omega_e$  and  $\dot{\phi}_e$  can be written as [101]:

$$\omega_e = B(\phi_e) \dot{\phi}_e \quad (\text{B.32})$$

whereby the exact expression of the matrix  $B$  depends on the chosen orientation representation. For  $zyx$ -Euler angles,  $B$  becomes:

$$B = \begin{bmatrix} 0 & -\sin(\psi) & \cos(\psi)\cos(\beta) \\ 0 & \cos(\psi) & \sin(\psi)\cos(\beta) \\ 1 & 0 & -\sin(\beta) \end{bmatrix} \quad (\text{B.33})$$

The relation between the geometric and analytic Jacobian can then be written as:

$$\begin{aligned} J(q)\dot{q} &= \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & B(\phi_e) \end{bmatrix} \begin{bmatrix} \dot{p}_e \\ \dot{\phi}_e \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & B(\phi_e) \end{bmatrix} J_A(q)\dot{q} \end{aligned} \quad (\text{B.34})$$

and therefore:

$$J_A(q) = \begin{bmatrix} I & 0 \\ 0 & B^{-1}(\phi_e) \end{bmatrix} J(q) \quad (\text{B.35})$$

Since both the geometric Jacobian  $J$  and the matrix  $B$  can be calculated from the DH-parameters for the current value of the joint angles  $q_i$ , the expression for the analytical Jacobian can be evaluated as well.

## B.4 Runge-Kutta

To calculate the joint angles  $q_i$  from their derivatives  $\dot{q}_i$ , the Runge-Kutta method is used [152]. Runge-Kutta is an iterative method to numerically integrate differential equations. In our gesture software, the initial value problem is specified as follows:

$$\dot{q} = f(q) \quad \text{and} \quad q_{t_0} = q_0 \quad (\text{B.36})$$

whereby  $f(q)$  is specified by equation B.24.

An approximation for  $q_{t_{i+1}}$ , denoted as  $q_{i+1}$ , is calculated using the present value  $q_i$  and the weighted average of four increments:

$$q_{i+1} = q_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{B.37})$$

with

$$k_1 = f(t_i, q_i) \quad (\text{B.38})$$

$$k_2 = f\left(t_i + \frac{h}{2}, q_i + h\frac{k_1}{2}\right) \quad (\text{B.39})$$

$$k_3 = f\left(t_i + \frac{h}{2}, q_i + h\frac{k_2}{2}\right) \quad (\text{B.40})$$

$$k_4 = f(t_i + h, q_i + hk_3) \quad (\text{B.41})$$

Whereby  $h$  is the time step:

$$t_{i+1} = t_i + h \quad (\text{B.42})$$

APPENDIX B

# Bibliography

- [1] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of Honda humanoid robot. In *IEEE International Conference on Robotics and Automation (ICRA 1998)*, volume 2, pages 1321–1326, May 1998.
- [2] Christian Ott, Oliver Eiberger, Werner Friedl, B. Bauml, Ulrich Hillenbrand, Christoph Borst, A. Albu-Schaffer, Bernhard Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M Suppa, T. Wimbrock, F. Zacharias, and G. Hirzinger. A humanoid two-arm system for dexterous manipulation. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 276–283. IEEE, 2006.
- [3] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. Mechatronic design of nao humanoid. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 769–774. IEEE, 2009.
- [4] Kerstin Dautenhahn and Chrystopher L Nehaniv. The correspondence problem. In *Imitation in Animals and Artifacts*, MIT Press. MIT Press, 2002.
- [5] Aris Alissandrakis, Chrystopher L. Nehaniv, and Kerstin Dautenhahn. Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(4):482–496, 2002.
- [6] Marjorie Fink Vargas. *Louder than words: An introduction to nonverbal communication*. Iowa State Pr, 1986.
- [7] E.T. Hall. *The dance of life*. Doubleday New York, 1989.
- [8] Julius Fast. *Body language*. Simon and Schuster, 1988.
- [9] David Efron. Gesture and environment. 1941.

