Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Grant Agreement Type: Collaborative Project
Grant Agreement Number: 611391

D6.4 Expression and Actuation Subsystem

Due date: **1/4/2019**
Submission Date: **29/3/2019**

Start date of project: **01/04/2014**   Duration: **54 months**

Organisation name of lead contractor for this deliverable: **Vrije Universiteit Brussel**

Responsible Person: **Bram Vanderborght**   Revision: **5.0**

Contents

Executive Summary

Deliverable D6.4 defines the specification, design, implementation and validation of the Expression and Actuation subsystem within the cognitive architecture in Work Package 6.

More specifically, this report presents the advances done in task T6.4. During the first year a generic method to generated upper body emotional expressions was designed and the Actuation and Expression subsystem was designed. During the second year the focus was on implementing robot low-level motor control and extending the generic method with reaching a pointing gestures. In the third year, the design of the actuation and expression subsystem was improved and implemented to enable the delivery of the core robot behaviour for use in interventions. During the fourth year, the complexity of the actuation and expression subsystem was reduced and its performance significantly improved. Finally, during the fifth year the generic method behind the actuation and expression subsystem was revised to study the influence of different design aspects, ending up in the design of a new social robot. The subsystem is fully deployed and running with the cognitive architecture developed by WP6 for use in interventions.

Principal Contributors

The main authors of this deliverable are as follows (in alphabetical order).

Tony Belpaeme, Plymouth University
Erik Billing, University of Skövde
Hoang-Long Cao, Vrije Universiteit Brussel
Albert de Beir, Vrije Universiteit Brussel
Pablo Gómez, Vrije Universiteit Brussel
Daniel Hernández García, Plymouth University
James Kennedy, Plymouth University
Emmanuel Senft, Plymouth University
Greet Van de Perre, Vrije Universiteit Brussel
Bram Vanderborght, Vrije Universiteit Brussel

Revision History

Version 1.0 (P.G. 27-01-2015)
First draft.

Version 2.0 (P.G. 26-02-2015)
Some texts have been corrected.

Version 2.1 (P.G. 23-03-2015)
Appendices added.

Version 2.2 (P.B. 25-03-2015)
Addition of details of preliminary version of Expression and Actuation subsystem component.

Version 2.3 (P.G. 02-03-2016)
Addition of details related to the second year of the project. Included Period 2 annex and papers.

Version 3.0 (P.G. 31-01-2017)
Addition of related information with the third year and inclusion of Period 3 annex and papers.

Version 4.0 (P.G. 20-03-2018)
Addition of related information with the fourth year of the project. Included Period 4 annex and papers.

Version 4.1 (P.G. 5-04-2018)
Some texts have been modified

Version 5.0 (P.G. 18-03-2019)
Addition of related information with the fifth year of the project. Included Period 5 annex and papers.

1 Overview of WP6 architecture

In DREAM we are moving away from Wizard of Oz-controlled (WoZ) behaviour for the robot, which too often is the de facto mode of interaction in Robot Assisted Therapy [1]. Therefore, work package WP6 aims to progress the theoretical and methodological understanding of how an embodied system can interact autonomously with young users in a learning task, specifically developed for atypically developing children. WP6 is concerned with the development of the robot behaviour subsystems to provide social robots with a behaviour underlying social interaction, which permits the robot to be used in Robot Enhanced Therapy (RET) autonomously with supervision. This involves both autonomous behaviour and behaviour created in supervised autonomy, whereby an operator requests certain interventions, which are then autonomously executed by the robot.

A general high level description of the robot control system is shown in figure 1. This describes how the autonomous controller is informed by three external sources: the child behaviour description, sensory information, and current intervention script state. Input from a therapist, e.g., emergency stop, is also present, but not shown in the diagram. Combining these sources, the autonomous controller should trigger an appropriate sequence of action primitives to be performed (as well as some feedback via a graphical user interface), which then gets executed on the robot.
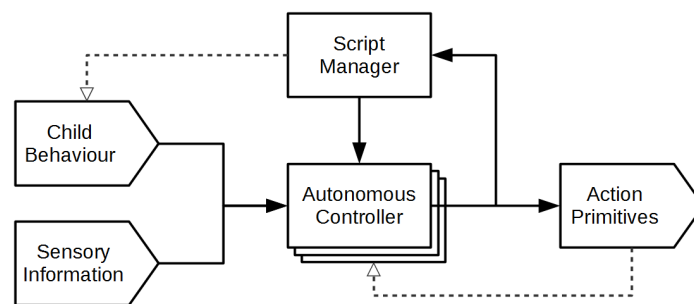

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied. The intervention script provides context for child behaviour interpretation.

The autonomous controller is composed of a number of subsystems, as described in the DoW: reactive, attention, deliberative, self-monitor and expression and actuation. In the reactive subsystem, sensory inputs are immediately acted upon with appropriate actuator outputs. The attention subsystem determines the robot's focus of attention. In the deliberative subsystem, the necessary interventions are implemented in a general approach so it is not scenario-specific. The self-monitoring subsystem acts as an alarm system in two specifications. An internal one when the robot detects that it cannot act because of a technical limitation or an ethical issue. An external alarm is one where the therapist overrules the robot behaviour selection. Finally, the expression and actuation subsystem is responsible for generating believable human/animal-like smooth and natural motions and sounds that are platform independent. These subsystems interact, and must combine their suggested courses of actions to produce a coherent robot behaviour, in the context of constraints laid down by the therapist (for example, the script to be followed, types of behaviour not permissible for this particular child because of individual sensitivities, etc). As a result, we have formulated the following architecture describing how cognitive control informed by the therapy scripts is to be achieved (figure 2). This design is an iterative

improvement on earlier plans laid out in Annex 3.5; the fundamental principles remain the same, but the design required modification to ensure logical information flow when specified to a lower level.

A detailed description of the cognitive architecture was provided in deliverable D6.1 at month 18. Within this report we describe the functionality of the expression and actuation subsystem as well as any other additional modules required to execute the motor commands on the robot.
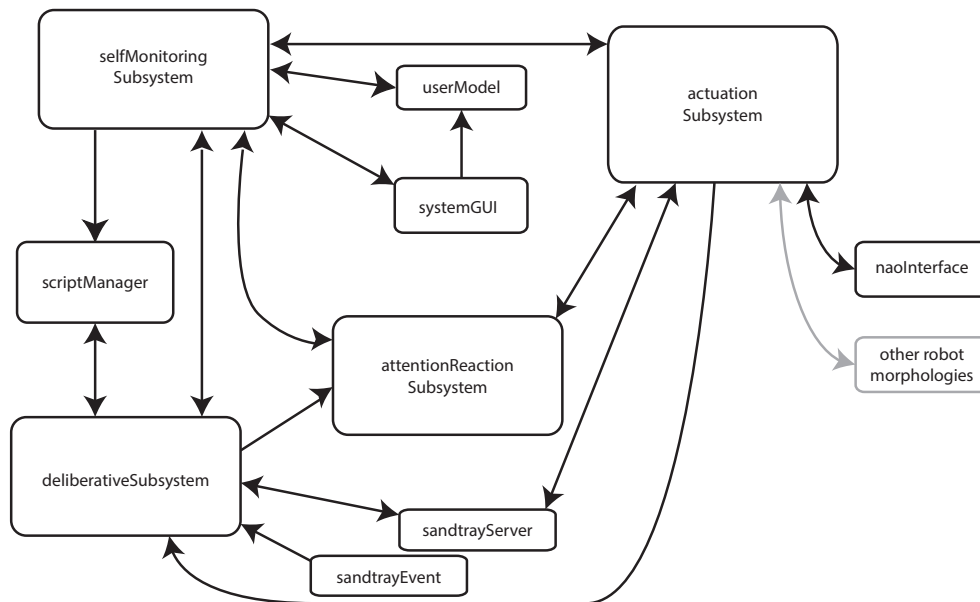
Figure 2: Diagram of the cognitive controller subsystem. The overall WP6 architecture decomposes into 10 components for delivery as part of the DREAM integrated system for use in therapeutic evaluations. This deliverable is concerned with the actuation aspects of the controller; this includes the following components: actuationSubsystem and naoInterface. The remaining components are discussed in other WP6 deliverables.

2 The expression and actuation subsystem

According to the DREAM project DoW, the goal of the Expression and Actuation subsystem is to translate the actions of the social behaviour into readable social verbal and non-verbal cues, especially for our particular audience of young users with ASD. Since the specification is that all internal descriptions of behaviour are robot-neutral, this subsystem also has to be platform independent.

2.1 State of the art

A number of robots capable of gesturing have been developed to study different aspects of gesturing in HRI. Gestures implemented in robots are however, up to now, subject to two important limitations. Firstly, the gestures implemented in a robot are always limited to a set of gestures necessary for the current research, and often limited to one type of gesture. The robot WE-4RII [2] for example, was developed to study human-like emotion, hence, the incorporated gestures are mainly focused on emotional expressions. On the other hand, the developers of Robovie aimed for communication robots

that interact naturally with humans. Since Robovie applications were focused on object indication and route direction-giving, mostly deictic gestures were used [3]. The reason for the limited amount of gestures implemented in specific robots can be found in the second limitation; namely the way gestures are implemented. Gestures are mostly preprogrammed off-line for the current robot configuration. The resulting postures are stored in a database and are replayed during interaction. This is the case for, among others, Robovie [4], HRP-2 [5] and Kobian [6]. Since the postures 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 is for example the case for Repliee Q2 [7], where a marker-based motion capture system is used. Another possibility is to use the Kinect to perform skeleton tracking [8]. In [9], both a marker-based (Vicon) and a markerless motion capture system was used to reproduce human motion for the robot ARMAR-IIIb. Since the mapping of the captured data is robot specific, also these resulting gestures are dependent on the morphology and not usable for other robots. The result is that, when working with a new robot platform, new joint trajectories to reach the desired postures need to be implemented, which can be time consuming. It would however be much more efficient to make the implementation of gestures more flexible and to design a general method that allows easy implementation of gestures on different robots, see figure 3.
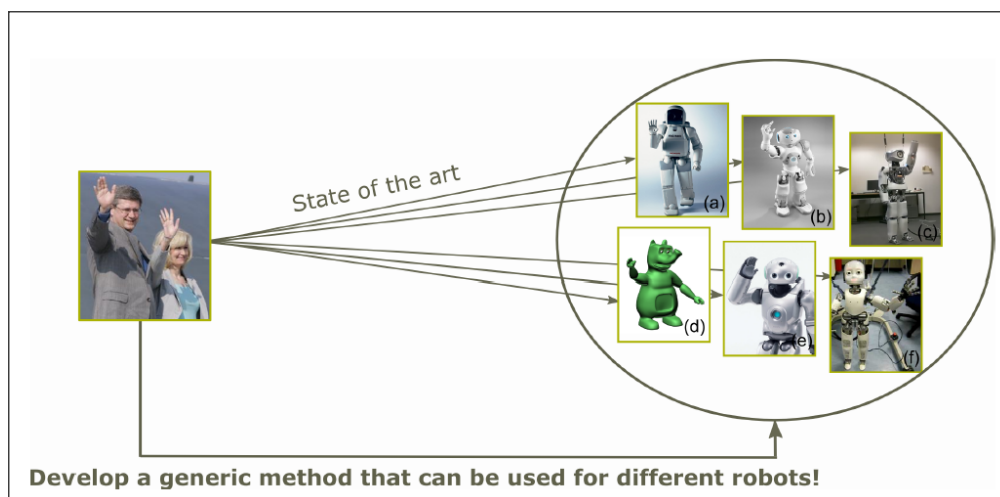
Figure 3: In the state of the art, gestures are always implemented for a specific robot platform. Our method aims to facilitate implementing gestures for a new robot platform by storing gestures independently of a morphology, and mapping them on a specific configuration. Robots: (a) ASIMO [10], (b) NAO [11], (c) Myon [12], (d) Probo [13], (e) QRIO [14], (f) iCub [15].

One of the approaches that flexibly generate gestures by different robots is based on neural networks, see [16]. However, this technique requires training. In both [10] and [17], a gesture framework initially developed for virtual agents is applied on a humanoid robot. In [10], the speech and gesture production model developed for the virtual agent Max is used to generate gestures for the ASIMO robot. Similarly to the ideas behind our subsystem, in [17], gestures are described independently of the embodiment by specifying features as the hand shape, wrist position and palm orientation. To generate gestures for the NAO robot, the correct angles for the shoulder and elbow joints are selected from a predetermined table listing all possible wrist positions and the corresponding joint values. The values for the remaining joints, namely the wrist joint and fingers are calculated by taking into con-

sideration the values of other features such as the hand shape and palm orientation. So although the gestures are described independently of the robot configuration, mapping these gestures to the robot requires hard coded joint information. Our desired method should aim to fully automate the mapping of gestures to a random robot configuration. Specifically for manipulation tasks, [18] presented a semi-general approach for the automatic generation of natural arm motions for human figures. In their inverse kinematics algorithm, which is based on neurophysiological findings, the problem of finding joint angles for the arm is decoupled from finding those from the wrist. The sensorimotor transformation model of [19] is used to determine the arm posture, while the wrist angles are found by assuming a spherical wrist and using orientation inverse kinematics.

Different robots use the Facial Action Coding System (FACS) by Ekman and Friesen [20] to abstract away from the physical implementation of the robot face. FACS decomposes different human facial expressions in the activation of a series of Action Units (AU), which are the contraction or relaxation of one or more muscles. In our work, we use FACS for the facial expressions and a similar framework for the rest of the body of the robot.

2.2   Research Objectives

The main functionality of this subsystem is to determine which combination of low-level actions the robot should execute next, and how these actions are to be performed. Suggestions for actions to take come from three other subsystems: deliberative, reactive/attention, and self-monitoring, see left side of figure 4. Along with this, it is assumed that the supervising therapist, through the GUI, determines (either beforehand or in real time) the aspects of robot behaviour that should be executed, from which relative priorities are determined for the three subsystems. This covers for example whether external disturbances (a loud noise in the background, or the appearance of a new face) should be reacted to by the robot (by leaving the script for a while for example), or ignored (with the script rigidly adhered to). The expression and actuation subsystem combines these sources of information in an appropriate manner, see motion mixer in figure 4, ensuring that the stability of the robot is maintained. For example, if a greeting wave is requested by the deliberative subsystem, and the reactive/attention subsystem wants to look at a face that has been detected, then the expression and actuation subsystem can combine the two by executing both (if the robot can remain stable by doing so). For a basic first step switches based on priority level could be used: i.e. if the script requests an action, execute it (and only it), but if there is no script action requested, then do what the reactive/attention subsystem proposes. However, the intention is to provide full behaviour mixing capabilities based on derived priorities from the therapists.

All this might be complemented by affective information, if this is available and appropriate to use. For example, the speed of motor execution could be related to arousal levels, or the choice of action sequence could be based on valence levels (if appropriate alternative sequences exist). This functionality needs to be switched on or off as required by the therapist based on child-specific considerations, and the relation to the therapy script (it may not appropriate to add emotional colouring to actions during the diagnosis procedure for example).

To approach such challenges, the first task was to design a platform-independent representation of expressions. As explained above we have based our work on the Facial Action Coding System by Ekman and Friesen [20]. In a similar way, Body Action Units (BAU) are defined together with a Body Action Coding System (BACS), where the different gestures are decomposed in the activation of BAUs. The BACS point out the Action Units that need to be actuated for the generation of a desired gesture or body pose. This system avoids pre-programming of robot-dependent body poses and actions, which is relevant since humans are able to recognize actions and emotions from point

light displays (so without body shape) [21]. The physical actuation of Action Units depends on the morphology of the robot: a mapping is needed between Action Units and physical actuators, this mapping is specific to a robot platform and requires us to explore the possibility of learning this mapping. To translate this to the morphology of the robot, the Action Units need to be mapped to the degrees of freedom (DOF), and thus to the joints of the robot, see right side of figure 4.

The second task was the categorisation of actions, comprised of temporal series of FACS and BACS, and the organisation in libraries that are accessible from the behaviour subsystems (reactive, attention and deliberative). All actions for the different behaviours should be stored and expanded upon without the need to reprogram other subsystems.
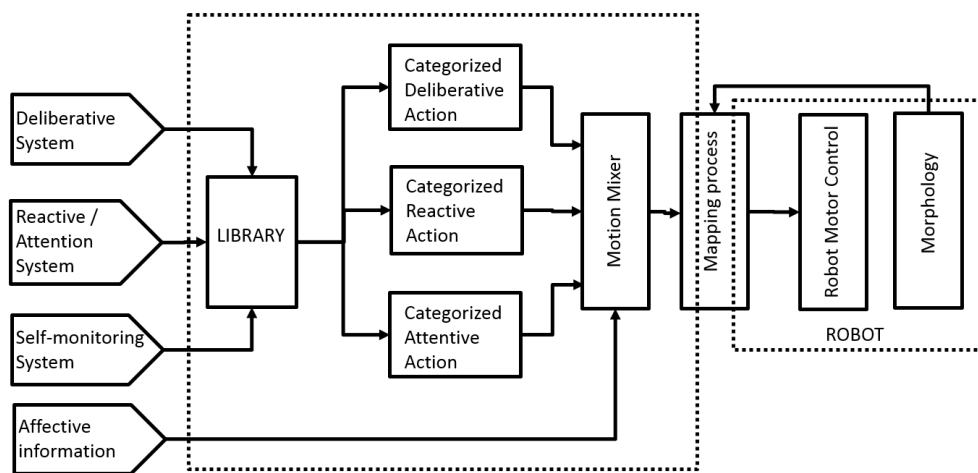

Figure 4: Overview of the expression and actuation subsystem. This subsystem receives inputs from several sources, categorizes them using the Library module and mixes them up to create a unique behaviour. Such behaviour is mapped into the joint configuration of the corresponding robot. This last process is done collaboratively between the subsystem and the robot.

2.3 Core research done

Our method divides the robot embodiment in three areas: the face expression, developed to provide the behaviours with natural and emotional features; the overall pose, developed to calculate gestures whereby the position of the main parts of the body is crucial, such as emotional expressions; and the end effector, developed for pointing and manipulation purposes.

We have implemented the FACS methodology in Probo to express emotions [22], see figure 5. The Action Units (AU) are used to define the motions for Probo's DOF. As Probo does not have a human face and for simplifying the design, some of the AU were missing, others were replaced and some were added. To make the robot capable of expressing emotions, a two-dimensional emotion space based on the circumplex model of affect by Russell [23] was used. In the emotion space a Cartesian coordinate system was used, where the x-coordinate represents the valence and the y-coordinate the arousal, consequently each emotion $e(v, a)$ corresponds to a point in the valence-arousal plane. Each emotion can be represented as a vector with the origin of the coordinate system as initial point and the corresponding valence-arousal values as the terminal point. The direction α of each vector defines the specific emotion whereas the magnitude defines the intensity of the emotion. Each DOF that

influences the facial expression was related to the angle of the emotion vector.
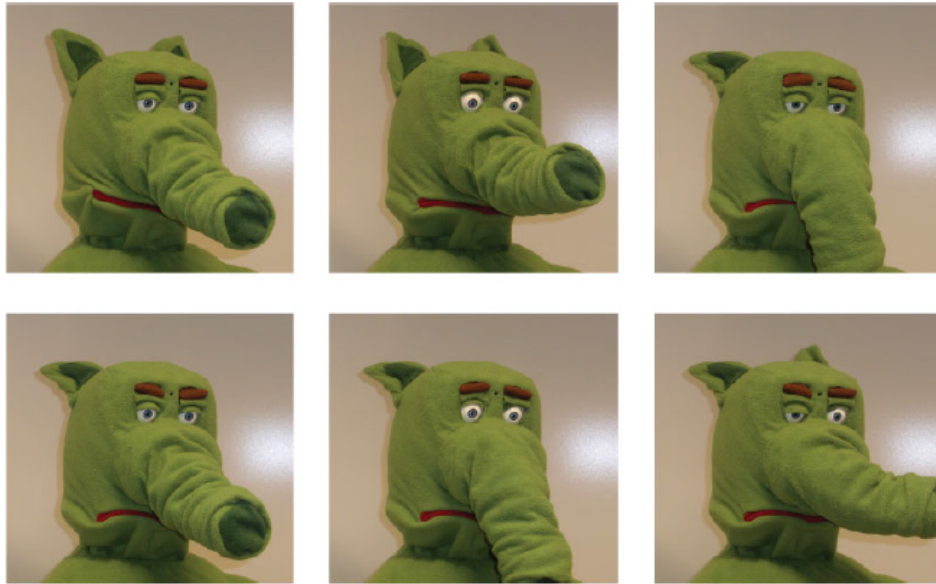
Figure 5: FACS has already been implemented in the huggable robot Probo to express different emotions.

The NAO robot has not got the facial expressibility that Probo has. It has no DOF in the face and the only mechanism that it has to express facial gestures is through the change of colors in its eyes. For this reason, an eyebrow system that will help to understand better emotional expressions on NAO's face has been developed, see figure 6 and annex 3.3 for further details.

Figure 6: Comparison of the NAO robot expressing anger and sadness with (right) and without (left) the eyebrow system.

To generate emotional expressions for a certain robot joint configuration, the developed method uses a set of target gestures listed in a database, which replaces the library of behaviours to be developed within the second task, and maps them to that specific configuration. To ensure a realistic and readable overall posture, it is necessary to take into account the rela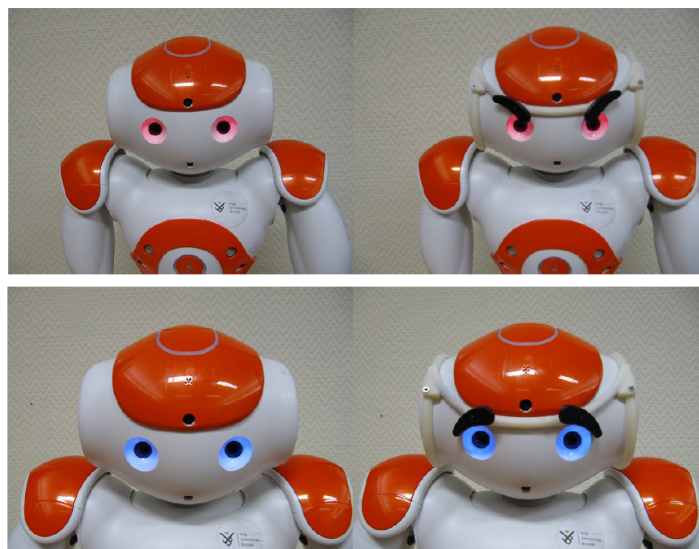tive orientations of every joint complex the robot has in common with a human. A base human model was defined, and the target postures were quantitatively described by the orientation of the different joint complexes in the model using a Body Action Coding System (BACS). This is similar to the Facial Action Coding System of Ekman and Friesen [20], in this case a set of Body Action Units (BAU's) is defined. While the Facial AU's are defined as a muscle or a muscle group, our BAU's are based on the human terms of motion. The units are grouped into different blocks, corresponding to one human joint complex, such as the shoulder or the wrist. These blocks can subsequently be grouped into three body parts, namely the head, body and arm, which we refer to as chains. In that way, a base human model was defined, consisting of four chains; the head, the body, the left arm and the right arm. Although the leg movements also contribute to the overall performance of the gesture, for a first validation of the method we decided to focus only on the upper body movements.

To make a certain model or robot perform a desired gesture or behaviour, this information should be mapped to its joint configuration. 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 by using an inverse kinematics algorithm. This step has been developed for the overall posture, which has been successfully validated on the virtual model of different robots through a survey, see figure 7. See annex 3.1 for further information.

To calculate pointing and manipulation gestures, another strategy is used. In some situations, for example when reaching for an object, the position of the end-effector is important and specified by the user. For pointing towards an object, several end-effector poses are possible to achieve a pointing gesture to the specified target. In that case, an optimal pose of the end-effector is chosen, according to a cost-function minimizing the deviation from a defined set of minimum posture angles. This specified end-effector pose then serves as input to calculated the corresponding joint angles, using the same inverse kinematics algorithm as used for the calculation of emotional expressions. Figure 8 shows the calculated end posture for a reaching gesture at $(34, -34, 38)$ for three 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, consisting of a two DOF clavicle, 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist (virtual model comes from the RocketBox Libraries [24]). Configuration 2 shows the ASIMO robot [25]. As for the human model, the targeted end-effector position was reachable, and a suitable end posture could be calculated, as shown in the second row. Configuration 3 is that of the NAO robot [26]. NAO is considerably smaller than the previous models, and as a result, the maximum 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. See annex 4.2 for further information.

During natural communication, humans use and combine different types of gestures. By combining the two modes of our method presented above, 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 motion mixer was designed, see figure 4. If the motion mixer is turned off, all gestures are treated separately; starting a new gestures entails a previously started gesture to be aborted. By enabling the motion 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
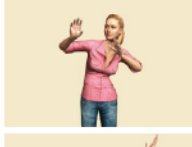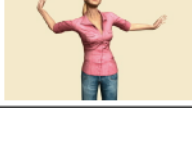
Figure 7: End postures of the gestures used in the survey. The first column shows the end posture of the target gestures for expressing the six basic emotions, while columns 2, 3 and 4 respectively show the mapped end postures for the robots ASIMO, Justin and NAO.

levels are defined using a number of rules. Firstly, for an emotional expression, the priority level for each chain is set on the basic value. For a deictic gesture, the priority level of the corresponding chain overrules the level of that chain for an emotional expression. In combination with a deictic gesture, the user can enable the gaze to be directed towards the point of interest. The necessary joint angles to reach the desired head orientation are calculated using the block mode. For the head block, gazing has a higher priority level than the calculation of the necessary joint angles for an emotional expression. For every separate chain, the priority levels determine which calculation principle has to be used for the current iteration; that of the block mode, or of the end-effector mode, and thus, which constraints are loaded for the different chains: orientational information for every block composing the chain, or the desired end-effector position for the complete chain. Therefore, when gestures with different priority levels are selected with the motion mixer enabled, the imposed end-effector conditions originating from the different gestures result in a blended end posture. See annex 5.1 for further details.

In some situations, it is desirable to express an emotional condition in a different manner than by
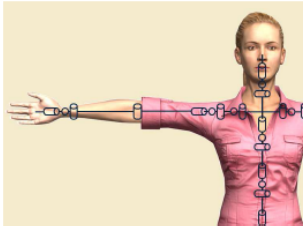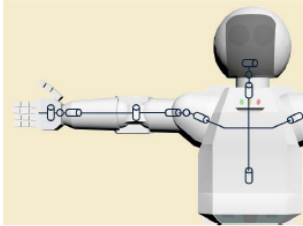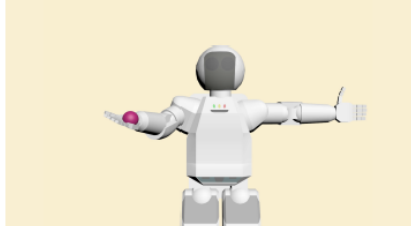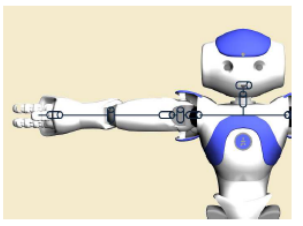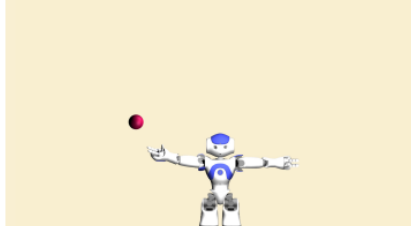
Configuration	Calculated posture
Config 1: 9 DOF arm	
Config 2: ASIMO	
Config 4: NAO	

Figure 8: 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 $(34, -34, 38)$.

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 line with [27], we then speak of mood expressions. In both [28] and [29], it was experimentally confirmed that the motion speed influences the perceived level of both valence and arousal; a fast motion is associated with a hight arousal and valence, while a slow motion is attributed to low arousal and valence values. By considering the two dimensional emotion space of valance and arousal, based on the circumplex model of affect [23], we obtained an appropriate speed scaling factor for each emotion (see figure 9). 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 second modification parameter, the amplitude of the motion, refers to the spatial extent; the amount of space occupied by the body. Xu et al. [28] 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 posters are related to states with

a low valence (see figures 10 and 11). See annex 5.1 for further details.
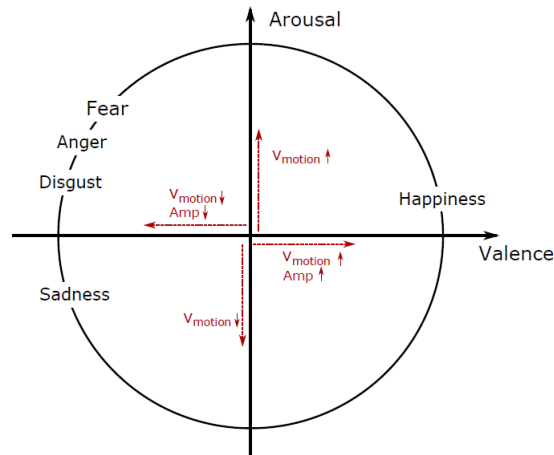
Figure 9: Dependency of the modification factors *motion speed* (v_{motion}) and *Amplitude* (Amp) on the valence and arousal value, depicted on the circumplex model of affect [23].
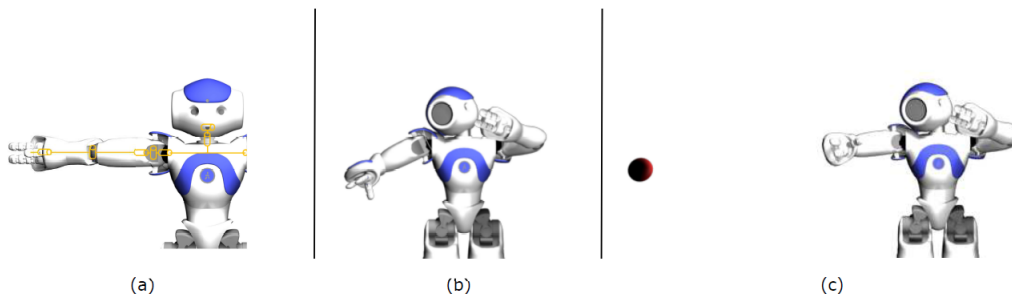
Figure 10: Example illustrating the calculation of a blended gesture for NAO. (a) Joint configuration of the robot. (b) Calculated end posture for the emotional expression of disgust. (c) Calculated end posture for a combination of a pointing gesture with the right arm, and the emotional expression of disgust.

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, among others, a highly actuated human model with 9 DOF arms, and the robots ASIMO [30] and Justin [31], both having 6 DOF arms, but however with considerably different morphology. For this validation on physical robots we worked with the robots Romeo [32], Pepper [33] and Nao [26]. All three robots have a different morphology. The specifications for Romeo are grouped in table 3. 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

(a) (b) (c)
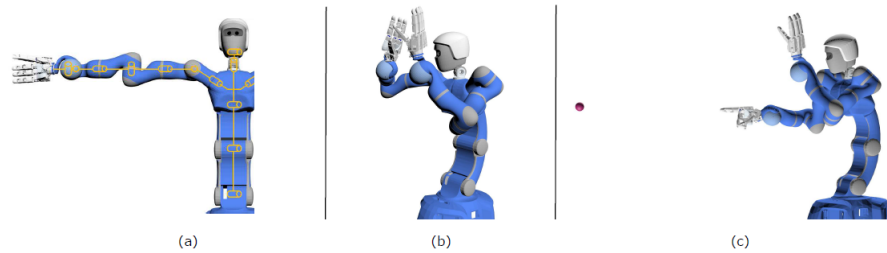
Figure 11: Example illustrating the calculation of a blended gesture for Justin. (a) Joint configuration of the robot. (b) Calculated end posture for the emotional expression of fear. (c) Calculated end posture for a combination of a pointing gesture with the right arm, accompanied by gazing towards the pointing location, and the emotional expression of disgust.

arm chain are grouped into the different blocks, which results in a 3 DOF shoulder and wrist, and a 1 DOF elbow block. Pepper 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. 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.
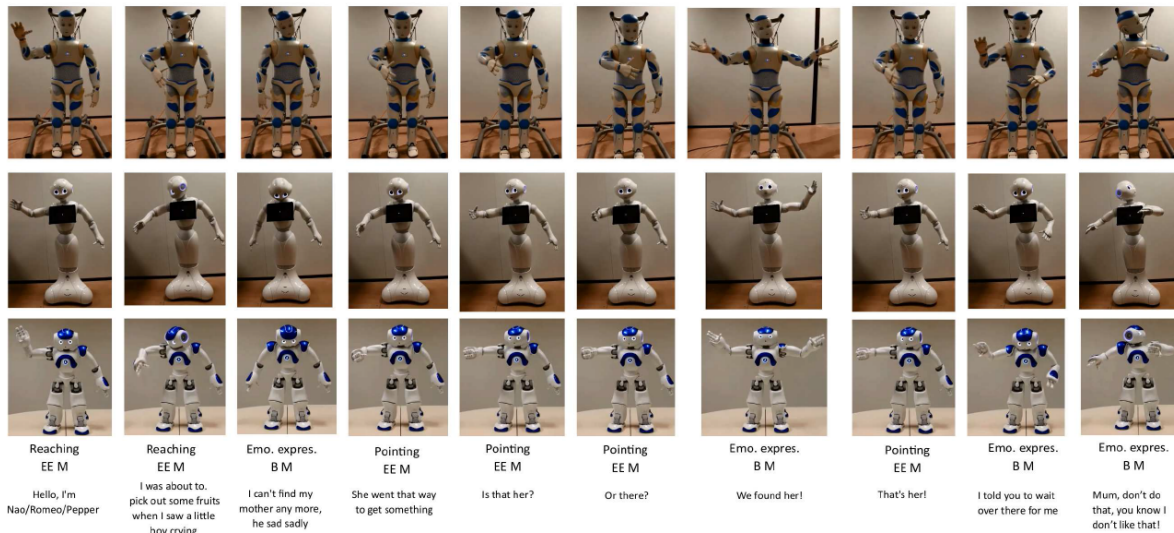
Figure 12: Postures captured from the gestures calculated by the method for the robots Romeo, Pepper and NAO. Below every posture, the type of gesture, the used calculation mode (B M: Block mode or EE M: End-effector mode) and the context in the story are added.

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 his mother back. A number of calculated postures for all three robots are listed in figure 12. 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. See annex 5.2 for further details.

Given the flexibility of the method described above, generating gestures for different morphologies is very fast and straightforward. Using this method, we have proposed a novel methodology [34] to study the influence of different design aspects of social robots on a set of predefined behaviors. By generating a set of gestures for different morphologies, the importance of specific joints and their influence on the performance of the mapped gestures can be studied. Using this methodology in the design process can give interesting insights in the design requirements and help determining the optimal design/complexity trade-off for a certain application, see annex 7.1 for further details.
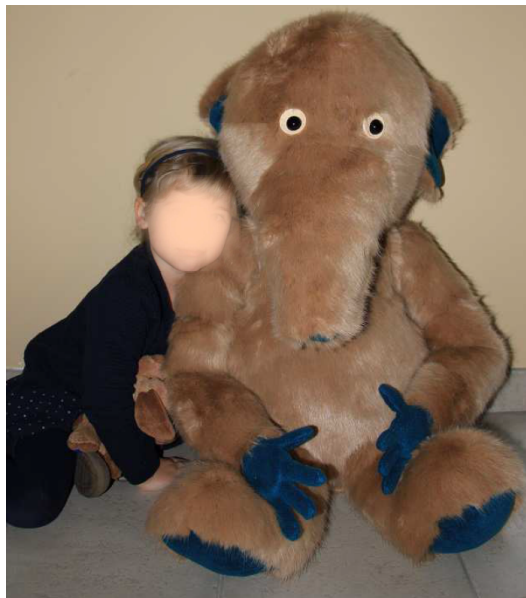
Figure 13: The social robot Elvis.

Following the above mentioned methodology, we applied to design the social robot Elvis, see figure 13. To select a morphology, optimally suited for the DREAM project, a gesture study was performed to investigate the effect of different design aspects on the expressibility. Since the robot is aimed to interact with children on an emotional level, in a first step, a set of emotional expressions was used. The gestures were calculated for a number of different joint configurations, using our previously developed gesture method. Three configurations were selected to study in more detail, and were later physically developed, see figure 14.

The configuration of Elvis-Cc, featuring only 4 DOF in each arm, performed 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 DREAM project, we believe that Elvis-Cb could be the optimal configuration, seen it's performance of the tested emotional expressions, and the possibility of successfully generating additional gestures while keeping the design relatively low-complex with a total of 8 servos per arm.

At this stage, we focused on designing an optimal arm system to generate gestures. To guarantee a safe interaction with children, the actuation mechanism should be improved. Most importantly, a
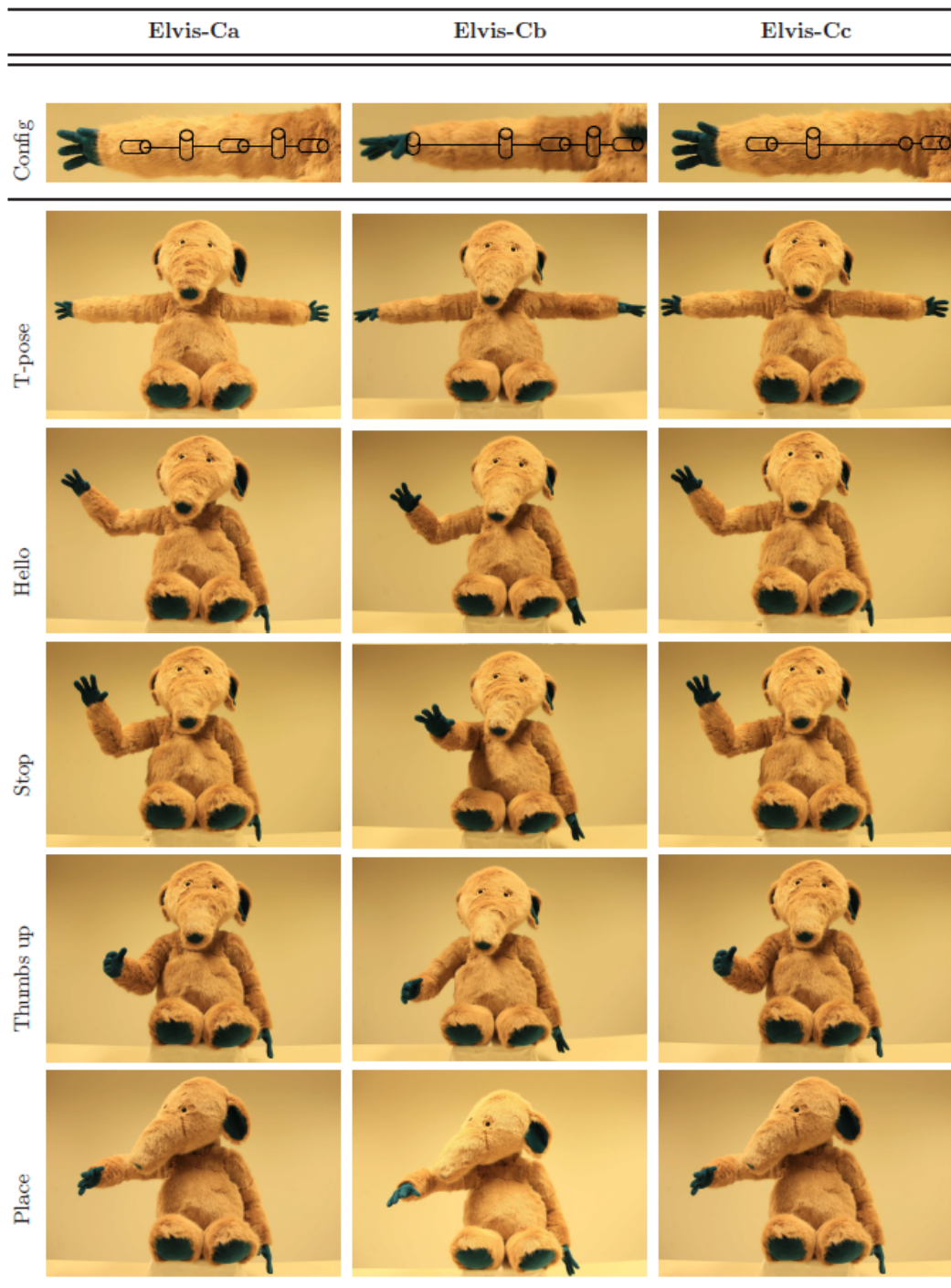
Figure 14: Set of gestures, generated by prototypes Elvis-Ca, Elvis-Cb and Elvis-Cc. The first row visualizes the robot's arm configuration.

feedback concerning the servos' current position should be provided to the global control software. Since at this stage hobby servos are used, the positional feedback is restricted to the servo's internal

control circuit. Other possible improvements include the use of gravity compensation techniques to reduce the necessary motor torques and providing non-back drivable mechanisms to protect the motors.