## BIBLIOGRAPHY

- [10] P. Ekman. Emotional and conversational nonverbal signals. *Philosophical Studies Series*, 99:39–50, 2003.
- [11] Paul Ekman and Wallace V. Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *semiotica*, 1(1):49–98, 1969.
- [12] Mark L. Knapp, Judith A. Hall, and Terrence G. Horgan. *Nonverbal communication in human interaction*. Cengage Learning, 2013.
- [13] Adam Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.
- [14] Adam Kendon. Language and gesture: Unity or duality. *Language and gesture*, (2), 2000.
- [15] David McNeill. So you think gestures are nonverbal? *Psychological review*, 92(3):350, 1985.
- [16] D. McNeill. *Hand and mind: What gestures reveal about thought*. University of Chicago Press, 1996.
- [17] Cynthia Breazeal, Atsuo Takanishi, and Tetsunori Kobayashi. Social robots that interact with people. In *Springer handbook of robotics*, pages 1349–1369. Springer, 2008.
- [18] Omar Mubin, Catherine J Stevens, Suleman Shahid, Abdullah Al Mahmud, and Jian-Jie Dong. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1:209–0015, 2013.
- [19] Martin Saerbeck, Tom Schut, Christoph Bartneck, and Maddy D Janse. Expressive robots in education: varying the degree of social supportive behavior of a robotic tutor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1613–1622. ACM, 2010.
- [20] Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. Receptionist or information kiosk: How do people talk with a robot? In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 31–40. ACM, 2010.
- [21] Masahiro Shiomi, Daisuke Sakamoto, Takayuki Kanda, Carlos Toshinori Ishi, Hiroshi Ishiguro, and Norihiro Hagita. Field trial of a networked robot at a train station. *International Journal of Social Robotics*, 3(1):27–40, 2011.

- [22] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. An affective guide robot in a shopping mall. In *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pages 173–180. IEEE, 2009.
- [23] Shirley Elprama, Ilias El Makrini, Bram Vanderborght, and An Jacobs. Acceptance of collaborative robots by factory workers: a pilot study on the role of social cues of anthropomorphic robots. In *The 25th IEEE International Symposium on Robot and Human Interactive Communication*, pages 919–919, 2016.
- [24] Guy Hoffman and Cynthia Breazeal. Collaboration in human-robot teams. In *AIAA 1st Intelligent Systems Technical Conference*, page 6434, 2004.
- [25] Astrid Weiss, Daniela Wurhofer, Michael Lankes, and Manfred Tschelegi. Autonomous vs. tele-operated: How people perceive human-robot collaboration with hrp-2. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 257–258. ACM, 2009.
- [26] Sarah M. Rabbitt, Alan E. Kazdin, and Brian Scassellati. Integrating socially assistive robotics into mental healthcare interventions: Applications and recommendations for expanded use. *Clinical psychology review*, 35:35–46, 2015.
- [27] Toshiyo Tamura, Satomi Yonemitsu, Akiko Itoh, Daisuke Oikawa, Akiko Kawakami, Yuji Higashi, Toshiro Fujimoto, and Kazuki Nakajima. Is an entertainment robot useful in the care of elderly people with severe dementia? *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 59(1):M83–M85, 2004.
- [28] Kazuyoshi Wada, Takanori Shibata, Toshimitsu Musha, and Shin Kimura. Effects of robot therapy for demented patients evaluated by eeg. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1552–1557. IEEE, 2005.
- [29] Ramona Simut, Johan Vanderfaeillie, Andreea Peca, Greet Van de Perre, and Bram Vanderborght. Children with Autism Spectrum Disorders make a fruit salad with Probo, the social robot: an interaction study. *Journal of autism and developmental disorders*, 46(1):113–126, 2016.
- [30] Ramona Simut, Cristina A Costescu, Johan Vanderfaeillie, Greet van de Perre, Bram Vanderborght, Dirk Lefever, et al. Can you cure me? children with autism spectrum disorders playing a doctor game with a social robot. *International Journal of School Health*, 3(3), 2016.

## BIBLIOGRAPHY

- [31] Ramona Simut, Greet Van de Perre, Cristina Costescu, Jelle Saldien, Johan Vanderfaillie, Daniel David, Dirk Lebefer, and Bram Vanderborght. Probogotchi: a novel edutainment device as a bridge for interaction between a child with asd and the typically developed sibling. *Journal of Evidence-Based Psychotherapies*, 16(1):91, 2016.
- [32] Hoang-Long Cao, Cristina Pop, Ramona Simut, Raphaël Furnémont, Albert De Beir, Greet Van de Perre, Pablo Gómez Esteban, Dirk Lefeber, and Bram Vanderborght. Probolino: a portable low-cost social device for home-based autism therapy. In *International Conference on Social Robotics*, pages 93–102. Springer, 2015.
- [33] Hoang-Long Cao, Greet Van de Perre, Ramona Simut, Cristina Pop, Andreea Peca, Dirk Lefeber, and Bram Vanderborght. Enhancing my keepon robot: A simple and low-cost solution for robot platform in human-robot interaction studies. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 555–560. IEEE, 2014.
- [34] Pablo G. Esteban, Paul Baxter, Tony Belpaeme, Erik Billing, Haibin Cai, Hoang-Long Cao, Mark Coeckelbergh, Cristina Costescu, Daniel David, Albert De Beir, et al. How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. *Paladyn, Journal of Behavioral Robotics*, 8(1):18–38, 2017.
- [35] Hoang-Long Cao, Pablo Gómez Esteban, Ramona Simut, Greet Van de Perre, Dirk Lefeber, Bram Vanderborght, et al. A collaborative homeostatic-based behavior controller for social robots in human–robot interaction experiments. *International Journal of Social Robotics*, pages 1–16.
- [36] Hoang-Long Cao, Greet Van de Perre, James Kennedy, Emmanuel Senft, Pablo Gómez Esteban, Albert De Beir, Ramona Simut, Tony Belpaeme, Dirk Lefeber, and Bram Vanderborght. A personalized and platform-independent behavior control system for social robots in therapy: development and applications. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.
- [37] Eunil Park, Ki Joon Kim, and Angel P. del Pobil. The effects of robot body gesture and gender in human-robot interactionnn. *Human-Computer Interaction*, 6:91–96, 2011.
- [38] Maha Salem, Friederike Eyssel, Katharina Rohlfing, Stefan Kopp, and Frank Joublin. To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. *Int. Journal of Social Robotics*, pages pp. 1–11, 2013.

- [39] Cynthia Breazeal, Cory D. Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 708–713, 2005.
- [40] Minhua Zheng and Max Q-H Meng. Designing gestures with semantic meanings for humanoid robot. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 287–292. IEEE, 2012.
- [41] Crystal Chao and Andrea L. Thomaz. Controlling social dynamics with a parametrized model of floor regulation. 2013.
- [42] Masao Yokoyama, Kazumi Aoyama, Hideaki Kikuchi, and Katsuhiko Shirai. Use of non-verbal information in communication between human and robot. In *ICSLP*, 1998.
- [43] Marianne Gullberg. Gestures in spatial descriptions. *Working papers - Lund University, Department of Linguistics*, pages 87–98, 1999.
- [44] Gary L. Allen. Gestures accompanying verbal route directions: Do they point to a new avenue for examining spatial representations? *Spatial cognition and computation*, 3(4):259–268, 2003.
- [45] Michael Scaife and Jerome S Bruner. The capacity for joint visual attention in the infant. *Nature*, 1975.
- [46] Chris Moore and Phil Dunham. *Joint attention: Its origins and role in development*. Psychology Press, 2014.
- [47] Aaron St Clair, Ross Mead, and Maja J Matarić. Investigating the effects of visual saliency on deictic gesture production by a humanoid robot. In *RO-MAN, 2011 IEEE*, pages 210–216. IEEE, 2011.
- [48] Allison Sauppé and Bilge Mutlu. Robot deictics: How gesture and context shape referential communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 342–349. ACM, 2014.
- [49] Yusuke Okuno, Takayuki Kanda, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. Providing route directions: design of robot’s utterance, gesture, and timing. In *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pages 53–60. IEEE, 2009.
- [50] Lola Cañamero and Jakob Fredslund. I show you how i like you-can you read it in my face? *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 31(5):454–459, 2001.