2.4 Expression and Actuation Subsystem Component

Based on the functional description of the cognitive controller system of the DREAM architecture (see "*Organisation of Cognitive Control and Robot Behaviour*" annex 3.5), a preliminary implementation of the expression and actuation subsystem was formulated. This first version of the component was defined in terms of the input and output ports, following the guidelines established in the software engineering standards (WP3). This is directly informed by the development of the WP6 control architecture in Y1 and Y2, where each subsystem was defined in terms of the interactions with other subsystems, and their functions as outlined in the DREAM DoW. Please refer to figure 1, above, to provide this context. In Y3, the robot interface has been reimplemented for simplicity and performance issues. The main goal of Y3 was to prepare the software required for the clinical study in a supervised autonomy situation. This study using exclusively Nao, it has been decided to rely solely on NaoQi's scheduling capacity to manage conflicting actions rather than the home-made system described in annex 3.4. In Y4 and Y5, the complexity of the actuation and expression subsystem has been reduced by using stored joint trajectories that were generated a priori by the generic gesture method. This approach was implemented and tested in [35, 36], see annex 6.1 and 7.3 for further details.
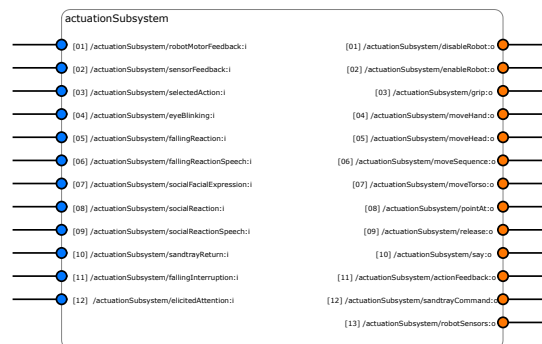
Figure 15: Expression and actuation subsystem component yarp ports. The functional description of this component may be found above.

The main inputs of the actuation subsystem are high level actions received from the therapist through the GUI or from the script to follow through the deliberative subsystem, reactions to events from the attention and reaction subsystem or feedback from the robot interface or the Sandtray. And the outputs are mostly the action primitives for the robot as defined in Deliverable D1.2, sandtray commands and a port to relay the events on the robot sensors to other components. A description of the ports of this preliminary version of the component may be see in figure 15.

The main goal of the actuation subsystem is to transform these high level actions into series of primitives. For example the action for having the robot making a good move is converted into this sequence of actions:

1. Request the coordinates of an image to move from the Sandtray.

2. Transform these coordinates into robot coordinates and call the primitive *point* at this point.

3. Request bezier curve points to the Sandtray.

4. Transform coordinate and simultaneously call a move on the Sandtray and call the primitive *point at* with series of points.

5. Return to idle position when the robot finishes its movement.

Once the desired high-level gesture has been mixed and mapped into the corresponding joint configuration of the robot. The low-level actions are to be executed by the robotic platform.

The execution of these low-level actions is handled in a number of steps, as outlined in the "Robot Low-Level Motor Control" technical report, see annex 3.4. This provides an interface between the control system (handled in a YARP-based system) and the API of the robot hardware (NaoQi in the case of the Nao). The purpose is both to provide a bridge between the two systems, and to provide information to behaviour planning and supervisory oversight regarding the progress of motor command execution, including why a fail occurs if it does. This can be used to inform future action selection for example (by providing feedback for learning).

In addition to this low-level control system, there is the possibility that hardware abstraction can be handled automatically: i.e. that motor commands at the joint level can be determined automatically for different robot embodiments, without having to manually encode each specific action.

Figure 16 shows the description of NaoInterface in term of port configuration. The inputs are the primitive defined in Deliverable D1.2 which are transformed directly into robot commands and call to NaoQi. The outputs are a motor feedback port announcing the end of an action and the result (failure with reason or success) and a sensor feedback port for the sensory event directly sensed by the robot (such as a touch on the head).

naoInterface

[01] /naoInterface/disableRobot:i [01] /naoInterface/robotMotorFeedback:o
[02] /naoInterface/enableRobot:i [02] /naoInterface/sensorFeedback:o
[03] /naoInterface/grip:i
[04] /naoInterface/moveHand:i
[05] /naoInterface/moveHead:i
[06] /naoInterface/moveSequence:i
[07] /naoInterface/moveTorso:i
[08] /naoInterface/pointAt:i
[09] /naoInterface/release:i
[10] /naoInterface/say:i

Figure 16: NaoInterface component yarp ports: the prefix for the port names is listed in the main text. This component is responsible to convert primitive into motor or actuator commands.

References

[1] Serge Thill, Cristina A Pop, Tony Belpaeme, Tom Ziemke, and Bram Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, 2012.

[2] Itoh K., Miwa H., Matsumoto M., Zecca M., Takanobu H., Roccella S., Carrozza M., Dario P., and Takanishi A. Various emotional expressions with emotion expression humanoid robot we-4rii. In *IEEE technical exhibition based conference on robotics and automation*, pages 35–36, 2004.

[3] Ishiguro H., Ono T., Imai M., Maeda T., Kanda T., and Nakatsu R. Robovie: an interactive humanoid robot. *Industrial robot: An international journal*, 28(6):498–504, 2001.

[4] Sugiyama O., Kanda T., Imai M., Ishiguro H., and Hagita N. Natural deictic communication with humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1441–1448, 2007.

[5] Ido J., Matsumoto Y., Ogasawara T., and Nisimura R. Humanoid with interaction ability using vision and speech information. In *IEEE/RSJ International conference on Intelligent Robots and Systems*, pages 1316–1321, 2006.

[6] Zecca M., Mizoguchi Y., Endo K., Iida F., Kawabata Y., Endo N., Itoh K., and Takanishi A. Whole body emotion expressions for kobian humanoid robot: preliminary experiments with different emotional patterns. In *In the 18th IEEE international sympsium on robot and human interactive communication*, pages 381–386, 2009.

[7] Matsui D., Minato T., MacDorman K., and Ishiguro H. Generating natural motion in an android by mapping human motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3301–3308, 2005.

[8] Tapus A., Peca A., Aly A., Pop C., Jisa L., Pintea S., Rusu A.S., and David D.O. 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.

[9] Do M., Azad P., Asfour T., and Dillmann R. Imitation of human motion on a humanoid robot using non-linear optimization. In *IEEE/RAS International Conference on Humanoid Robots*, pages 545–552, 2008.

[10] M. Salem, S. Kopp, I. Wachsmuth, and F. Joublin. Towards meaningful robot gesture. In *Human Centered Robot Systems*, pages 173–182. Springer, 2009.

[11] A. Beck, L. Cañamero, and K. A. Bard. Towards an affect space for robots to display emotional body. Ro-man, 2010.

[12] M. Hild, T. Siedel, C. Benckendorff, C. Thiele, and M. Spranger. Myon, a new humanoid. In *Language Grounding in Robots*, pages 25–44. Springer, 2012.

[13] K. Goris, J. Saldien, B. Vanderborght, and D. Lefeber. Mechanical design of the huggable robot probo. *International Journal of Humanoid Robotics*, 8(03):481–511, 2011.

[14] F. Tanaka, K. Noda, T. Sawada, and M. Fujita. Associated emotion and its expression in an entertainment robot qrio. In *Entertainment Computing-ICEC 2004*, pages 499–504. Springer, 2004.

[15] G. Saponaro and A. Bernardino. Generation of meaningful robot expressions with active learning. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 243–244. ACM, 2011.

[16] C. Stanton, A. Bogdanovych, and E. 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.

[17] Q. A. Le, S. Hanoune, and C. Pelachaud. Design and implementation of an expressive gesture model for a humanoid robot. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 134–140. IEEE, 2011.

[18] 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.

[19] John F Soechting and Martha Flanders. Errors in pointing are due to approximations in sensorimotor transformations. *Journal of Neurophysiology*, 62(2):595–608, 1989.

[20] Ekman P and Friesen W. *Facial Action Coding System*. Consulting Psychologists Press, 1978.

[21] A. P. Atkinson, W. H. Dittrich, A. J. Gemmell, A. W. Young, et al. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception-London*, 33(6):717–746, 2004.

[22] Jelle Saldien, Kristof Goris, Bram Vanderborght, Johan Vanderfaeillie, and Dirk Lefeber. Expressing emotions with the social robot probo. *International Journal of Social Robotics*, 2(4):377–389, 2010.

[23] Jonathan Posner, James A Russell, and Bradley S Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17(03):715–734, 2005.

[24] Website. http://www.rocketbox-libraries.com.

[25] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1321–1326. IEEE, 1998.

[26] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jér Ome 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.

[27] Junchao Xu, Joost Broekens, Koen Hindriks, and Mark A Neerincx. *Mood expression through parameterized functional behavior of robots*. IEEE, 2013.

[28] Junchao Xu, Joost Broekens, Koen Hindriks, and Mark A Neerincx. Bodily mood expression: Recognize moods from functional behaviors of humanoid robots. In *International conference on social robotics*, pages 511–520. Springer, 2013.

[29] Atsushi Yamaguchi, Yoshikazu Yano, Shinji Doki, and Shigeru Okuma. A study of emotional motion description by motion modification and adjectival expressions. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pages 1–6. IEEE, 2006.

[30] R. Hirose and T. Takenaka. *Development of the humanoid robot asimo*. Technical report, 2001. Honda R&D Technical Review 13(1): 16.

[31] Ch Ott, Oliver Eiberger, Werner Friedl, B Bauml, Ulrich Hillenbrand, Ch Borst, Alin Albu-Schaffer, Bernhard Brunner, H Hirschmuller, S Kielhofer, et al. A humanoid two-arm system for dexterous manipulation. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 276–283. IEEE, 2006.

[32] Aldebaran Robotics. Romeo documentation. `http://doc.aldebaran.com/2-4/home_romeo.html`, 2014. accessed February-2016.

[33] Aldebaran Robotics. Pepper documentation. `http://doc.aldebaran.com/2-4/home_pepper.html`, 2014. accessed February-2016.

[34] Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gomez Esteban, Dirk Lefeber, and Bram Vanderborght. Studying design aspects for social robots using a generic gesture method. *International Journal of Social Robotics*, 2018. To be published.

[35] 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.

[36] Hoang-Long Cao, Lars Christian Jensen, Xuan Nhan Nghiem, Huong Vu, Albert De Beir, Pablo Gomez Esteban, Greet Van de Perre, Dirk Lefeber, and Bram Vanderborght. Dualkeepon: a human–robot interaction testbed to study linguistic features of speech. *Intelligent Service Robotics*, pages 1–10, 2018.

3   Period 1 Annexes

3.1   Van de Perre, G. et al. (2014), Development of a generic method to generate upper-body emotional expressions for different social robots

Bibliography -   Van de Perre, G., Van Damme, M., Lefeber, D., and Vanderborght, B. (2014) Development of a generic method to generate upper-body emotional expressions for different social robots, *submitted to International Journal on Advanced Robotics for the special issue on humanoid robotics*.

Abstract -   To investigate the effect of gestures in human-robot interaction, a number of social robots capable of gesturing have been designed. Gestures are often preprogrammed 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. This method aims to minimize the workload when implementing gestures on a new robot platform and facilitate the sharing of gestures between different robots. The innovative aspect of this method is that it is constructed independently of any robot configuration, and therefore it can be used to generate gestures for different robot platforms. To calculate a gesture for a certain configuration, the developed method uses a set of target gestures listed in a database and maps them to that specific configuration. The database currently consists of a set of emotional expressions. The method was validated on the virtual model of different robots.

Relation to WP -   This work directly contributes to Task T6.4.

3.2   Cao, H.L. et al. (2014), Enhancing My Keepon robot: A simple and low-cost solution for robot platform in Human-Robot Interaction studies

Bibliography -   Cao, H.L., Van de Perre, G., Simut, R., Pop, C., Peca, A., Lefeber, D., Vanderborght, B. (2014), Enhancing My Keepon robot: A simple and low-cost solution for robot platform in Human-Robot Interaction studies. *In* the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, Scotland, UK.

Abstract -   Many robots capable of performing social behaviours have recently been developed for Human-Robot Interaction (HRI) studies. Besides the undisputed advantages, a major difficulty in HRI studies with social robots is that the robot platforms are typically expensive and/or not open-source. It burdens researchers to broaden experiments to a larger scale or apply study results in practice. This paper describes a method to modify My Keepon, a toy version of Keepon robot, to be a programmable platform for HRI studies, especially for robot-assisted therapies. With an Arduino micro-controller board and an open-source Microsoft Visual C# software, users are able to fully control the sounds and motions of My Keepon, and configure the robot to the needs of their research. Peripherals can be added for advanced studies (e.g., mouse, keyboard, buttons, PlayStation2 console, Emotiv neuroheadset, Kinect). Our psychological experiment results show that My Keepon modification is a useful and low-cost platform for several HRI studies.

Relation to WP -   This work directly contributes to Task T6.4.

3.3 De Beir, A. (2016), Enhancing Nao Expression of Emotions Using Pluggable Eyebrows

Bibliography - De Beir, A., Cao, H. L., Esteban, P. G., Van de Perre, G., and Vanderborght, B. (2016). Enhancing Nao Expression of Emotions Using Pluggable Eyebrows. *International Journal of Social Robotics*. Subject to minor revisions.

Abstract - Robots can express emotions for better Human Robot Interaction. In this field, NAO robot is a platform widely used. This robot mainly expresses emotions by gestures and colored LED eyes, but, due to its white at and inanimate face, the robot cannot express facial expressions. This work proposes a pluggable eyebrows device allowing NAO to express anger or sadness while performing other tasks. This device is plug-and-play and can be controlled directly by NAO's main software. Additionally we develop a platform independent mapping of colors and eyebrows angles with emotions. We first conducted an experiment that qualitatively attests the interest of this device. Three following experiments were conducted to: 1) Confirm the relation between eyebrows angle and expressed emotion; 2) evaluate different shapes in order to select the most appropriate one; 3) prove that NAO is able to use the eyebrows to express emotions while performing non emotional tasks.

Relation to WP - This work directly contributes to Task T6.4.

3.4 Baxter, P. et al. (2014), Technical Report: Robot Low-Level Motor Control

Abstract - This technical report describes the first version of the low-level robot control system using YARP as the communications infrastructure. This system is designed to be extensible, and flexible to the requirements of the higher level robot behavioural components. A demonstrator system has been constructed for the Nao, but the structure is intended to be applicable to other robot embodiments (i.e. specifically the Probo, assuming a similar level of partially abstracted control is possible).

Relation to WP - This work directly contributes to Task T6.4.

3.5 Baxter, P. et al. (2015), Technical Report: Organisation of Cognitive Control and Robot Behaviour

Abstract - The purpose of this technical report is to summarise the motivations and constraints underlying the cognitive control structures, and to outline an organisation of these subsystems. This is a proposal only; this document is intended to be a working one, to be updated as required during development. This version of the report is based primarily on the discussions that took place in Brussels (23/01/15).

Relation to WP - This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

4 Period 2 Annexes

4.1 Kennedy, J. et al. (2015). Can less be more? the impact of robot social behaviour on human learning

Bibliography - Kennedy, J., Baxter, P., and Belpaeme, T. (2015). Can less be more? The impact of robot social behaviour on human learning. In Salem, M., Weiss, A., Baxter, P., and Dautenhahn, K. (Eds.) Proceedings of the 4th International Symposium on New Frontiers in HRI at AISB.

Abstract - In a large number of human-robot interaction (HRI) studies, the aim is often to improve the social behaviour of a robot in order to provide a better interaction experience. Increasingly, companion robots are not being used merely as interaction partners, but to also help achieve a goal. One such goal is education, which encompasses many other factors such as behaviour change and motivation. In this paper we question whether robot social behaviour helps or hinders in this context, and challenge an often underlying assumption that robot social behaviour and task outcomes are only positively related. Drawing on both human-human interaction and human-robot interaction studies we hypothesise a curvilinear relationship between social robot behaviour and human task performance in the short-term, highlighting a possible trade-off between social cues and learning. However, we posit that this relationship is likely to change over time, with longer interaction periods favouring more social robots.

Relation to WP - This paper informs the generation of autonomous behaviours, and systems used in this research are also used in DREAM.

4.2 Van de Perre, G. et al. (2016), Reaching and pointing gestures calculated by a generic gesture system for social robots

Bibliography - Van de Perre, G., De Beir, A., Cao, H.L., Gómez Esteban, P., Lefeber, D. and Vanderborght, B. (2016), Reaching and pointing gestures calculated by a generic gesture system for social robots. *To be published in Robotics and Autonomous systems.*

Abstract - Since the implementation of gestures for a certain robot generally involves the use of specific information about it's 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. In this paper, we discuss the novel end-effector mode of the method, which can be used to calculate gestures whereby the position of the end-effector is important, for example for reaching for or pointing towards an object. The interesting and innovative feature of our method is its high degree of flexibility in both the possible configurations wherefore the method can be used, as in the gestures to be calculated. The method was validated on several configurations, including those of the robots ASIMO, NAO and Justin. In this paper, the working principles of the end-effector mode are discussed and a number of results are presented.

Relation to WP - This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

4.3 Wills, P. et al. (2016) Socially Contingent Humanoid Robot Head Behaviour Results in Increased Charity Donations

Bibliography - Wills, P., Baxter, P., Kennedy, J., Senft, E. and Belpaeme, T. (2016) Socially Contingent Humanoid Robot Head Behaviour Results in Increased Charity Donations. In Proceedings of the 11th Annual ACM/IEEE International Conference on Human-Robot Interaction.