## BIBLIOGRAPHY

- [51] Tetsuya Ogata and Shigeki Sugano. Emotional communication robot: Wamoeba-2r emotion model and evaluation experiments. In *Proceedings of the International Conference on Humanoid Robots*. Citeseer, 2000.
- [52] Aryel Beck, Lola Cañamero, and Kim A Bard. Towards an affect space for robots to display emotional body language. In *RO-MAN*, pages 464–469. IEEE, 2010.
- [53] Tiago Ribeiro and Ana Paiva. The illusion of robotic life: principles and practices of animation for robots. In *Human-Robot Interaction (HRI)*, pages 383–390. IEEE, 2012.
- [54] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.
- [55] Cynthia Breazeal, Andrew Brooks, David Chilongo, Jesse Gray, Guy Hoffman, Cory Kidd, Hans Lee, Jeff Lieberman, and Andrea Lockerd. Working collaboratively with humanoid robots. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 1, pages 253–272. IEEE, 2004.
- [56] Cynthia Breazeal. Toward sociable robots. *Robotics and Autonomous Systems*, 42(3-4):167–175, 2003.
- [57] Mental models and cooperation with robotic assistants. Mental models and cooperation with robotic assistants. In *Proceedings of CHI*, 2002.
- [58] Cynthia Breazeal. Emotion and sociable humanoid robots. *International Journal of Human computer Studies*, 59(1-2):119–155, July 2003.
- [59] Jelle Saldien, Kristof Goris, Bram Vanderborght, Johan Vanderfaeilli, and Dirk Lefeber. Expressing Emotions with the Huggable Robot Probo. *International Journal of Social Robotics, Special Issue on Social Acceptance in HRI*, 2(4):377–389, 2010.
- [60] Jochen Hirth, Norbert Schmitz, and Karsten Berns. Emotional Architecture for the Humanoid Robot Head ROMAN. *Robotics and Automation, 2007 IEEE International Conference on*, pages 2150–2155, 2007.
- [61] K. Itoh, H. Miwa, M. Matsumoto, M. Zecca, H. Takanobu, S. Roccella, M.C. Carrozza, P. Dario, and A. Takanishi. Various emotional expressions with emotion expression humanoid robot WE-4RII. In *IEEE Technical Exhibition Based Conference on Robotics and Automation*, pages 35–36, November 2004.

- [62] Beatrice De Gelder. Why bodies? Twelve reasons for including bodily expressions in affective neuroscience. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1535):3475–3484, 2009.
- [63] Ariel Beck, Brett Stevens, and Kim Bard. Comparing perception of affective body movements displayed by actors and animated characters. In *Proceedings of the Symposium on Mental States, Emotions, and their Embodiment*, pages 169–178, 2009.
- [64] Massimiliano Zecca, Yu Mizoguchi, Keita Endo, Fumiya Iida, Yousuke Kawabata, Nobutsuna Endo, Kazuko Itoh, and Atsuo Takanishi. Whole body emotion expressions for kobian humanoid robot: preliminary experiments with different emotional patterns. In *The 18th IEEE Int. Symp. on Robot and Human Interactive Communication. RO-MAN 2009*, pages 381–386, 2009.
- [65] Hiroyasu Miwa, Kazuko Itoh, Munemichi Matsumoto, Massimiliano Zecca, Hideaki Takanobu, S Rocella, Maria Chiara Carrozza, Paolo Dario, and Atsuo Takanishi. Effective emotional expressions with expression humanoid robot we-4rii: integration of humanoid robot hand rch-1. In *Intelligent Robots and Systems, 2004. Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2203–2208. IEEE, 2004.
- [66] Raymond H. Cuijpers and Marco A.M.H. Knops. Motions of robots matter! the social effects of idle and meaningful motions. In *International Conference on Social Robotics*, pages 174–183. Springer, 2015.
- [67] Pablo Gómez Esteban, Hoang-Long Cao, Albert De Beir, Greet Van de Perre, Dirk Lefeber, and Bram Vanderborght. Study on the acceptability of social reactive behaviours during human-robot interactions. *International Journal of Social Robotics*, 2017.
- [68] Osamu Sugiyama, Takayuki Kanda, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. Natural deictic communication with humanoid robots. In *IROS 2007*, pages 1441–1448, 2007.
- [69] Junichi Ido, Yoshio Matsumoto, Tsukasa Ogasawara, and Ryuichi Nisimura. Humanoid with interaction ability using vision and speech information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2006)*, pages 1316–1321, 2006.
- [70] Daisuke Matsui, Takashi Minato, Karl F. MacDorman, and Hiroshi Ishiguro. Generating natural motion in an android by mapping human motion. In *IROS*, pages 3301–3308, 2005.