Abstract - The role of robot social behaviour in changing peoples behaviour is an interesting and yet still open question, with the general assumption that social behaviour is beneficial. In this study, we examine the effect of socially contingent robot behaviours on a charity collection task. Manipulating only behavioural cues (maintaining the same verbal content), we show that when the robot exhibits contingent behaviours consistent with those observable in humans, this results in a 32% increase in money collected over a non-reactive robot. These results suggest that apparent social agency on the part of the robot, even when subtle behavioural cues are used, can result in behavioural change on the part of the interacting human.

Relation to WP - This modest study shows that contingent social behaviour production in a social robot can have an important impact on the behaviour of onlookers. While tangential to the purpose of DREAM, the paper underlines that the study of contingent behaviour should not be neglected.

5 Period 3 Annexes

5.1 Van de Perre, G. et al. (2017a), Generic method for generating blended gestures and affective functional behaviors for social robots

Bibliography - Van de Perre, G., Cao, H.L., De Beir, A., Gómez Esteban, P., Lefeber, D. and Vanderborght, B. (2017), Generic method for generating blended gestures and affective functional behaviors for social robots. Autonomous Robots, 42(3), 569-580.

Abstract - Gesturing is an important modality in human-robot interaction. Up to date, gestures are often implemented for a specific robot configuration and therefore not easily transferable to other robots. To cope with this issue, we presented a generic method to calculate gestures for social robots. The method was designed to work in two modes to allow the calculation of different types of gestures. In this paper, we present the new developments of the method. We discuss how the two working modes can be combined to generate blended emotional expressions and deictic gestures. In certain situations, it is desirable to express an emotional condition through an ongoing functional behavior. Therefore, 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 implementations were validated on different configurations, including those of NAO, Justin and ASIMO.

Relation to WP - This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

5.2 Van de Perre, G. et al. (2017b), Generating gestures for different robot morphologies through one generic gesture system: validation on physical robots

Bibliography - Van de Perre, G., Cao, H.L., De Beir, A., Gómez Esteban, P., Lefeber, D. and Vanderborght, B. (2017), Generic method for generating blended gestures and affective functional behaviors for social robots. *To be published in International Journal of Robotics Research.*

Abstract - To overcome the difficulties in transferring joint trajectories to different robots, we proposed the use of a generic system to calculate gestures for social robots. The developed method 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 previous work, the different modalities were validated on the virtual model of different robots. In this paper, we present the innovations made to the method to be able to use it on physical robots. 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.

Relation to WP - This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

6 Period 4 Annexes

6.1 Cao, Hoang-Long et al. (2018), A personalized and platform-independent behavior control system for social robots in therapy: development and applications

Bibliography -   Cao, H. L., Van de Perre, G., Kennedy, J., Senft, E., Esteban, P. G., De Beir, A., Simut, R., Belpaeme, T., Lefeber, D., and Vanderborght, B. A personalized and platform-independent behavior control system for social robots in therapy: development and applications. *To be published in IEEE Transactions on Cognitive and Developmental Systems*.

Abstract -   Social robots have been proven beneficial in different types of health care interventions. An ongoing trend is to develop (semi-)autonomous socially assistive robotic systems in health care context to improve the level of autonomy and reduce human workload. This paper presents a behavior control system for social robots in therapies with a focus on personalization and platform-independence. This system architecture provides the robot an ability to behave as a personable character, which behaviors are adapted to user profiles and responses during the human-robot interaction. Robot behaviors are designed at abstract levels and can be transferred to different social robot platforms. We adopt the component-based software engineering approach to implement our proposed architecture to allow for the replaceability and reusability of the developed components. We introduce three different experimental scenarios to validate the usability of our system. Results show that the system is potentially applicable to different therapies and social robots. With the component-based approach, the system can serve as a basic framework for researchers to customize and expand the system for their targeted health care applications.

Relation to WP -   This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

7 Period 5 Annexes

7.1 Van de Perre, G. et al. (2019), Studying Design Aspects for Social Robots Using a Generic Gesture Method

Bibliography - Van de Perre, G., De Beir, A., Cao, H.L., Gómez Esteban, P., Lefeber, D. and Vanderborght, B. (2018) Studying Design Aspects for Social Robots Using a Generic Gesture Method. *To be published in International Journal of Social Robotics, https://doi.org/10.1007/s12369-019-00518-x.*

Abstract - Since social robots are aimed to interact and communicate with humans in a natural way and operate 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 their specific application. In this paper, we propose a novel methodology to study the influence of different design aspects, based on a generic gesture method. The gesture method was developed to overcome the difficulties in transferring gestures to different robots, providing a solution for the correspondence problem. A small set of morphological information, inputted by the user, is used to evaluate the generic framework of the software at run-time. Therefore, gestures can be calculated fast and easy for a desired robot configuration. By generating a set of gestures for different morphologies, the importance of specific joints and their influence on a series of postures and gestures 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 specified task, and can give interesting insights in the required joint angle range. In this paper, this design methodology is illustrated by using the virtual model of the robot Probo.

Relation to WP - This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

7.2 Van de Perre, G. et al. (2019), Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing

Bibliography - Van de Perre, G., Cao, H.L., De Beir, A., Gómez Esteban, P., Lefeber, D. and Vanderborght, B. (2019) Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing. *Submitted to International Journal of Robotics Research.*

Abstract - Depending on their exact application, social robots have been designed with different arm morphologies, ranging from under-actuated designs to arms featuring 9 degrees of freedom (DOF). It is however difficult to investigate if the chosen arm morphology is the best possible solution for the intended application. Existing robots differ from each other in a large range of aspects and therefore, 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-offs in the design process of new robots, we proposed a methodology based on the calculation of gestures for different morphologies and their visualization on one single virtual model. In this paper, we present the design process followed to select and optimal morphology for our new social robot Elvis. A gesture study was performed to select an optimal morphology, resulting in three interesting joint configurations that were studied in more detail and later physically developed. To realize the three different morphologies, the arm system was designed semi-modular, allowing different

modules to be switched. Different gestures were generated for all three Elvis-variants,resulting in the selection of one final morphology to be used for future work.

Relation to WP -   This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

7.3   Cao, H-L. et al. (2019), DualKeepon: a humanrobot interaction testbed to study linguistic features of speech

Bibliography -   Cao, H-L., Jensen, L.C., Nghiem, X.N., Vu, H., De Beir, A., Esteban, P.G., Van de Perre, G., Lefeber, D. and Vanderborght, B. (2019) DualKeepon: a humanrobot interaction testbed to study linguistic features of speech. *To be published in Intelligent Service Robotics.*

Abstract -   In this paper, we present a novel dual-robot test-bed called DualKeepon for carrying out pairwise comparisons of linguistic features of speech in humanrobot interactions. Our solution, using a modified version of the MyKeepon robotic toy developed by Beatbots, is a portable open-source system for researchers to set up experiments quickly, and in an intuitive way. We provide an online tutorial with all required materials to replicate the system. We present two humanrobot interaction studies to demonstrate the test-bed. The first study investigates the perception of robots using filled pauses. The second study investigates how social roles, realized by different prosodic and lexical speaking profiles, affect trust. Results show that the proposed test-bed is a helpful tool for linguistic studies. In addition to the basic setup, advanced users of the system have the ability to connect the system to different robot platforms, i.e., NAO, Pepper.

Relation to WP -   This work directly contributes to Task T6.4, and the general organisation of the other systems within WP6.

Period 1

FULL PAPER

Development of a generic method to generate upper-body emotional expressions for different social robots

Greet Van de Perre*, Michaël Van Damme, Dirk Lefeber and Bram Vanderborght

Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium

(*v1.0 March 2014*)

To investigate the effect of gestures in human-robot interaction, a number of social robots capable of gesturing have been designed. Gestures are often preprogrammed 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. This method aims to minimize the workload when implementing gestures on a new robot platform and facilitate the sharing of gestures between different robots. The innovative aspect of this method is that it is constructed independently of any robot configuration, and therefore it can be used to generate gestures for different robot platforms. To calculate a gesture for a certain configuration, the developed method uses a set of target gestures listed in a database and maps them to that specific configuration. The database currently consists of a set of emotional expressions. 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 developed method to map gestures to different robot morphologies gives good results.

Keywords: Generic gesture system, upper body posture, emotions

1.   Introduction

Gesturing is an important research topic in social robotics. As in human-human communication, gesturing is stated to be an essential feature to ensure natural and fluent communication. Indeed, in [1] it has been shown that gesturing is an important communication factor in HRI; 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 anthorpomorphism of a robot was also investigated by Salem et al. [2]. A number of robots capable of gesturing have been developed to study different aspects of gesturing in HRI. Gestures implemented in robots are however, up to now, subject to two important limitations. Firstly, the gestures implemented in a robot are always limited to a set of gestures necessary for the current research, and often limited to one type of gestures. The robot WE-4RII [3] for example, was developed to study human-like emotion, hence, the incorporated gestures are mainly focussed on emotional expressions. On the other hand, the developers of Robovie aimed for communication robots that interact naturally with humans. Since Robovies applications were focussed on object indication and route direction-giving, mostly deictic gestures were used [4]. The reason for the limited amount of gestures implemented in specific robots can be found in the second limitation;

---

*Corresponding author. Email: Greet.Van.de.Perre@vub.ac.be

namely the way gestures are implemented. Gestures are mostly preprogrammed off-line for the current robot configuration. The resulting postures are stored in a database and are replayed during interaction. This is the case for, amongst others, Robovie [5], HRP-2 [6] and Kobian [7]. Since the postures 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 is for example the case for Repliee Q2 [8], where a marker-based motion capture system is used. Another possibility is to use the Kinect to perform skeleton tracking [9]. In [10], both a marker-based (Vicon) as a markerless motion capture system was used to reproduce human motion for the robot ARMAR-IIIb. Since the mapping of the captured data is robot specific, also these resulting gestures are dependent on the morphology and not usable for other robots. This issue is know as the correspondence problem [11], [12]. 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 of different species, or 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 and when working with a new robot platform, new joint trajectories to reach the desired postures are calculated and implemented. This approach is time consuming; it would be much more efficient to make the implementation of gestures more flexible and to design a general method that allows easily implementing gestures in different robots. This methodology, depicted in Figure 1, fits in the objectives of ROS [13] and OROCOS [14] to make software modules and commonly used functionalities available for different platforms and those of RoboEarth to make the sharing of information and knowledge between robots possible [15].

The method proposed in this paper aims for this, and therefore, we aim to provide a solution for the correspondence problem. The innovative aspect of this method is that it is constructed independently of any robot configuration, and therefore it can be used to generate gestures for different robot platforms. The configuration of the robot is used as input, and the joint angles needed to establish a desired gesture or posture 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. We believe this generic method can be useful for different research teams since it easily allows gestures to be shared between different robots and minimizes the workload when implementing gestures on a new robot platform. Another approach to flexibly generate gestures by different robots is by using neural networks as in [16]. However, this technique requires training while the method proposed here is very straightforward in use. In both [17] and [18], a gesture framework initially developed for virtual agents is applied on a humanoid robot. In [17], the speech and gesture production model developed for the virtual agent Max is used to generate gestures for the ASIMO robot. Here, an XML-based Multi-modal Utterance Representation Markup Language (MURML) is used to abstractly describe gestures by specifying three features; the location of the wrist, the shape of the hand and the orientation of the wrist. 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. Similarly, in [18], gestures are described independently of the embodiment by specifying features as the hand shape, wrist position and palm orientation. To generate gestures for the NAO robot, the correct angles for the shoulder and elbow joints are selected from a predetermined table listing all possible wrist positions and the corresponding joint values. The values for the remaining joints, namely the wrist joint and fingers are calculated by taking into consideration the values of other features such as the hand shape and palm orientation. So although the gestures are described independently of the robot configuration, mapping these gestures to the robot requires hard coded joint information. Our developed method aims to fully automate the mapping of gestures to a random robot configuration.

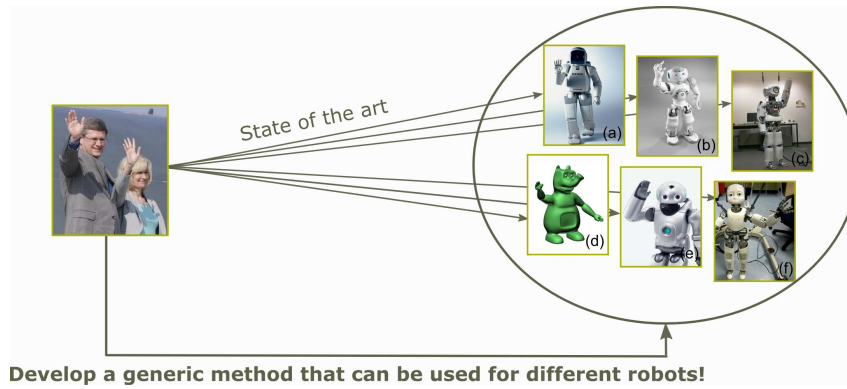**Develop a generic method that can be used for different robots!**

Figure 1. In the state of the art, gestures are always implemented for a specific robot platform. Our method aims to facilitate implementing gestures for a new robot platform by storing gestures independently of a morphology, and mapping them on a specific configuration. Robots: (a) ASIMO [19], (b) NAO [20], (c) Myon [21], (d) Probo [22], (e) QRIO [23], (f) iCub [24].

2. Methodology

To generate gestures for a certain configuration, the developed method uses a set of target gestures listed in a database and maps them to that specific configuration. For manipulation and pointing, the end effector position is crucial while for some other gestures, including emotional expressions, the overall pose of the arms is very important to convey the gesture. In [25], 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 gesture. Salem et al. [17] decided to work with the end effector pose and calculate task-space trajectories using inverse kinematics, based on the findings of [26]. However, for a generic method usable by robots with different joint configurations and link lengths, a good scaling of the end effector position, depending on the robot configuration, is crucial to guarantee a natural and human-like overall calculated posture. Therefore, 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. In the end effector mode, the position and orientation of the end effector necessary to, for example, point in a desired direction or grasp a certain object will be imposed. The corresponding joint angles will then be calculated by inverse kinematics. It will be possible to combine the two working modes of the method, so that emotional information can be conveyed while performing an end effector depending gesture (e.g. waving towards a person when feeling happy). This paper will focus on the working principle and results of the block mode.

To ensure a good overall posture in the block mode, it is necessary to take into account the relative orientations of every joint complex the robot has in common with a human, and not only impose the orientation and position of the end effector. A base human model was defined, and the target postures were described by the orientation of the different joint complexes in the model. This is discussed in detail in Section 3. To make a certain model or robot perform a desired emotional expression, this information is mapped to its joint configuration. The methodology of this mapping is covered in Section 4. Section 5 describes the results of the method for a number of configurations together with the results of a survey aimed to validate the method, followed by a conclusion and a perspective on the future work.

3.   Base Model

3.1   *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 [3, 27], we opted to work not only with static postures but to implement the possibility of using motion sequences. The target gestures in the database therefore consist of a series of postures specified in time. Most of the target gestures were chosen by using the UCLIC Affective Body Posture and Motion Database [28]. 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, one of the best-scoring motion capture sets 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 in a later stage to allow for some variance of the gestures during human-robot interaction or other types of movements can be included.

3.2   *Body Action Coding System*

To describe the target postures in a quantitative way, a Body Action Coding System (BACS) was developed. Similar to the Facial Action Coding System of Ekman and Friesen [29], which defines a number of (Facial) Action Units to describe facial expressions, a set of Body Action Units (BAU's) is defined. While the Facial AU's are defined as a muscle or a muscle group, our BAU's are based on the human terms of motion. The defined BAU's are listed in Table 1. Although the leg movements also contribute to the overall performance of the gesture, for a first validation of the method we decided to focus on the upper body movements. The BAU's are therefore restricted to the upper body. The units are grouped into different *blocks*, corresponding to one human joint complex, such as the shoulder or the wrist. These blocks can subsequently be grouped into three body parts, namely the head, body and arm, which we refer to as *chains*. In that way, a base human model was defined, consisting 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 human 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 1).

The target body postures could then be taxonomized into the activation of the BAU's. They are described by the orthopaedic angles of every block of the base model. Orthopaedic angles are similar to Euler angles, but are defined according to clinical terms such as flexion and abduction [30]. A similar strategy as in [30] was used; a standard reference frame was defined, whereby the 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). The orientation of block i is then 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). The data is stored in the program as rotation matrices.

Table 1.   The Body Action Coding System

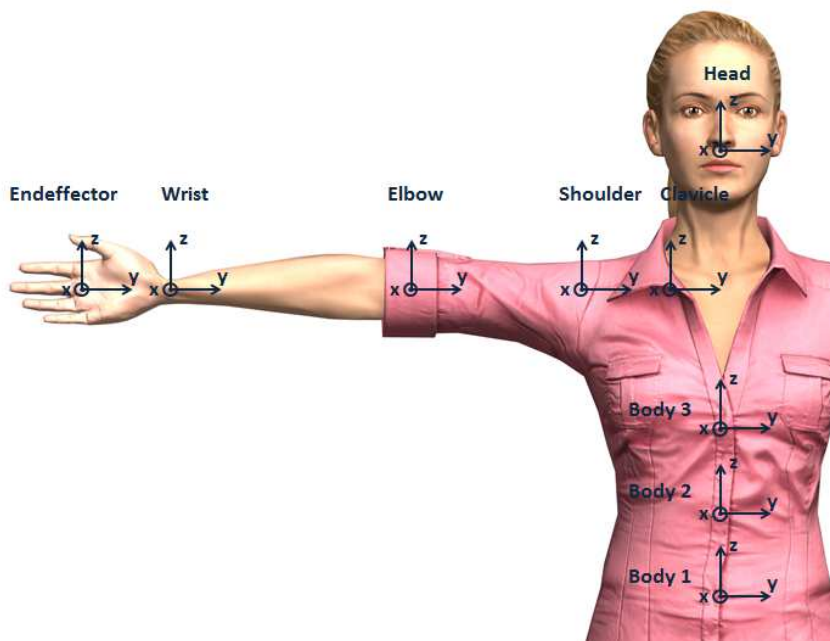| Chain	Block	BAU	Description
Head	Head	1	Flexion/extension of neck
		2	Abduction/adduction of neck
		3	Rotation of neck
Body	Body	4	Flexion/extension of spinal column
		5	Lateral flexion of spinal column
		6	Transversal rotation of spinal column
Arm	Clavicle	7	Abduction/adduction of shoulder girdle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

Figure 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.

4.   Mapping the gestures to a configuration

To make a model or robot perform a desired gesture, the target posture sequences described in Section 3 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 [31] for example, is a snowman-like robot without arms. Therefore, the mapping of a posture will be 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 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 successors is displayed. For the base model these are the assigned frames as discussed in Subsection 3.2, 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 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 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 on 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:

$$R_i = {}^{b,i}R_{st} \cdot R_{i,des} \cdot {}^{st}R_{e,i} \tag{1}$$

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. 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 easily map gestures to different robots.

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 are calculated by interpolation 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.

Table 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	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}0&0&-1\\0&1&0\\1&0&0\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$
Config 1	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&0&-1\\0&1&0\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&0&-1\\0&1&0\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$
Config 2	$\begin{pmatrix}0&0&1\\0&-1&0\\1&0&0\end{pmatrix}$ $\begin{pmatrix}0&0&1\\-1&0&0\\0&-1&0\end{pmatrix}$ $\begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}0&0&1\\0&-1&0\\1&0&0\end{pmatrix}$ $\begin{pmatrix}0&0&1\\-1&0&0\\0&-1&0\end{pmatrix}$ $\begin{pmatrix}0&-1&0\\-1&0&0\\0&0&-1\end{pmatrix}$ $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$

4.1 *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 1. For each block, a mapped rotation matrix is calculated as described in Section 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. Since in this application the speeds are relatively low, it is sufficient to specify the end effector pose and speed. Hence, a first order algorithm was chosen. For each block, the joint angles are calculated by the closed-loop inverse kinematics algorithm shown in Figure 3 [32]. In a first step, the time derivate of the joint angles $\dot{q}$ is calculated:

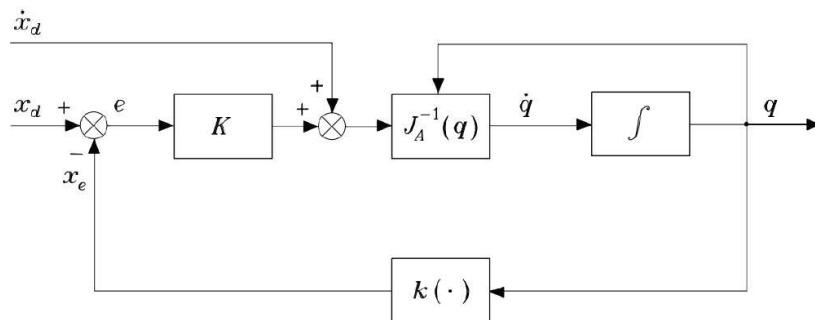$$\dot{q} = J_A^{-1}(q)(\dot{x_d} + K(x_d - x_e)) \tag{2}$$

Figure 3.   Schematic representation of the closed loop inverse kinematic algorithm used to calculate the joint angles for the desired body posture [32].

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. $J_A(q)$ is the analytical jacobian, 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 [33]. Since the complete configuration has the same rotational possibilities as the base human 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 calculated posture. The loop of the algorithm is closed by calculating the new actual end effector orientation by direct kinematics (depicted by $k(\cdot)$ in Figure 3) and using it as input to determine the current error.

4.2   *Configuration with a reduced number of DOF's*

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 [3] 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 [34] and ASIMO [35]. Joints in the wrist are also often omitted. NAO [36] and QRIO [23] 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 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
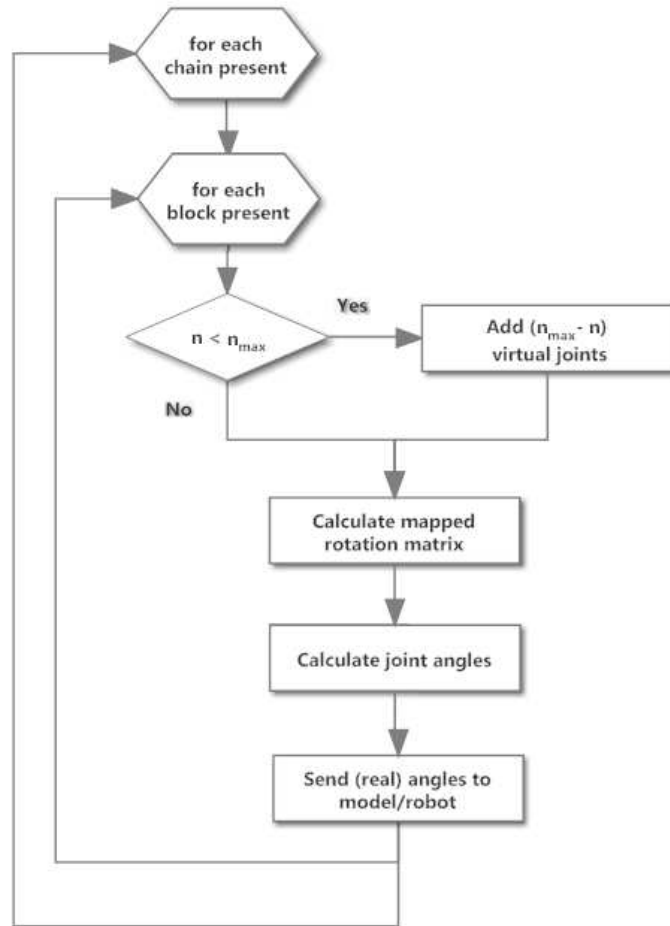
Figure 4. Schematic representation of the program flow. For every block present in the configuration, a mapped rotation matrix is calculated. The corresponding joint angles are calculated by inverse kinematics and sent to the virtual model or robot.

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. After the calculation, the angles corresponding to real and virtual joints are separated; only the real joint angles are sent to the virtual model to visualize the posture. Figure 4 summarises the program's work flow to calculate the correct joint angles for a desired posture.

5.   Results

The method was validated on several configurations. Table 3 shows a calculated posture for different arm configurations. The top row shows the base model with the targeted posture, in this case the end posture for the emotional expression for *happiness*. The second row shows a complete configuration, with a 9 DOF arm, 3 DOF head and 9 DOF body (virtual model from the RocketBox Libraries [37]. 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. Configuration 2 shows the ASIMO robot [38]. 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 constains 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

9

joint angels can be calculated to reach the desired orientation of the blocks. But since the body 1 block is incomplete, virtual joints needs 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 [39]. Its body contains three joints and is 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 colunm. Also the head block is incomplete: it consists of two joints, corresponding to BAU 1 and 3, 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 as ASIMO's arm configuation. 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 [40]. 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 when going from the T-pose to the end posture for the expression for *happiness* are plotted in Figure 5. 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 a dotted line.

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 at the website *http://probo.vub.ac.be/GestureSystem*. The end posture of the gestures can be found in Table 4. The survey's objective was to investigate the quality of the mapped gestures, so to check whether the calculated gestures for different configurations are recognizable from 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 asked to link the shown gesture with one of the target gestures by means of a multiple choice form (Figure 6). 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 [41]. 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 5. The overall rates for the correct linking of the gestures are relatively high, whereof we can conclude
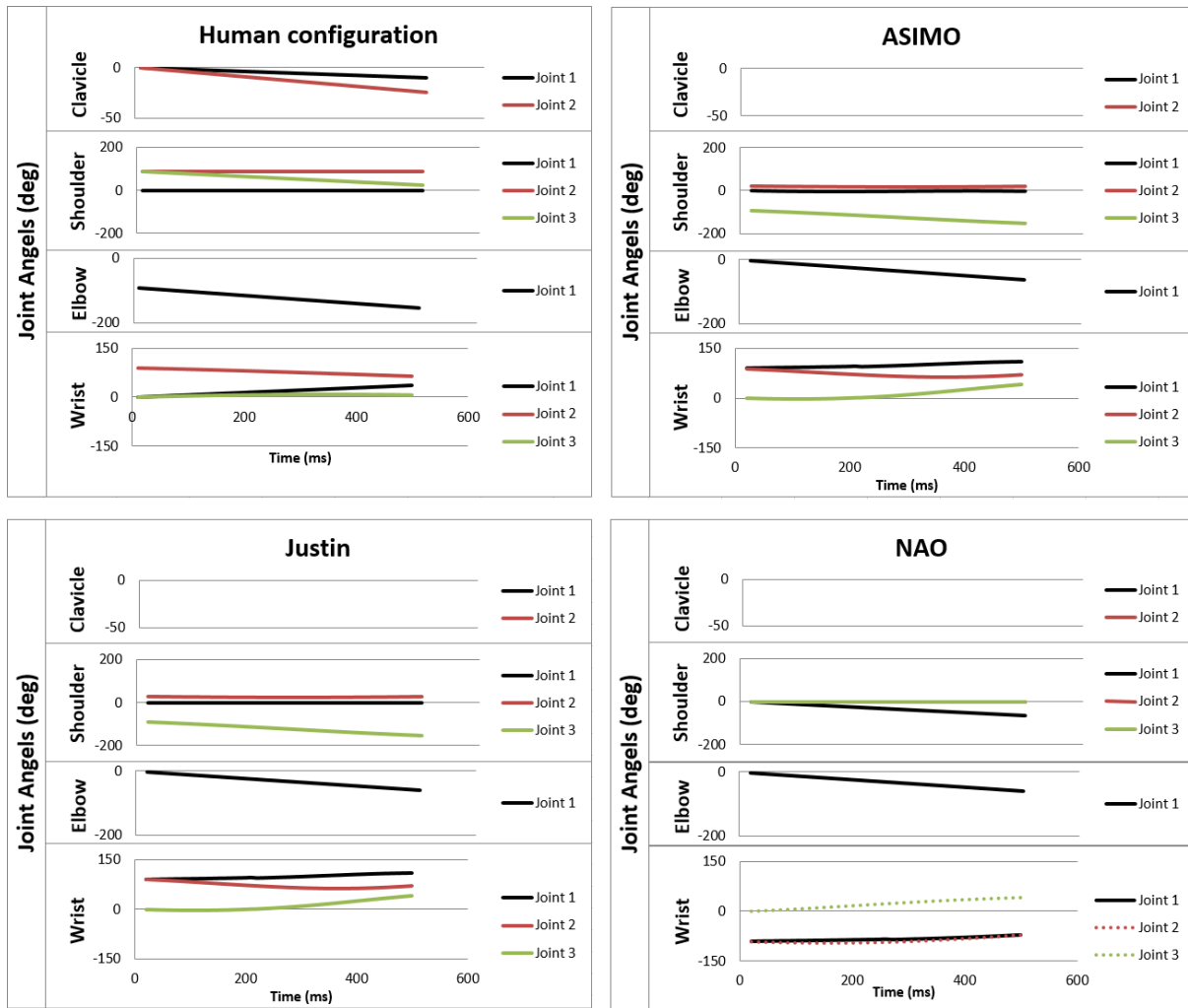
Figure 5.   This figure shows the calculated joint trajectories for the left arm chain for the four configurations listed in Table 3 when going from the T-pose to the end posture for the expression for *happiness*. As missing blocks are ignored by the method, no trajectories are calculated for the clavicle 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.

that the calculated gestures in general well resemble the target gestures from the database and therefore, that our method 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.

Table 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.
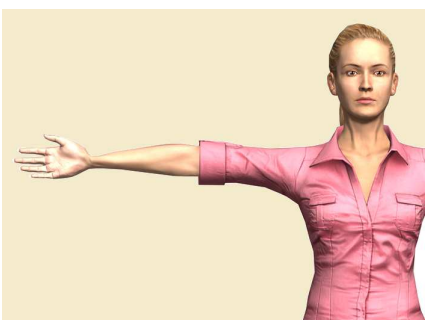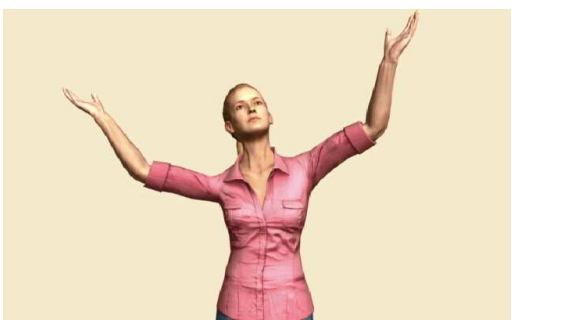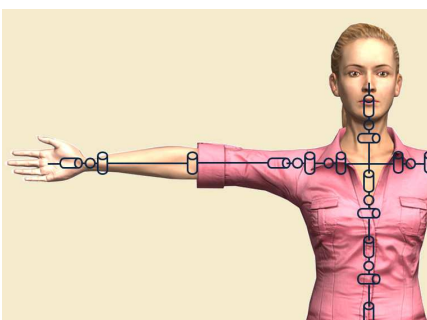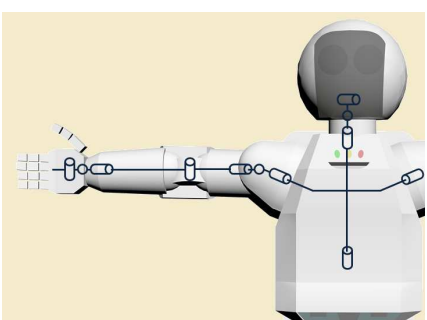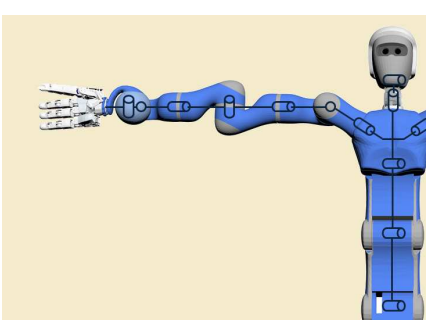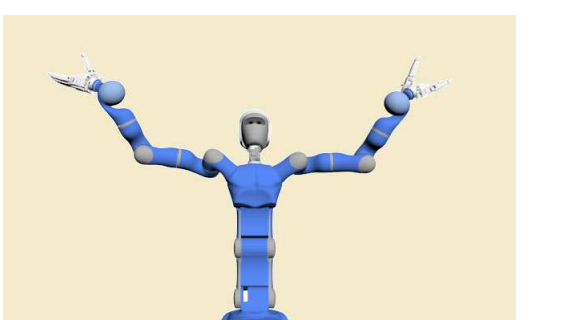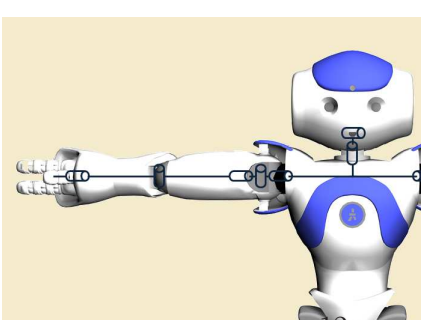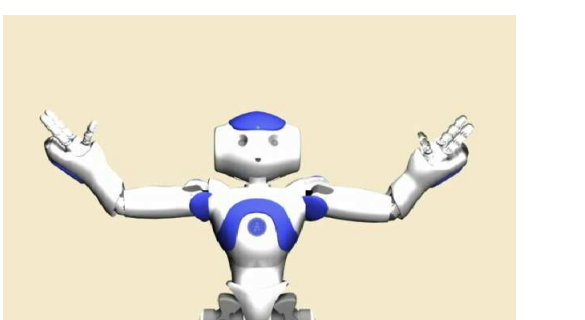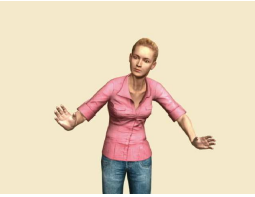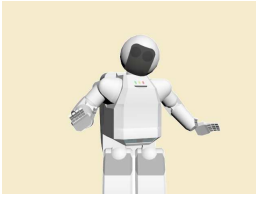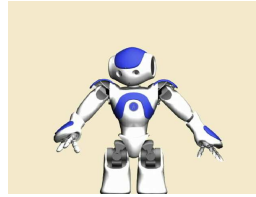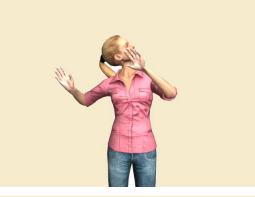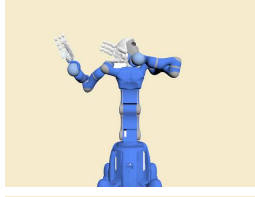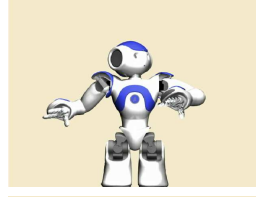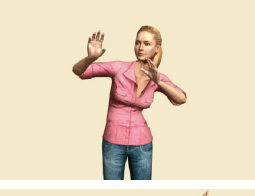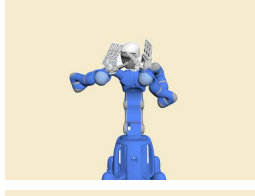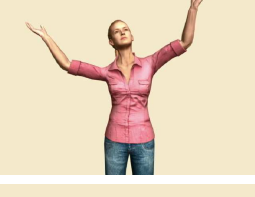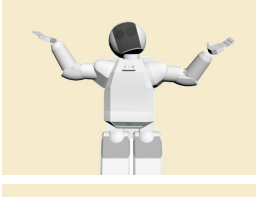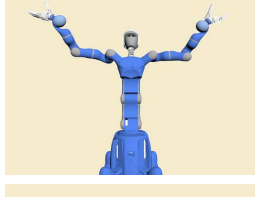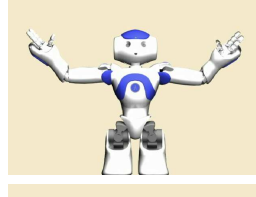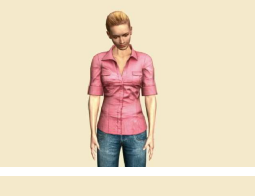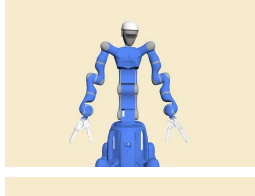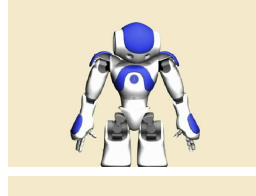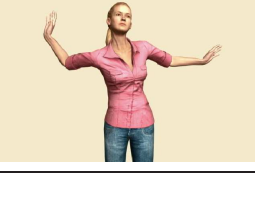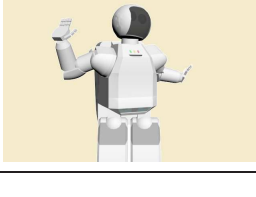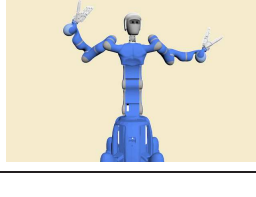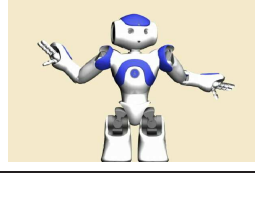
Configuration	Calculated posture
Base model	
Config 1: full configuration	
Config 2: ASIMO	
Config 3: Justin	
Config 4: NAO	

Table 4. End postures 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 mapped end posturea for the robots ASIMO, Justin and NAO.

	Target posture	ASIMO	Justin	NAO
Anger				
Disgust				
Fear				
Happiness				
Sadness				
Surprise				

6. Conclusions and future work

In this paper, a generic novel method for gesture generation for robots was described. Since the method is constructed independently of a robot configuration, it can be used by different robot platforms and models. By only inputting the Denavit-Hartenberg parameters of the configuration and a maximum of four rotation matrices, the method calculates the necessary joint angles for that configuration to establish a desired gesture. The method 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 them to a virtual model. Current work includes validating the method on the real robots instead of their virtual
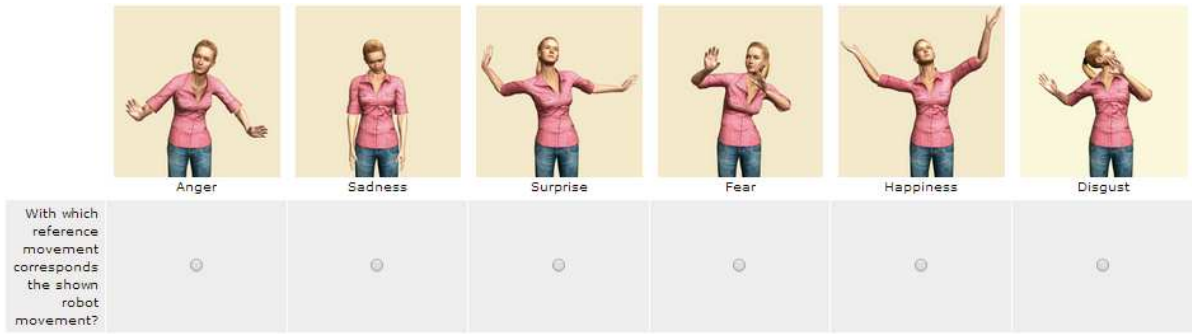
Figure 6.   The answer form used in the survey. The participant is asked to link a shown robot gesture with one of the target gestures previously shown in the survey. The end posture of every gesture is depicted together with its label.