## BIBLIOGRAPHY

- [71] Martin Do, Pedram Azad, Tamim Asfour, and Rüdiger Dillmann. Imitation of human motion on a humanoid robot using non-linear optimization. In *8th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids*, pages 545–552, 2008.
- [72] Adriana Tapus, Andreea Peca, Amir Aly, Cristina Pop, Lavinia Jisa, Sebastian Pintea, Alina S Rusu, and Daniel O David. Children with autism social engagement in interaction with nao, an imitative robot: A series of single case experiments. *Interaction studies*, 13(3):315–347, 2012.
- [73] Etienne Balit, Dominique Vaufreydaz, and Patrick Reignier. Integrating animation artists into the animation design of social robots. In *ACM/IEEE Human-Robot Interaction 2016*, 2016.
- [74] Pierre Andry, Philippe Gaussier, Sorin Moga, Jean-Paul Banquet, and Jacqueline Nadel. Learning and communication via imitation: An autonomous robot perspective. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(5):431–442, 2001.
- [75] Christopher Stanton, Anton Bogdanovych, and Edward Ratanasena. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In *Proc. Australasian Conference on Robotics and Automation*, 2012.
- [76] Manuel Mühlig, Michael Gienger, and Jochen J Steil. Interactive imitation learning of object movement skills. *Autonomous Robots*, 32(2):97–114, 2012.
- [77] Pedram Azad, Tamim Asfour, and Rüdiger Dillmann. Toward an unified representation for imitation of human motion on humanoids. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2558–2563. IEEE, 2007.
- [78] Orner Terlemez, Stefan Ulbrich, Christian Mandery, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. Master motor map (mmm) – framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 894–901. IEEE, 2014.
- [79] Yoshihito Koga, Koichi Kondo, James Kuffner, and Jean-Claude Latombe. Planning motions with intentions. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 395–408. ACM, 1994.

- [80] John F. Soechting and Martha Flanders. Errors in pointing are due to approximations in sensorimotor transformations. *Journal of Neurophysiology*, 62(2):595–608, 1989.
- [81] Tiago Ribeiro and Ana Paiva. Animating the Adelino robot with ERIK: the expressive robotics inverse kinematics. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017.
- [82] Maha Salem, Stefan Kopp, Ipke Wachsmuth, and Frank Joublin. Generating multi-modal robot behavior based on a virtual agent framework. In *Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR)*, 2010.
- [83] Quoc Anh Le, Souheil Hanoune, and Catherine Pelachaud. Design and implementation of an expressive gesture model for a humanoid robot. In *11th IEEE-RAS International Conference on Humanoid Robots*, pages 134–140. IEEE, 2011.
- [84] Michael Gienger, Herbert Janssen, and Christian Goerick. Task-oriented whole body motion for humanoid robots. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 238–244. IEEE, 2005.
- [85] Maja J. Matarić and Marc Pomplun. Fixation behavior in observation and imitation of human movement. *Cognitive Brain Research*, 7(2):191–202, 1998.
- [86] Anthony P Atkinson, Winand H Dittrich, Andrew J Gemmell, Andrew W Young, et al. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception London*, 33:717–746, 2004.
- [87] Tony Belpaeme, Paul E Baxter, Robin Read, Rachel Wood, Heriberto Cuayáhuitl, Bernd Kiefer, Stefania Racioppa, Ivana Kruijff-Korbayová, Georgios Athanasopoulos, Valentin Enescu, et al. Multimodal child-robot interaction: Building social bonds. *Journal of Human-Robot Interaction*, 1(2):33–53, 2012.
- [88] Maha Salem, Stefan Kopp, Ipke Wachsmuth, and Frank Joublin. Towards meaningful robot gesture. In *Human Centered Robot Systems*, pages 173–182. Springer, 2009.
- [89] Manfred Hild, Torsten Siedel, Christian Benckendorff, Christian Thiele, and Michael Spranger. Myon, a new humanoid. In *Language Grounding in Robots*, pages 25–44. Springer, 2012.