Table 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

model (Figure 7). Working with the physical models requires the implementation of some specific features. Firstly, a collision avoidance module needs to be integrated. Until now, no precautions are made to prevent collisions, since while working with virtual models this is not a critical issue. While working with the real robots, however, this becomes an important topic and an existing self collision module will be implemented. Another topic that becomes important at this stage are speed related issues that may arise when using different robots with different speeds limits. The speed at which a gesture is performed depends on the emotional state that is to be conveyed by the robot. The method uses normalized speeds which are stored in the gesture database and will be scaled depending on the speed limits of a certain robot. Another interesting topic that can be explored is the extension of the method to hand postures, since adding an appropriate hand posture will have a positive influence on the recognition of the performed emotion. Finally, the gesture library can be extended with other gestures. Because of the easiness to implement new configurations, this method is perfectly suited to investigate the importance of different joints on the performance of gestures. By adding or removing certain joints, their effect on the performance can be easily visualized and as such, an optimal robot configuration for gesturing can be obtained.

7.   Acknowledgments
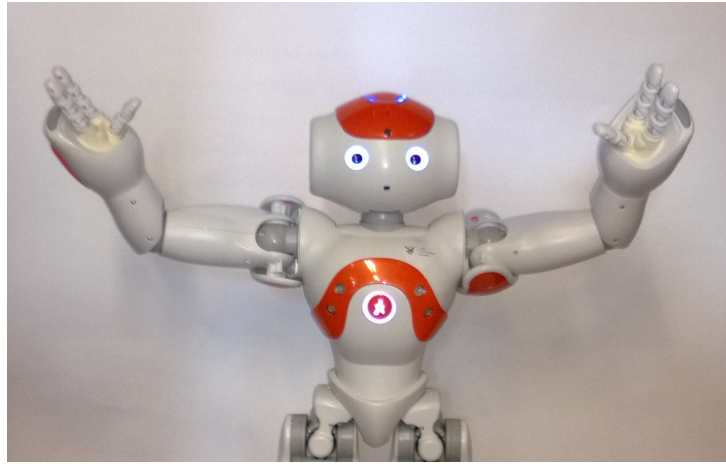
Figure 7. Current work includes validating the method on the real robots instead of their virtual model.

References

[1] Park E, Kim KJ, del Pobil AP. The effects of robot body gesture and gender in human-robot interaction. Human-Computer Interaction. 2011;6:91–96.

[2] Salem M, Eyssel F, Rohlfing K, Kopp S, Joublin F. To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. Int Journal of Social Robotics. 2013;:1–11.

[3] Itoh K, Miwa H, Matsumoto M, Zecca M, Takanobu H, Roccella S, Carrozza M, Dario P, Takanishi A. Various emotional expressions with emotion expression humanoid robot WE-4RII. In: Ieee technical exhibition based conference on robotics and automation. 2004 November. p. 35–36.

[4] Ishiguro H, Ono T, Imai M, Maeda T, Kanda T, Nakatsu R. Robovie: an interactive humanoid robot. Industrial robot: An international journal. 2001;28(6):498–504.

[5] Sugiyama O, Kanda T, Imai M, Ishiguro H, Hagita N. Natural deictic communication with humanoid robots. In: Iros 2007. 2007. p. 1441–1448.

[6] Ido J, Matsumoto Y, Ogasawara T, Nisimura R. Humanoid with interaction ability using vision and speech information. In: Ieee/rsj int. conf. on intelligent robots and systems,. 2006. p. 1316–1321.

[7] Zecca M, Mizoguchi Y, Endo K, Iida F, Kawabata Y, Endo N, Itoh K, Takanishi A. 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. 2009. p. 381–386.

[8] Matsui D, Minato T, MacDorman K, Ishiguro H. Generating natural motion in an android by mapping human motion. In: Iros 2005. 2005. p. 3301–3308.

[9] Tapus A, Peca A, Aly A, Pop C, Jisa L, Pintea S, Rusu AS, David DO. Children with autism social engagement in interaction with nao, an imitative robot. a series of single case experiments. Interaction studies. 2012;13(3):315–347.

[10] Do M, Azad P, Asfour T, Dillmann R. Imitation of human motion on a humanoid robot using non-linear optimization. In: 8th ieee-ras int. conf. on humanoid robots, humanoids 2008. 2008. p. 545–552.

[11] Dautenhahn K, Nehaniv CL. The correspondence problem. MIT Press. 2002.

[12] Alissandrakis A, Nehaniv CL, Dautenhahn K. Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on. 2002;32(4):482–496.

[13] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY. Ros: an open-source robot operating system. In: Icra workshop on open source software. Vol. 3. 2009.

[14] Bruyninckx H. Open robot control software: the OROCOS project. In: Ieee international conference on robotics and automation (icra 2001). Vol. 3. IEEE. 2001. p. 2523–2528.

[15] Waibel M, Beetz M, Civera J, D'Andrea R, Elfring J, Galvez-Lopez D, Haussermann K, Janssen R, Montiel J, Perzylo A, et al.. Roboearth. IEEE Robotics & Automation Magazine. 2011;18(2):69–82.

[16] Stanton C, Bogdanovych A, Ratanasena E. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In: Proc. australasian conference on robotics

and automation. 2012.

[17] Salem M, Kopp S, Wachsmuth I, Joublin F. 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.

[18] Le QA, Hanoune S, Pelachaud C. Design and implementation of an expressive gesture model for a humanoid robot. In: Humanoid robots (humanoids), 2011 11th ieee-ras international conference on. IEEE. 2011. p. 134–140.

[19] Salem M, Kopp S, Wachsmuth I, Joublin F. Towards meaningful robot gesture. In: Human centered robot systems. Springer. 2009. p. 173–182.

[20] Beck A, Cañamero L, Bard KA. Towards an affect space for robots to display emotional body language. In: Ro-man. IEEE. 2010. p. 464–469.

[21] Hild M, Siedel T, Benckendorff C, Thiele C, Spranger M. Myon, a new humanoid. In: Language grounding in robots. Springer. 2012. p. 25–44.

[22] Goris K, Saldien J, Vanderborght B, Lefeber D. Mechanical design of the huggable robot probo. International Journal of Humanoid Robotics. 2011;8(3):481–511.

[23] Tanaka F, Noda K, Sawada T, Fujita M. Associated emotion and its expression in an entertainment robot QRIO. Entertainment Computing–ICEC 2004. 2004;:265–298.

[24] Saponaro G, Bernardino A. Generation of meaningful robot expressions with active learning. In: Proceedings of the 6th international conference on human-robot interaction. ACM. 2011. p. 243–244.

[25] Atkinson AP, Dittrich WH, Gemmell AJ, Young AW, et al.. Emotion perception from dynamic and static body expressions in point-light and full-light displays. PERCEPTION-LONDON-. 2004; 33:717–746.

[26] Matarić MJ, Pomplun M. Fixation behavior in observation and imitation of human movement. Cognitive Brain Research. 1998;7(2):191–202.

[27] Ng-Thow-Hing V, Luo P, Okita S. Synchronized gesture and speech production for humanoid robots. In: Ieee/rsj international conference on intelligent robots and systems iros 2010. 2010. p. 4617–4624.

[28] Kleinsmith A, De Silva PR, Bianchi-Berthouze N. Cross-cultural differences in recognizing affect from body posture. Interacting with Computers. 2006;18(6):1371–1389.

[29] Ekman P, Friesen W. Facial Action Coding System. Consulting Psychologists Press. 1978.

[30] Kadaba MP, Ramakrishnan H, Wootten M. Measurement of lower extremity kinematics during level walking. Journal of Orthopaedic Research. 1990;8(3):383–392.

[31] Kozima H, Michalowski M, Nakagawa C. Keepon - a playful robot for research, therapy, and entertainment. International Journal of Social Robotics. 2008;1:3–18.

[32] Sciavicco L. Robotics: modelling, planning and control. Springer. 2009.

[33] Ascher UM, Petzold LR. Computer methods for ordinary differential equations and differential-algebraic equations. Vol. 61. Siam. 1998.

[34] Tsagarakis NG, Metta G, Sandini G, Vernon D, Beira R, Becchi F, Righetti L, Santos-Victor J, Ijspeert AJ, Carrozza MC, et al.. icub: the design and realization of an open humanoid platform for cognitive and neuroscience research. Advanced Robotics. 2007;21(10):1151–1175.

[35] Hirose R, Takenaka T. Development of the humanoid robot asimo. Honda R&D Technical Review. 2001;13(1):1–6.

[36] Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J, Maisonnier B. The nao humanoid: a combination of performance and affordability. CoRR abs/08073223. 2008;.

[37] http://www.rocketbox-libraries. com.

[38] Hirai K, Hirose M, Haikawa Y, Takenaka T. The development of honda humanoid robot. In: Ieee international conference on robotics and automation (icra 1998). Vol. 2. 1998 May. p. 1321–1326.

[39] Ott C, Eiberger O, Friedl W, Bauml B, Hillenbrand U, Borst C, Albu-Schaffer A, Brunner B, Hirschmuller H, Kielhofer S, et al.. A humanoid two-arm system for dexterous manipulation. In: Humanoid robots, 2006 6th ieee-ras international conference on. IEEE. 2006. p. 276–283.

[40] Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J, Maisonnier B. Mechatronic design of nao humanoid. In: Robotics and automation, 2009. icra'09. ieee international conference on. IEEE. 2009. p. 769–774.

[41] Breazeal C. Designing sociable robots. The MIT Press. 2004.

Enhancing My Keepon robot: A simple and low-cost solution for robot platform in Human-Robot Interaction studies

Hoang-Long Cao[1], Greet Van de Perre[1], Ramona Simut[2],
Cristina Pop[3], Andreea Peca[3], Dirk Lefeber[1], Bram Vanderborght[1]

Abstract— Many robots capable of performing social be-
haviors have recently been developed for Human-Robot In-
teraction (HRI) studies. These social robots are applied in
various domains such as education, entertainment, medicine,
and collaboration. Besides the undisputed advantages, a major
difficulty in HRI studies with social robots is that the robot
platforms are typically expensive and/or not open-source. It
burdens researchers to broaden experiments to a larger scale or
apply study results in practice. This paper describes a method
to modify My Keepon, a toy version of Keepon robot, to be a
programmable platform for HRI studies, especially for robot-
assisted therapies. With an Arduino microcontroller board and
an open-source Microsoft Visual C# software, users are able
to fully control the sounds and motions of My Keepon, and
configure the robot to the needs of their research. Peripherals
can be added for advanced studies (e.g., mouse, keyboard,
buttons, PlayStation2 console, Emotiv neuroheadset, Kinect).
Our psychological experiment results show that My Keepon
modification is a useful and low-cost platform for several HRI
studies.

I. INTRODUCTION

Human-Robot Interaction (HRI) is an emerging field
aimed at improving the interaction between human beings
and robots in various activities. Researchers in this field
are required to understand their research within a broader
context due to the interdisciplinary nature of HRI [1]. Ac-
cording to [2], HRI differs from human-computer interaction
and human-machine interaction because it concerns systems
which have complex, dynamic control systems, which exhibit
autonomy and cognition, and which operate in changing,
real-world environments. Traditionally, robots are operated
by experts and work as tools in industry. Many recent
commercial robot platforms have been developed with the
ability to exhibit social behaviors, and capability to work
with non-expert users at home, school, hospital, museum,
etc [3]. It is suggested that robots as partners can help us
accomplish more meaningful work and achieve better results
[2]. Hence, there is an observed niche market for social
robots.

A big challenge to broaden the scope of research and apply
the study results to daily life is the high cost of robot plat-
forms. Successful robots such as Asimo [4], Nao [5], iCub
[6], HRP-4C [7], Probo [8], Keepon Pro [9], are expensive

due to their complexity with high degrees of freedoms. On
the other hand, simpler robots like Furby [10], KASPAR
[11], Pleo [12], are cheaper but not officially modifiable
e.g. no SDK. Since many advanced studies need to connect
the research platform to peripherals (e.g., mouse, keyboard,
buttons, PlayStation2 console, Emotiv neuroheadset, Kinect,
Xbox 360 controller), the cheap robots mentioned above
prevent the possibility to adapt to the research requirements.
Therefore, there is a need to have low-cost and expandable
social robot platforms for HRI studies. Ono robot has re-
cently been developed which can be reproducible at the cost
of approximately €300 [13]. However, Ono is at the first
steps of development and its electronics needs to be improved
[13].

Modifying cheap commercial robots can be a solution to
have low-cost programmable platforms for HRI experiments.
In [14] and [15], Pleo robot is hacked by adding a bluetooth
interface with a complete tutorial given by LIREC project.
However, the instructions are complex and require a lot of
materials. Hasbro's Furby toy (2012 version) can be hacked
by using audio protocol with the official Furby applications
for iOS and Android [16]. This method is simple but cannot
enable Furby to connect with other devices. According to
[17], hacking Furby's hardware is probable by rewriting its
EEPROM but currently not practical due to the insufficient
understanding of the data structure. Also, the form of Furby's
casing with hidden screws makes the disassembling process
difficult. In [18], My Keepon robot is modified to be a
programmable robot platform. Its internal circuit board is
replaced by an Arduino microcontroller board and motor
drivers. Thus, the cost of modifying increases. Alternatively,
there exists a method to hack My Keepon by sending I^2C
commands to its microprocessors from Arduino. The source
code is supported by the manufacturer Beatbox [19], and
from Nonpolynomial Labs [20]. The hacking cost is low
since the internal electronics is retained. Moreover, the casing
of My Keepon can easily be opened to access the hardware.
To this end, My Keepon is a good candidate to be modified.

Although the source code to communicate with My
Keepon is available, there is a lack of instructions to build the
complementary hardware and user interface, and to adapt the
source code with external devices. In this paper, after a short
overview of Keepon Pro and My Keepon, we will present
the technical details to connect My Keepon to Arduino with
a user interface written in Microsoft Visual C#. Integrating
the modified platform with PlayStation2 console and Emo-
tiv neuroheadset will be introduced as a general example

[1]Vrije Universiteit Brussel, Robotics & Multibody Mechanics Research
Group, Pleinlaan 2, 1050 Brussels, Belgium
[2]Vrije Universiteit Brussel, Department of Clinical and Life Span Psy-
chology Group, Pleinlaan 2, 1050 Brussels, Belgium
[3]Babes-Bolyai University, Department of Clinical Psychology and Psy-
chotherapy, Cluj-Napoca, Romania
Hoang.Long.Cao@vub.ac.be

of expanding the modification for advanced studies. Then Section IV briefly describes our psychological experiments to demonstrate the performance of the modified My Keepon in several HRI studies.

II. KEEPON PRO AND THE HACKABLE MY KEEPON

Keepon Pro is a $30,000 research model developed by Hideki Kozima, and is sold commercially to Beatbots. It appears as a small yellow creature-like robot designed for simple, natural, nonverbal interaction with children to study social development of autistic children [9]. According to Beatbox website [21], Keepon Pro structure has four degrees of freedoms (DOFs): turning ($\pm 180°$), nodding ($\pm 40°$), rocking side-to-side ($\pm 25°$), and bouncing with a 15 mm stroke. A PID controller generates a trapezoidal velocity profile for each DOF. Keepon Pro's playroom perceptions are transmitted to a therapist by two cameras in its eyes and a microphone in its nose. A rubber skin ensures safe and comfortable contact between the hands and the hidden buttons at each side of the robot. Keepon Pro is an interesting robot platform and has been widely used in HRI studies on social development and behaviors (e.g. eye contact, joint attention, touching, caregiving, and imitation) [9], [18], [22], [23], [24], [25].

My Keepon is a low-cost version of Keepon Pro at the price of $40. It is designed as an interactive robotic toy for kids with two modes: Touch and Dance. In the Touch mode, My Keepon responds to pokes, pats, and tickles detected by hidden buttons with emotional movements and sounds. In the Dance mode, it detects beat in music and automatically generates movements in synchronized rhythm [21]. As shown in Figure 1, the external structures of Keepon Pro and My Keepon are basically identical. However, My Keepon has limited capabilities because of its simple internal structure.

Nonpolynomial Labs initiated the idea of hacking My Keepon with Arduino by reverse engineering. Based on this idea, we improved the firmware and developed a software to control My Keepon at the Vrije Universiteit Brussel (Belgium). Beatbox afterwards offered the official source code to make the robotic toy completely hackable. According to [20] and [26], My Keepon uses two microprocessors (PS232 and PS234) to control the movements and the sounds, which communicate with each other via I^{2}C protocol. The PS232 (Slave - address 0x52) deals with sounds and encoders. The PS234 (Master - address 0x55) handles driving the H-bridges, detecting button presses, and main processing. My Keepon can be controlled by sending commands to these two microprocessors over the I^{2}C bus.

III. MY KEEPON MODIFICATION

In this section, we briefly describe the procedure to turn My Keepon to be a programmable robot platform. Since 2012 before the official source code released, we started to provide a tutorial of hacking My Keepon which has been used in our experiments. Going further from the Arduino code, we have also offered an open-source software written
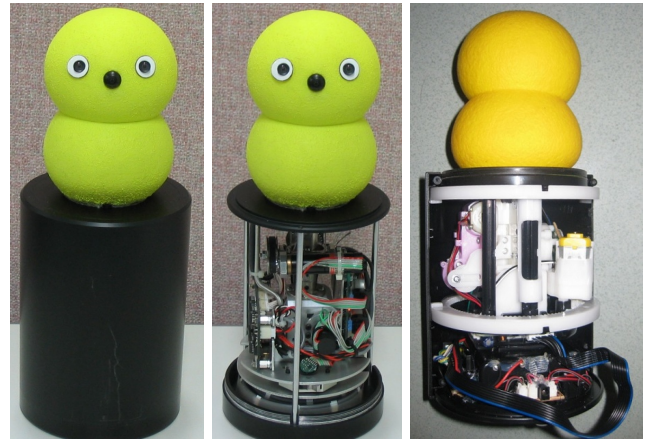
Fig. 1: Internal and external structures of Keepon Pro [9] and My Keepon.

(a)   (b)
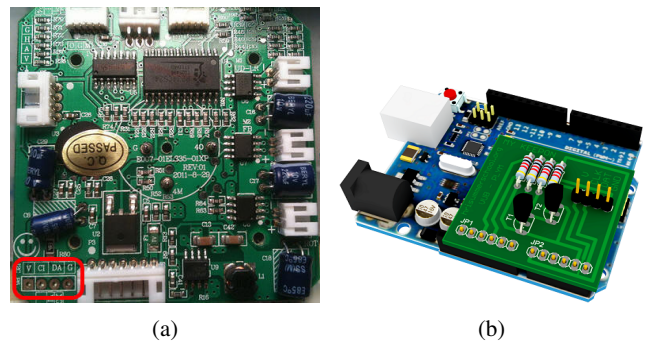
Fig. 2: My Keepon controller board (left) and Arduino shield for the modification (right).

in Microsoft Visual C# with a user-friendly Graphical User Interface (GUI) to control My Keepon. For more details of the hacking process, please visit our "Hacking Keepon" website at *http://probo.vub.ac.be/HackingKeepon* (previously at *http://vikeepon.tk*).