## BIBLIOGRAPHY

- [90] Hirohisa Hirukawaa, Fumio Kanehiroa, Kenji Kanekoa, Shuji Kajitaa, Kiyoshi Fujiwaraa, Yoshihiro Kawaia, Fumiaki Tomitaa, Shigeoki Hiraiia, Kazuo Taniea, Takakatsu Isozumib, Kazuhiko Akachib, Toshikazu Kawasakib, Shigehiko Otab, Kazuhiko Yokoyamac, Hiroyuki Handac, Yutaro Fukased, Jun ichiro Maedad, Yoshihiko Nakamurae, Susumu Tachie, and Hirochika Inoue. Humanoid robotics platforms developed in HRP. *Robotics and Autonomous Systems*, 48(4):165–175, October 2004.
- [91] P. Ekman and W.V. Friesen. *Facial Action Coding System*. Consulting Psychologists Press, 1978.
- [92] Kyung Won Chung and Harold M Chung. *Gross anatomy*. Lippincott Williams & Wilkins, 2008.
- [93] Richard . Snell. *Clinical anatomy by systems*. Lippincott Williams & Wilkins, 2007.
- [94] Lynn S. Lippert. *Clinical kinesiology and anatomy*. FA Davis, 2011.
- [95] Andrea Kleinsmith, Ravindra De Silva, and Nadia Bianchi-Berthouze. Cross-cultural differences in recognizing affect from body posture. *Interacting with Computers*, 18(6):1371–1389, 2006.
- [96] Victor Ng-Thow-Hing, Pengcheng Luo, and Sandra Okita. Synchronized gesture and speech production for humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2010*, pages 4617–4624, 2010.
- [97] Frank Thomas, Ollie Johnston, and Walton Rawls. *Disney animation: The illusion of life*. Abbeville Press New York, 1981.
- [98] Jelle Saldien, Bram Vanderborght, Kristof Goris, Michael Van Damme, and Dirk Lefeber. A motion system for social and animated robots. *International Journal of Advanced Robotic Systems*, 11(5):72, 2014.
- [99] M. P. Kadaba, H.K. Ramakrishnan, and M.E. Wootten. Measurement of lower extremity kinematics during level walking. *Journal of Orthopaedic Research*, 8(3):383–392, 1990.
- [100] H. Kozima, M. Michalowski, and C. Nakagawa. Keepon - a playful robot for research, therapy, and entertainment. *International Journal of Social Robotics*, 1:3–18, 2008.
- [101] Lorenzo Sciavicco. *Robotics: modelling, planning and control*. Springer, 2009.

- [102] Uri M. Ascher and Linda R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.
- [103] Nikolaos G. Tsagarakis, Giorgio Metta, Giulio Sandini, David Vernon, Ricardo Beira, Francesco Becchi, Ludovic Righetti, Jose Santos-Victor, Auke Jan Ijspeert, Maria Chiara Carrozza, and D.G. Caldwell. iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21(10):1151–1175, 2007.
- [104] Masato Hirose and Toru Takenaka. Development of the humanoid robot ASIMO. *Honda R&D Technical Review*, 13(1):1–6, 2001.
- [105] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The NAO humanoid: a combination of performance and affordability. *Arxiv preprint arXiv:0807.3223*, 2008.
- [106] Fumihide Tanaka, Kuniaki Noda, Tsutomu Sawada, and Masahiro Fujita. Associated emotion and its expression in an entertainment robot QRIO. *Entertainment Computing-ICEC 2004*, pages 265–298, 2004.
- [107] Rocketbox. Rocketbox libraries, 2013. [www.rocketbox-libraries.com](http://www.rocketbox-libraries.com).
- [108] Asimo - 3d model, 2010. <http://tf3dm.com/3d-model/asimo-95121.html>.
- [109] Cynthia Breazeal. *Designing Sociable Robots*. The MIT Press, 2004.
- [110] Armin Biess, Dario G. Liebermann, and Tamar Flash. A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *The Journal of Neuroscience*, 27(48):13045–13064, 2007.
- [111] Jing Zhao, Biyun Xie, and Chunyu Song. Generating human-like movements for robotic arms. *Mechanism and Machine Theory*, 81:107–128, 2014.
- [112] Timothy Marler, Jingzhou Yang, Jasbir S. Arora, and Karim Abdel-Malek. Study of bi-criterion upper body posture prediction using pareto optimal sets. In *IASTED International Conference on Modeling, Simulation, and Optimization, Oranjestad, Aruba, Canada*, 2005.
- [113] Yoji Uno, Mitsuo Kawato, and Rika Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics*, 61(2):89–101, 1989.

## BIBLIOGRAPHY

- [114] John M. Hollerbach and Ki C. Suh. Redundancy resolution of manipulators through torque optimization. *Robotics and Automation, IEEE Journal of*, 3(4):308–316, 1987.
- [115] Eui S. Jung, Dohyung Kee, and Min K. Chung. Upper body reach posture prediction for ergonomic evaluation models. *International Journal of Industrial Ergonomics*, 16(2):95–107, 1995.
- [116] Charles A. Klein and Bruce E. Blaho. Dexterity measures for the design and control of kinematically redundant manipulators. *The International Journal of Robotics Research*, 6(2):72–83, 1987.
- [117] Holk Cruse, Michael Brüwer, and Jeffrey Dean. Control of three-and four-joint arm movement: Strategies for a manipulator with redundant degrees of freedom. *Journal of motor behavior*, 25(3):131–139, 1993.
- [118] Nikolaj A Bernstein. The co-ordination and regulation of movements. 1967.
- [119] JA Scott Kelso, Dan Southard, and David Goodman. On the nature of human interlimb coordination. *Science*, 203(4384):1029–1031, 1979.
- [120] Pietro Morasso. Spatial control of arm movements. *Experimental brain research*, 42(2):223–227, 1981.
- [121] Christopher G Atkeson and John M Hollerbach. Kinematic features of unrestrained vertical arm movements. *The Journal of Neuroscience*, 5(9):2318–2330, 1985.
- [122] William Abend, Emilio Bizzi, and Pietro Morasso. Human arm trajectory formation. *Brain: a journal of neurology*, 105(Pt 2):331–348, 1982.
- [123] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics interface*, volume 96, pages 222–229. Toronto, Canada, 1996.
- [124] Catherine Pelachaud. Studies on gesture expressivity for a virtual agent. *Speech Communication*, 51(7):630–639, 2009.
- [125] Atsushi Yamaguchi, Yoshikazu Yano, Shinji Doki, and Shigeru Okuma. A study of emotional motion description by motion modification and adjectival expressions. In *IEEE Conference on Cybernetics and Intelligent Systems 2006*, pages 1–6, 2006.
- [126] Yueh-Hung Lin, Chia-Yang Liu, Hung-Wei Lee, Shwu-Lih Huang, and Tsai-Yen Li. Evaluating emotive character animations created with procedural animation. In *Intelligent Virtual Agents*, pages 308–315. Springer, 2009.

- [127] Junchao Xu, Joost Broekens, Koen Hindriks, and Mark A Neerincx. Mood expression through parameterized functional behavior of robots. In *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN 2013)*, pages 533–540, 2013.
- [128] Junchao Xu, Joost Broekens, Koen Hindriks, and Mark A Neerincx. The relative importance and interrelations between behavior parameters for robots' mood expression. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 558–563. IEEE, 2013.
- [129] Jonathan Posner, James A Russell, and Bradley Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17:715–734, 2005.
- [130] Alain Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics*, 7(12):868–871, 1977.
- [131] Dániel András Drexler and István Harmati. Joint constrained differential inverse kinematics algorithm for serial manipulators. *Periodica Polytechnica. Electrical Engineering and Computer Science*, 56(4):95, 2012.
- [132] Aldebaran Robotics. Romeo documentation, 2014. [accessed February-2016].
- [133] Aldebaran Robotics. Pepper documentation, 2014. [accessed February-2016].
- [134] Albert Albers, Sven Brudniok, Jens Ottnad, Christian Sauter, and Korkiat Sedchaicharn. Upper body of a new humanoid robot-the design of armar iii. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 308–313. IEEE, 2006.
- [135] Yu Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, Hun ok Lim, and A. Takanishi. Development of a new humanoid robot WABIAN-2. In *IEEE International Conference on Robotics and Automation (ICRA 2006)*, pages 76–81, May 2006.
- [136] Head Robot Team/Takanshi Laboratory. Emotion expression humanoid robot we-4rii, 2006.
- [137] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu. Robovie: an interactive humanoid robot. *Industrial robot: An international journal*, 28(6):498–504, 2001.