A. Hacking the electronics

As previously mentioned, My Keepon can be hacked by sending I^{2}C commands to its microprocessors from an Arduino microcontroller board. In order to do this, we need to connect the I^{2}C pins of My Keepon's controller board to Arduino. This board lies under the mechanism after opening the casing (Figure 2a). The I^{2}C pads (V, CI, DA and G) are on the top right corner of the control board, indicated by a smiley face. These pads need to be connected to the corresponding pins of Arduino board: V-A0, CL-A5, DA-A4 and G-GND (Figure 4a). Since My Keepon's controller board works at 3.3V, a logic level converter such as the dedicated shield in Figure 2b is required if using 5V Arduino (e.g, Uno, Leonardo). Drill a hole above the power connector in order to let the four wires enter the casing, and then re-assemble the robot carefully. The last step is to upload the firmware (.ino or .pde extension) to Arduino board. After that, we can fully control My Keepon by sending string commands from
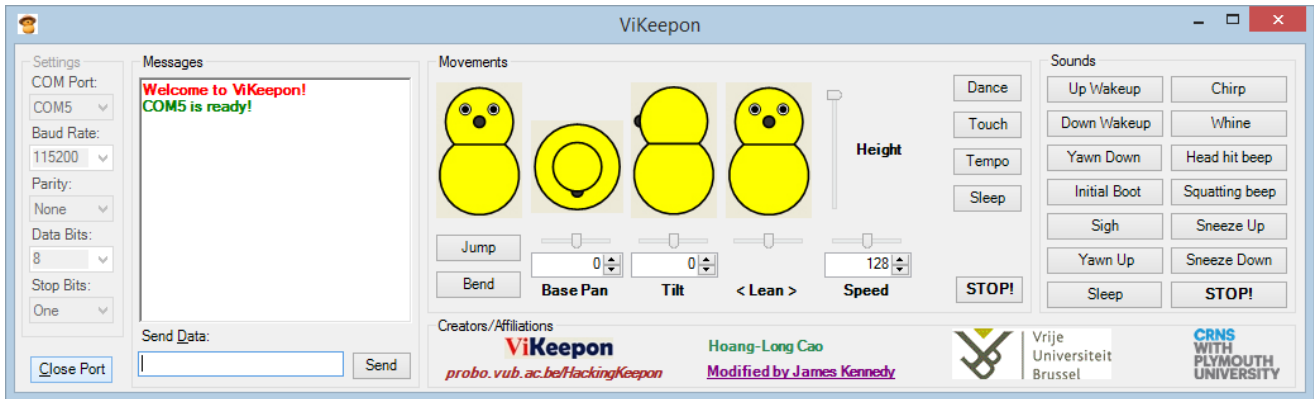
Fig. 3: Basic GUI to control the modified My Keepon. Adapted depending on specific experiments.

computer to Arduino via serial communication (UART). The list of commands with respect to Beatboxs firmware is given in Table I. Commands for Nonpolynomial Labs firmware are purely in hexadecimal values and not obviously understandable [20]. Arduino converts these string commands to the corresponding I^{2}C commands of My Keepon.

TABLE I: Commands and incoming data to control My Keepon [21].

Commands	Incoming data
SOUND (play, repeat, replay, stop)	BUTTON(built-in buttons)
SPEED (pan, tilt, ponside)	MOTOR (pan, tilt, pon, side)
MOVE (pan, tilt, pon, side, stop)	ENCODER (pan, tilt, pon, side)
MODE (dance, touch, tempo, sleep)	EMF (pan, tilt, ponside)
	POSITION (pan, tilt, ponside)
	AUDIO (tempo, mean, range,...)

Note that hacking My Keepon voids its warranty. Therefore, it is recommended to take care of the positions of the different parts to make sure My Keepon can be re-assembled. Apart from soldering wires to the I^{2}C pads, do not intervene other parts of the controller board. However, the hacking process is forthright and My Keepon can function at original condition if Arduino has no supply power.

B. Graphical user interface

Since most of non-engineering researchers (e.g. therapists, psychologists) are not familiar with I^{2}C commands, an easy-to-use software written in Visual C# provides a more convenient way to control My Keepon. This software is available to download from our website. Through a set of buttons of the GUI as shown in Figure 3, users are able to control all movements and built-in sounds of My Keepon. The software can also receive data from encoders and detect button presses. Functions of GUI components are as follows.

- **Settings:** configuration of serial port (e.g., port number, baud rate) and Close/Open button
- **Messages**: display outgoing (sound and motion commands) and incoming data (encoders, button presses)
- **Movements:** two buttons and four numeric track-bars with up-down boxes to change the movements

- **Mode buttons:** start Dance, Touch, Tempo, Sleep mode
- **Sounds:** buttons corresponding to different sounds generated by My Keepon's hardware

It is important to follow the following steps in order to make the modified My Keepon platform operate properly.

1) Connect the Arduino to My Keepon
2) Connect the Arduino to a computer using a USB cable
3) Launch the software (.exe file)
4) Power up My Keepon
5) Open serial communication (Baud rate: 115200, Parity: None, Data bits: 8, Stop bits: 1)

The software usability is intuitive. When users click on a button e.g. "Jump", the software converts it into a string command and then transmits to Arduino via a serial communication. Thanks to the firmware, My Keepon performs the corresponding movement or sound which is "bouncing" in this case. During this process, commands and incoming data from My Keepon are displayed in "Messages" groupbox for debugging purpose.

Basically, it is sufficient to set up experiments with manual or teleoperated mode by using the GUI components. With the Microsoft Visual Studio programming environment, advanced users can go further into the source code to create their desired functions such as generating complex movements or playing their own sounds. To do this, the basics of the communication protocol mentioned above must be understood. The software is compatible with Windows 8 (32-/64-bit), Windows 7 (32-/64-bit), and Windows XP.

C. Expanding the platform

In some HRI studies, it is necessary to integrate other devices to the modified My Keepon platform. This requires users to have knowledge of electronics and programming. The big resources from Arduino and Visual C# communities provide tremendous possibilities to expand the hardware and improve the software.

Devices are connected to Arduino or computer depending on their specifications. Simple devices (e.g., LEDs, buttons, keypad, joystick, PlayStation2 console) are typically connected to Arduino by virtue of standards and contributed libraries. On the other hand, advanced devices (e.g., Kinect,
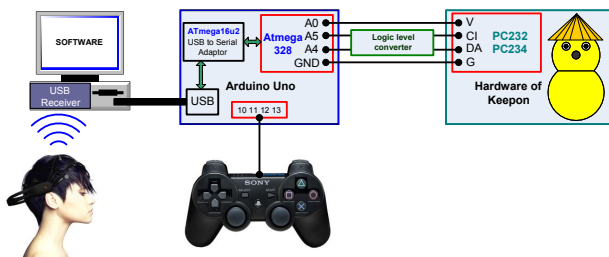
(a) Hardware connections.

(b) User testing.

Fig. 4: An example of expanding the modified My Keepon platform with PlayStation2 console and Emotiv system.

Fig. 5: "Hacking Keepon" workshop at the 2013 International Summer School on Social HRI.

Emotiv neuroheadset, Xbox 360 controller) need to be connected to the computer and communicate with the Visual C# software via SDKs.

We present here a general example to demonstrate the possibility to expand the modified My Keepon platform in terms of both hardware and software. Figure 4a illustrates the expanded platform in which a PlayStation2 console and Emotiv system are connected with Arduino and a computer, respectively. Therefore, Arduino firmware and Visual C# software have to be modified.

1) PlayStation2 console: PlayStation2 console is realized to be a handy tool to control My Keepon. Since it is widely used in DIY projects, tutorials of interfacing the console and Arduino are easily found on the Internet. The hardware connection is set up by connecting six wires of the console (clock, data, command, power, ground, and attention) to Arduino pins. The codes to handle button presses from the console need to be put inside the main loop of the Arduino firmware. The button press events are then converted to I^{2}C commands of My Keepon.

2) Emotiv system: Getting inspired from the recent studies on brain-machine interface (e.g., [27], [28], [29]), Emotiv system is integrated to control My Keepon by thought. The system consists of a neuroheadset with electrodes to measure brain signals, and a USB dongle to communicate with computer wirelessly. In order to read brain signals, the Visual C# software includes Emotiv API which is exposed as an ANSI C interface implemented in two Windows DLLs ($edk.dll$ and $edk\_utils.dll$) [30]. Thanks to this API, raw signals are translated into users' thought in form of EmoState structure such as forward, backward, rotate left, rotate right, etc. Emostate is then converted into string commands to

control My Keepon.

In this experiment setting, users are able to control My Keepon by a PlayStation2 console or Emotiv neuroheadset as can be seen in Figure 4b. The performance of user's thought detection significantly depends on the training process.

IV. APPLICATIONS IN HRI STUDIES

The modified My Keepon platform to some extent can achieve similar performances as of Keepon Pro in HRI studies. The modification method is simple and does not require advanced knowledge of electronics and programming. We organized a workshop of hacking Keepon at The 2013 International Summer School on Social HRI for a multidisciplinary group of students such as engineers, computer scientists, psychologists, etc (see Figure 5). Even though many of them lack technical experience of soldering and programming, they were able to modify My Keepon in two hours. The workshop result proved that researchers can easily be familiar with the software, complementary Arduino shield and expand the platform. Hence, they can quickly set up the platform for their experiments.

Since 2012 at the Vrije Universiteit Brussel (Belgium) and Babes-Bolyai University (Romania), we have conducted three psychological experiments with the modified My Keepon with typically developing children and autistic children. In these experiments, we expand the hardware by adding simple devices such as buttons, LEDs, a buzzer to Arduino. The Visual C# GUI is extended so that the operator can easily control the actions of the robot depending on the requested scenario for the intervention. Specifically, buttons are created to make specific combination of movements (e.g, $45°$ left/right turns, nodding or bouncing within a period of time) by using string commands, and to play sound files. Another button is additionally made to turn My Keepon back to its neutral position.

A. Joint attention

A pilot-study was conducted to investigate the performances of typically developing infants in tasks focused on joint attention. Joint attention refers to a set of behaviors that serve to enable two partners to either vocally or non-vocally communicate about, or "jointly attend to" a third entity, object, or event [31]. Two types of joint attention behaviors were targeted in this pilot study, such as gaze-following and initiations of joint attention, during the infants interaction
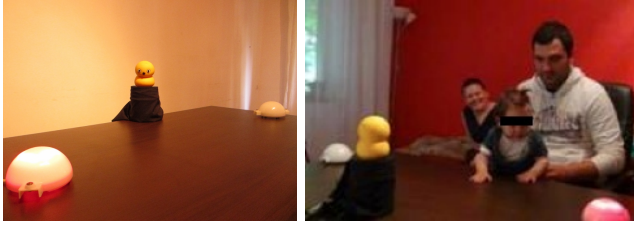
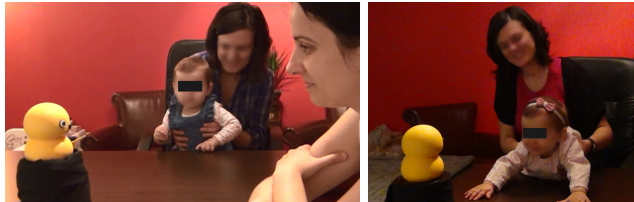Fig. 6: Joint attention study with the modified My Keepon.

Fig. 7: Social agent study with the modified My Keepon.

with a robot (My Keepon) and a human partner. The extended platform for this experiment is shown in Figure 6. A pair of attractive objects was developed containing bright LEDs (red and blue) with an opaque white plastic cover. Every time when the target object was activated by the operator a pre-programmed light animation was started, together with a monotonic melody. The result of this study is beneficial for joint attention interventions for children who are impaired in this ability e.g. autistic children.

B. My Keepon as a social agent

The second experiment used My Keepon to investigate if infants perceive an unfamiliar agent as a social agent after observing an interaction between the robot and an adult. Twenty-three infants, aged 9-17 month, were exposed, in a first phase, to either a contingent or a noncontingent interaction between the two agents, followed by a second phase, in which the children were offered the opportunity to initiate a turn-taking interaction with My Keepon. The measured variables were: (1) mean looking time to each of the two interacting partners, (2) the number of anticipatory orientations of attention toward the agent that follows in the conversation, and (3) the number of verbal and motor initiations of the child toward the robot. A snapshot of the experimental setting is presented in Figure 7. In the contingent condition, My Keepon responds contingently, by a pre-programmed set of sounds and motions to the adult verbal initiations, while in the noncontingent condition, My Keepon remains still. In the testing phase, My Keepon responds by the same pre-programmed set of sounds and motions to any intentional babbling or motion of the child. At the moment, the data collected are still under analyses. The results will indicate if children, before their second birthday, attribute intentions to unknown agents, based on the dynamics of their interaction.

C. Role of My Keepon in a cognitive flexibility task

Another experiment was performed to investigate the role of My Keepon in a cognitive flexibility task performed by children with autism and typically developing children. The number of participants included in this study was 81 children: 40 typically developing children aged 4-7 years and 41 children with autism aged 4-13 years. Each participant had to go through two conditions: robot interaction and human interaction. In both conditions, they had a reversal learning task, meaning that they had learned to choose the correct stimulus location from a pair of locations to receive a positive feedback from My Keepon or from the human (acquisition). After making the correct choice over multiple trials, the rewarded stimulus location changed without warning (reversal). We have measured the number of errors from acquisition phase and from reversal phase, as primary outcomes, and shared attention and positive affect, as secondary outcomes. Even though data are still being analyzed, we expect that the children with autism will have better performance in the robot condition compared with human condition.

V. Conclusion

Taking the body of this paper as a whole, we present a method to modify My Keepon to be a programmable research platform for HRI studies. Nonpolynomial Labs initialized this idea by sending I^2C commands to the controller board of My Keepon from an Arduino microcontroller board. After that, Beatboxs supported the official firmware with a full set of commands. However, the use of this firmware is not intuitive for non-engineering researchers. Since 2012, we offered an open-source Microsoft Visual C# software to control My Keepon by a GUI. Our website gives a complete tutorial with instructions for hacking the electronics and guidelines for software usage. Users are welcomed to modify the source code or integrate devices to fulfill their research needs or educational purposes. Our psychological experiments are used as examples of using the modified My Keepon in HRI studies. This work is expected to solve the current problem in HRI studies, i.e., the lack of low-cost robot platforms to enlarge the experiment scale or popularize the research results in society.

Future work includes making the modified My Keepon platform compatible with the Robot Operating System (ROS) software framework. With the advantages of ROS, developing software for robot will be easier thanks to ROS tools and libraries, as well as code sharing among researchers in the community.

REFERENCES

[1] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[2] T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, human-robot interaction," in *Robotics Research*. Springer, 2003, pp. 255–266.

[3] P. Salvine, M. Nicolescu, and H. Ishiguro, "Benefits of human-robot interaction [TC Spotlight]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 98–99, 2011.

[4] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2478–2483.

[5] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 769–774.

[6] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. J. Ijspeert, M. C. Carrozza, *et al.*, "iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research," *Advanced Robotics*, vol. 21, no. 10, pp. 1151–1175, 2007.

[7] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka, and S. Kajita, "Cybernetic human HRP-4C," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, 2009, pp. 7–14.

[8] K. Goris, J. Saldien, B. Vanderborght, and D. Lefeber, "Mechanical design of the huggable robot Probo," *International Journal of Humanoid Robotics*, vol. 8, no. 03, pp. 481–511, 2011.

[9] H. Kozima, M. P. Michalowski, and C. Nakagawa, "Keepon," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 3–18, 2009.

[10] Hasbro, "Furby," http://www.furby.com.

[11] K. Dautenhahn, C. L. Nehaniv, M. L. Walters, B. Robins, H. Kose-Bagci, N. A. Mirza, and M. Blow, "KASPAR–a minimally expressive humanoid robot for human–robot interaction research," *Applied Bionics and Biomechanics*, vol. 6, no. 3-4, pp. 369–397, 2009.

[12] E. S. Kim, D. Leyzberg, K. M. Tsui, and B. Scassellati, "How people talk when teaching a robot," in *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*. IEEE, 2009, pp. 23–30.

[13] C. Vandevelde, J. Saldien, M.-C. Ciocci, and B. Vanderborght, "Systems overview of Ono," in *Social Robotics*. Springer, 2013, pp. 311–320.

[14] J. Gregory, A. Howard, and C. Boonthum-Denecke, "Wii nunchuk controlled dance pleo! dance! to assist children with cerebral palsy by play therapy." in *FLAIRS Conference*, 2012.

[15] J. Dimas, J. Leite, A. Pereira, P. Cuba, R. Prada, and A. Paiva, "Pervasive pleo: long-term attachment with artificial pets," in *Mobile HCI*, 2010.

[16] Iafan, "Hacksby," https://github.com/iafan/Hacksby.

[17] Michael Coppola's Blog, "Reverse Engineering a Furby," http://poppopret.org/2013/12/18/reverse-engineering-a-furby/.

[18] H. Admoni, B. Hayes, D. Feil-Seifer, D. Ullman, and B. Scassellati, "Are you looking at me?: perception of robot attention is mediated by gaze type and group size," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 389–396.

[19] Beatboxs, "MyKeepon," https://github.com/BeatBots/MyKeepon/.

[20] Nonpolynomial Labs, "Keepon hacking - proof of concept," http://www.nonpolynomial.com.

[21] Beatboxs, "My Keepon," http://beatbots.net.

[22] H. Kozima, C. Nakagawa, and H. Yano, "Attention coupling as a prerequisite for social interaction," in *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on*. IEEE, 2003, pp. 109–114.

[23] H. Kozima, C. Nakagawa, and Y. Yasuda, "Children–robot interaction: a pilot study in autism therapy," *Progress in Brain Research*, vol. 164, pp. 385–400, 2007.

[24] E. Bernier and B. Scassellati, "The similarity-attraction effect in human-robot interaction," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, 2010, pp. 286–290.

[25] H. Admoni, C. Bank, J. Tan, M. Toneva, and B. Scassellati, "Robot gaze does not reflexively cue human attention," in *Proceedings of the 33rd Annual Conference of the Cognitive Science Society, Boston, MA, USA*, 2011, pp. 1983–1988.

[26] M. G. Marek Michalowski, Kyle Machulis and T. Hersan, "Hack "My Keepon" with an Arduino brain," http://makezine.com/projects/make-35/my-franken-keepon/.

[27] M. Rosen, "Mind to motion: Brain-computer interfaces promise new freedom for the paralyzed and immobile," *Science News*, vol. 184, no. 10, pp. 22–26, 2013.

[28] G. Ranky and S. Adamovich, "Analysis of a commercial EEG device for the control of a robot arm," in *Bioengineering Conference, Proceedings of the 2010 IEEE 36th Annual Northeast*. IEEE, 2010, pp. 1–2.

[29] H. Nisar, Q. Yeoh, H. Balasubramanium, W. Wei, and A. S. Malik, "Analysis of brain activity while performing cognitive actions to control a car," in *The 15th International Conference on Biomedical Engineering*. Springer, 2014, pp. 947–950.

[30] Emotiv, *Emotiv Software Development Kit - User Manual*.

[31] P. Mundy and A. Gomes, "Individual differences in joint attention skill development in the second year," *Infant behavior and development*, vol. 21, no. 3, pp. 469–482, 1998.