## BIBLIOGRAPHY

- [138] Ill-Woo Park, Jung-Yup Kim, Jungho Lee, and Jun-Ho Oh. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 321–326. IEEE, 2005.
- [139] Alberto Parmiggiani, Alessandro Scalzo Luca Fiorio and, Anand Vazhapilli Sureshbabu, Marco Randazzo, Marco Maggiali, Ugo Pattacini, Hagen Lehmann, Vadim Tikhanoff, Daniele Domenichelli, Alberto Cardellino, Pierpaolo Congiu, Andrea Pagnin, Roberto Cingolani, Lorenzo Natale, and Giorgio Metta. The design and validation of the r1 personal humanoid. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [140] Tamim Asfour, Julian Schill, Heiner Peters, Cornelius Klas, Jens Bücker, Christian Sander, Stefan Schulz, Artem Kargov, Timo Werner, and Volker Bartenbach. Armar-4: A 63 dof torque controlled humanoid robot. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 390–396. IEEE, 2013.
- [141] Olivier Stasse, Thomas Flayols, Rohan Budhiraja, Kevin Giraud-Escalas, Justin Carpentier, Andrea Del Prete, Philippe Soueres, Nicolas Mansard, Florent Lamiriaux, Jean-Paul Laumond, L. Marchionni, H. Tome, and F. Ferro. Talos: A new humanoid research platform targeted for industrial applications. 2017.
- [142] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, M Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.
- [143] Bram Vanderborght, Nikos G Tsagarakis, Claudio Semini, Ronald Van Ham, and Darwin G Caldwell. Maccepa 2.0: Adjustable compliant actuator with stiffening characteristic for energy efficient hopping. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 544–549. IEEE, 2009.
- [144] Björn Verrelst, Ronald Van Ham, Bram Vanderborght, Dirk Lefeber, Frank Daerden, and Michael Van Damme. Second generation pleated pneumatic artificial muscle and its robotic applications. *Advanced Robotics*, 20(7):783–805, 2006.
- [145] Kristof Goris, Jelle Saldien, Bram Vanderborght, and Dirk Lefeber. Mechanical design of the huggable robot Probo. *International Journal of Humanoid Robotics*, 8(3):481–511, 2011.
- [146] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer-Verlag New York Inc, 2008.

- [147] Sooyeon Jeong, Kristopher Dos Santos, Suzanne Graca, Brianna O'Connell, Laurel Anderson, Nicole Stenquist, Katie Fitzpatrick, Honey Goodenough, Deirdre Logan, Peter Weinstock, et al. Designing a socially assistive robot for pediatric care. In *Proceedings of the 14th international conference on interaction design and children*, pages 387–390. ACM, 2015.
- [148] Kristof Goris, Jelle Saldien, Bram Vanderborght, and Dirk Lefeber. How to achieve the huggable behavior of the social robot Probo? A reflection on the actuators. *Mechatronics*, 21(3):490–500, April 2011.
- [149] Bram Vanderborght, Ramona Simut, Jelle Saldien, Cristina Pop, Alina Rusu, Sebastian Pintea, Dirk Lefeber, and Daniel David. Using the social robot probo as a social story telling agent for children with ASD. *Interaction Studies*, 13(3):348–372, 2012.
- [150] Ramona Simut, Greet Van de Perre, Cristina Pop, BramVanderborght, Jelle Saldien, Alina Rusu, Sebastian Pintea, Johan Vanderfaeillie, Dirk Lefeber, and Daniel David. The Huggable Social Robot Probo for Social Story Telling for Robot Assisted Therapy with ASD Children. In *International Conference on Social Robotics, Works-In-Progress*, 2011.
- [151] Cristina Anamaria Pop, Ramona Simut, Sebastian Pintea, Jelle Saldien, Alina Rusu, Daniel David, Johan Vanderfaeillie, Dirk Lefeber, and Bram Vanderborght. Can the social robot Probo help children with autism to identify situation-based emotions? A series of single case experiments. *International Journal of Humanoid Robotics*, 10(03):1350025, 2013.
- [152] Ernst Hairer, Christian Lubich, and Michel Roche. *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, volume 1409. Springer, 2006.

## BIBLIOGRAPHY