Enhancing Emotional Facial Expressiveness on NAO

A Case Study Using Pluggable Eyebrows

Albert De Beir · Hoang-Long Cao · Pablo Gómez Esteban · Greet Van de Perre · Dirk Lefeber · Bram Vanderborght

Abstract Robots can express emotions for better Human Robot Interaction. In this field, NAO robot is a platform widely used. This robot mainly expresses emotions by gestures and colored LED eyes, but, due to its white flat and inanimate face, the robot cannot express facial expressions. This work proposes a pluggable eyebrows device allowing NAO to express anger or sadness while performing other tasks. This device is plug-and-play and can be controlled directly by NAO's main software. Additionally we develop a platform independent mapping of colors and eyebrows angles with emotions. We first conducted an experiment that qualitatively attests the interest of this device. Three following experiments were conducted to: 1) Confirm the relation between eyebrows angle and expressed emotion; 2) evaluate different shapes in order to select the most appropriate one; 3) prove that NAO is able to use the eyebrows to express emotions while performing non emotional tasks.

1 Introduction

The capability of recognizing and expressing emotions is an important feature in human-robot communication. Within the field of social robots, and especially in contexts of health care or education, robots should

A. De Beir (✉)
Vrije Universiteit Brussel, Robotics & Multibody Mechanics Research Group, Pleinlaan 2, 1050 Brussels, Belgium
E-mail: albert@debeir.com
Website: http://www.dream2020.eu
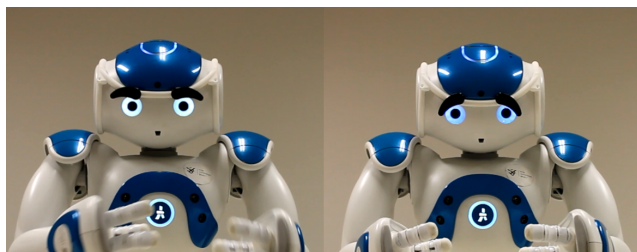
Fig. 1: Demonstration of the NAO eyebrows. These two picture are screenshots from the demo video available on: https://youtu.be/EuKMrGNTtog.

be able to engage with people on an emotional level, expressing emotions in a certain degree [19].

Facial expressiveness is considered as of great importance in building and maintaining social relationships together with facial and head micro movements [5, 6]. According to Cole [4] the face plays a crucial role in the expression of character and identity. Mehrabian [11] showed that facial expressions is the major modality in human face-to-face communication (*55%* of affective information is transferred this way, *38%* by paralanguage and only *7%* is transferred by spoken language).

Some social robots are only able to show a discrete set of facial expressions or move abruptly and unnaturally, in contrast to the smooth, elegant motion displayed by humans and animals [18]. Movements of the facial degrees of freedom (DOFs) associated to emotions of these robots are hard-coded and platform-dependent. A recent trend in social robotics focuses on platform-independent implementation is which robot's movements are coded using parameters. Consequently, movements of a certain robot can be transferred to another without reprogramming. Van de Perre et al.

[12] developed a generic method to generate upper-body emotional expressions for different social robots. A number of social robots use valence-arousal emotional space based on the circumplex model of affect defined by Russell [16] to select emotions [14] e.g. [10, 17].

1.1 Expressing emotion with NAO

The NAO robot, from Aldebaran Robotics, is a widely used robot in human-robot interaction (HRI) studies. As NAO has no means for facial expression except the color of its eye LEDs, some researchers have designed gestures to express emotional states [3], where they studied the impact of the head position on the identification of displayed emotions; or [7], which created different designs using body movements, sounds and eye colors; or [8], which used the eye LEDs to express 6 different emotions. In particular, in the work of [7], they created and evaluated emotion expression of NAO robot by a combination of fixed movement and sound, and fixed eye colors for discrete emotions i.e. red for anger, dark violet for sadness, bright yellow for joy, dark green for fear. In another study, Johnson et al. [8] also decides fixed colors for discrete emotions i.e. red for anger, yellow for surprise, green for disgust, blue for sadness, orange for happiness, cyan for fear.

Since NAO robot does not have enough DOFs in its head, it makes difficult to use facial expressions to replicate more sophisticated scenarios where other robots have been used, as is the case of Probo, with 20 degrees of freedom in its head, the robot is capable of expressing several emotions [17]; or the robot FACE, with 32 degrees of freedom, it has mapped the major facial muscles to simulate realistic facial expressions [13]. Today, the lack of emotion expression of NAO is one of the main limitation for social interaction and studies. Indeed, the robot is unable to perform an action (for example pointing) while performing a pose to express an emotion. Similarly, because the library proposed by Haring et al.[7] perform sounds during the poses, the robot is also unable to express emotion while interacting verbally. In this paper we want to propose a novel method to provide emotional feedback while playing the game, winning precious time and providing more realistic HRI.

Throughout this paper we aim at enhancing the facial expressivity of the NAO robot using a 3D printed pluggable mechanism that emulates the robot's eyebrows. To our knowledge, it is the first time a device is proposed to enhance NAOs emotion. This paper also
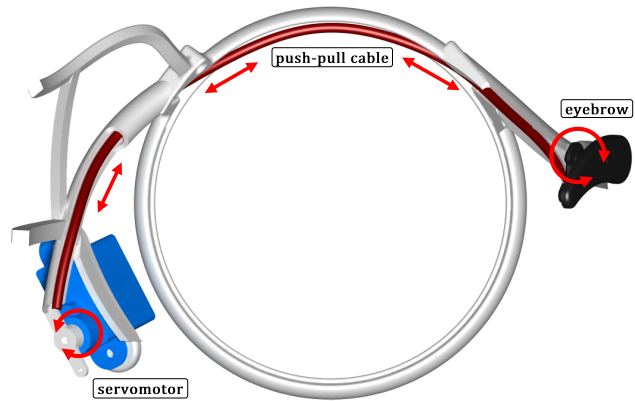
Fig. 2: Actuation of the eyebrows. The arrows show the conversion between rotation and translation. The white structure is 3D printed and can be cliped on the robot's face.

proposes a general method for platform independent social robots, to express emotions using eyebrow and eye colors.

In the first part of the method, we describe the mapping and parameters we use for eyebrows and color emotion expression. The second part of the method proposes an extensive description of the eyebrow setup focusing on the mechanical design, the electronics, and its integration using Aldebaran's Choregraphe software.

We then propose four experiments on the eyebrows. The first experiment conducts an open survey to asset the eyebrows emotion recognition. The second experiment describes the relation between the eyebrows angle and the expressed emotion. The third experiment evaluates different eyebrow shapes to find the most suitable one for social interaction. Finally the fourth experiment proves that, using the eyebrows, the NAO robot is able to express emotion while performing other tasks.

2 Method

2.1 Design of the eyebrows

2.1.1 Mechanical design

The main difficulty in designing such a device for NAO comes from the lack of space available for the actuators as the device should not modify to much the robot's appearance. Therefore, we propose a design where two micro servo-motors are placed at the back of NAOs head supported by a 3-D printed structure in Acrylonitrile butadiene styrene (ABS) which is clipped around the head. The torque needed to move the eyebrows is transmitted from the back to the front of the head through

a rigid cable. This cable is sufficiently rigid to act as a push-pull mechanism. As illustrated in Figure 2, the rotation of the servo is converted in translation of the cable (cable in red). At the front of the head, this translation is converted back in rotation of a pulley behind the eyebrow. The eyebrow is directly clipped on this pulley, allowing to easily change the shape of the eyebrows (as for example in Figures 7a to 7f). The micro servo-motor actuating the eyebrow is controlled by an Arduino-based board. As the two micro servos and the board have small power consumption, the board can be directly connected to the NAO robot through its USB port (see Figure 3a) at the back of its head without any additional power supply.

The eyebrows solution is simple and easy to use. Indeed, this device does not require any screw or glue and can be directly clipped in a few seconds on NAO's head without affecting the robot's hardware. Consequently, there is no risk of loosing the warranty as no modification of the robot are required. Finally, when not needed, the eybrows device can be unclipped and removed the same way they are clipped, in few a seconds.

2.1.2 Electronics

NAO robot has a USB port behind the head of the robot opening a possibility to connect the robot with external hardware which is the NAO eyebrows system consisting of two servo motors in this case. Figure 3a illustrates how NAO eyebrows system connects with NAO robot. An Arduino-based PCB acts as a bridge to transfer commands from the robot to PWM values in order to control the two motors. The Arduino-based PCB is designed with the dimension of 46mm x 15mm with a USB connector and headers to connect two servo motors as depicted in Figure 3a and 3b.

A firmware is uploaded to the ATmega328P microcontroller of the PCB which manages the data received from NAO robot via the USB communication. The data is then translated into PWM values to control the positions of two servo motors. The firmware is programmed using `serial` and `servo` libraries of Arduino. The PCB sketches as well as the Arduino code have been made available on Github: `https://github.com/hoanglongcao/ArduiNao-RMM`.

2.1.3 Choregraphe

In order to control two DOFs of NAO eyebrows from Choregraphe, we created a box called *NAO Eyebrows* as shown in Figure 4. The functionalities of this box are to setup a serial communication between NAO and
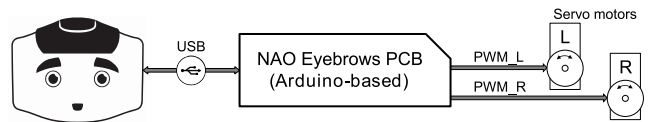
(a) An Arduino-based PCB is connected to the USB port of NAO robot to control two servo motors. Each servo motor can be controlled separately.

(b) The NAO eyebrows PCB is plugged into the USB port behind the head of the robot.

Fig. 3: The design of NAO eyebrows PCB. Its small dimension allows an embedded plug and play solution. Additionally, servos are directly powered by the USB port, avoiding the necessity of an external power supply.

the Arduino-based PCB, and to send the desired positions of the two eyebrows to the mechanical system. The serial communication is configured with the following parameters: Port name=`/dev/ttyUSB0`, Baud rate=`115200`, Data bits=`8`, Parity=`None`, Stop bits=`1`. Users are allowed to change these parameters, however, they have to change the corresponding parameters of the Arduino firmware. The box is programmed in Python using the `serialtools` library with two integer inputs for the two positions of NAO eyebrows. Once the input values are received, the data are processed and then sent to the PCB in formatted data i.e.
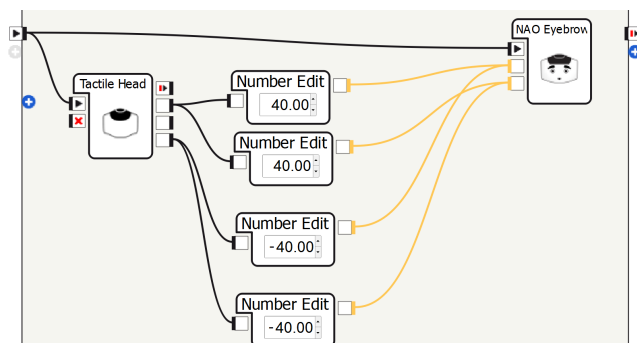
Fig. 4: An example of using tactile head sensors to control two movements of the eyebrows in Choregraphe. On the top right the "NAO Eyebrows" box allows users to easily control the positions of the NAO eyebrows.

(a) angry pose   (b) angry pose with eyebrows   (c) sad pose   (d) sad pose with eyebrows
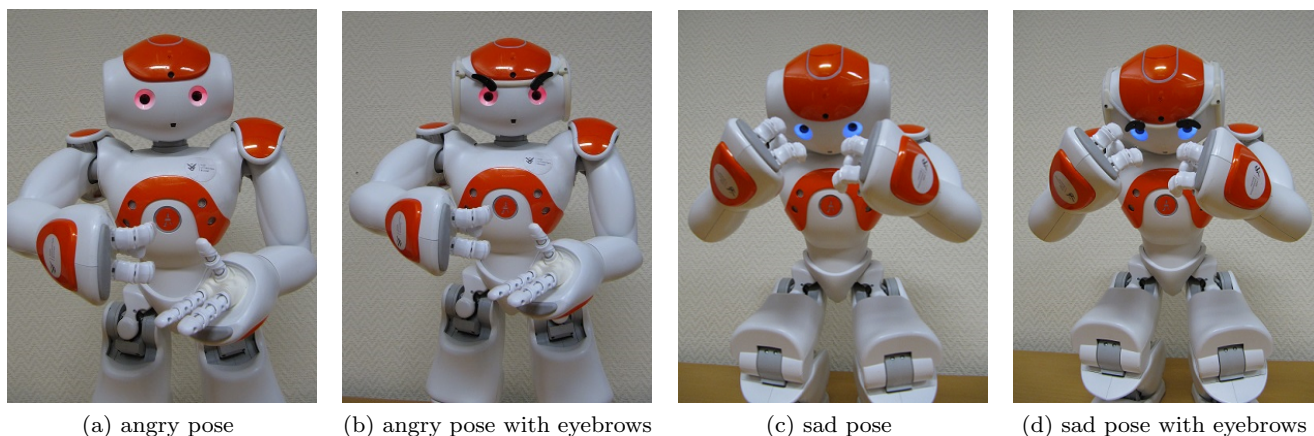
Fig. 5: Pictures used in the first experiment. For each picture, participants had to guess the robot emotion. We used poses expressing emotions from [7]. For each pose, the robot was either with or without eyebrows.

["positionL", int] and ["positionR", int]. Figure 4 presents an example of toggling the eyebrows between two positions (-40° and 40°). For users using NAOqi SDKs e.g. Python, C++, Java, the same process is required to communicate between NAO and the NAO eyebrows system.

An online video https://youtu.be/EuKMrGNTtog illustrating how easy it is to use the device. In this video, we show how the eyebrow can be directly clipped on Nao face without damaging the robot. We also perform a demo where the eyebrows and eyes' color are synchronized with the others actions of the robot using Choregraphe.

3 Experiments

Experiments are organized to validate the eyebrows setup. We hypothesize that: 1) the eyebrows are able to express emotion of anger and sadness. 2) The relation between eyebrows angle and expressed emotion can be approximate by a linear relation. 3) The size and the shape of the eyebrows will influence its likeability. 4) Using the eyebrows, the Nao robot can express emotions while performing another non-expressive task. Consequently, four independent experiments are conducted. The first experiment is an exploratory questionnaire with open questions to ensure that the eyebrows are conveying the emotions. The second experiment focuses on the relation between the output angles of the eyebrows and the robot's emotion. The third one investigates different eyebrows designs and evaluates the most appropriate one. The last experiment demonstrates the interest of the eyebrows to express emotions while the robot performs certain tasks.

3.1 Experiment 1

The first experiment is interested in the emotion that the robot conveys and the improvements that can be achieved by using the eyebrows. Therefore, we compare a conditions of the Nao robot expressing emotion with or without the eyebrows. We hypothesize that the recognition rate of *anger* and *sadness* emotions as defined by [6] will be higher using this eyebrows device than without it.

Procedure

To assess the functionality of this device, an online questionnaire has been filled in by 70 voluntary participants (23 were rejected because they did not answer all questions). All the participants belong to the 3rd year of a bachelor degree in psychology, as such they had no prior experience in robots. This exploratory questionnaire was made using LimeSurvey [9] and contained eight open questions. Participants were randomly split in two groups, one control group (without eyebrows), and one group with eyebrows. For each group, eight pictures of Nao expressing emotions were presented in a random order. For both groups, pictures contained emotional expressions from the literature: four pictures using body language [7] (2 for anger and 2 for sadness), two pictures using eyes colors (1 for anger and 1 for sadness) and two being neutral. Figure 5 shows an example of pictures used in the study. For each picture, the participant had to write in the questionnaire the emotion that was, according to him, expressed by the Nao robot. Participants had to guess the robot emotions as none of them were suggested during the survey. Consequently,

participants' answers were not influenced by pre-defined choice. Finally, participants were encouraged to answer "I do not know" if it was the case. This was done to ensure that the participants were not answering randomly.

Results and discussion

Participants were randomly separated in two groups and rejection of incomplete answer led to N=21 for the group **without** eyebrows and N=26 for the group **with** eyebrows. A content analysis was performed on the participant answers. Each of them was classified in one of the following categories: the six basic Ekman's emotions: *anger, disgust, fear, happiness, sadness,* and *surprise, others* for the emotions that did not fit the previous categories, or *no anwer* if the participant answered "I do not know". Responses were evaluated by three independent raters and were considered as correct when they had a meaning similar or close to the targeted emotion. With R [15], the inter-rater reliability for the raters was found to be almost perfect, *Kappa* = 0.84 (p=0). The results revealed that the rate of recognition is greatly improved by the eyebrows device. In fact, the recognition rate of sadness increased by *32.7%* (*5.8%* without eyebrows to *38.5%* with eyebrows). More impressively, the recognition rate of anger was improved by *80.6%* (*14.2%* without eyebrows to *94.8%* with eyebrows). Pre-defined choice answer would probably give even higher recognition rates. It should be noted that we did not perform a qualitative analysis on the neutrals picture because the recognition rate was too low (only one participant). In their study, Haring et al. [7] expressed emotions using sequences of body movements and sound. In this experiment, only pictures representing one body movement where used, suggesting why the pictures without eyebrows had such low recognition rate. This first experiment shows that it is therefore possible to express emotions using only the eyebrows.

3.2 Experiment 2

This experiment focuses on the relation between the eyebrow's angle and the corresponding expressed emotion. Moreover, we are also interested in the eyebrows' neutral position. Indeed, the eyebrows allows NAO to express emotion, but it is very important that it can also not express emotion, that is being neutral. This study aims to confirm the angle at which no emotions are carried with the eyebrows. In consequence we hypothesize that: 1) the relation between expresses emotions and eyebrow angle can be approximate by a linear

regression; 2) at an angle of 0° the eyebrows do not express emotions (neutral).

Procedure

For this study, 40 participants (11 women and 29 men) with a mean age of 29.02 (SD=12.17) we recruited using Prolific Academic website [1]. To access the study, participants had to be aged between 18 and 80. The study, made in LimeSurvey [9], consisted of nine questions and lasted on average three minutes in total. Each participants received a compensation of 0.6€ after completing the study. For each question, a picture of NAO with eyebrows was presented and participant had to rate NAO emotion. NAO's eyes were turned off to avoid emotions bias, and angle of the eyebrow varied from -40° to 40° by a step of 10°. Each picture was presented one time in a randomized order. To evaluate the robot emotions we asked "What is the emotion expressed by the robot?" using a 7-point Likert scale: very angry (1) - angry (2) - a little bit angry (3) - neutral (4) - a little bit sad (5) - sad (6) - very sad (7).

Results and discussion

Using R [15], questionnaires results indicated a strong positive correlation between the eyebrow angle and the perceived emotion, $r(358)$ =0.88, p <.0001. An increase of the eyebrows angle was correlated by an increase in the Likert score for the expressed emotion. Moreover a one-sample t-test performed on the results obtained for $\alpha = 0°$ ($M = 3.92$, $SD = .57$) show they are not significatively different from the neutral value (4 on the Likert scale), $t(39) = $ -0.83, p =.41. As all other angles are significatively different from the neutral value (except $\alpha = -10°$ that is only marginally significant), we therefore assume that an angle of $\alpha = 0°$ can be considered as close to the neutral value. Table 1 summarize the statistical analysis and Likert results (emotion rating in function of eyebrow angle) are represented in Figure 6. These results confirm our hypothesis and show that we can assume a linear relation between the eyebrows angle and its affective interpretation. This approximation however, does not mean that the relation **is** linear, but only that a linear relation is a sufficient way to describe it. Indeed, an attentive eye might see a slight sigmoid shape in Figure 6. This would then suggest a visual hint of categorical perception, such as that a small angle variation would end up in large affective interpretation. Further research should be conducted to answer this question.
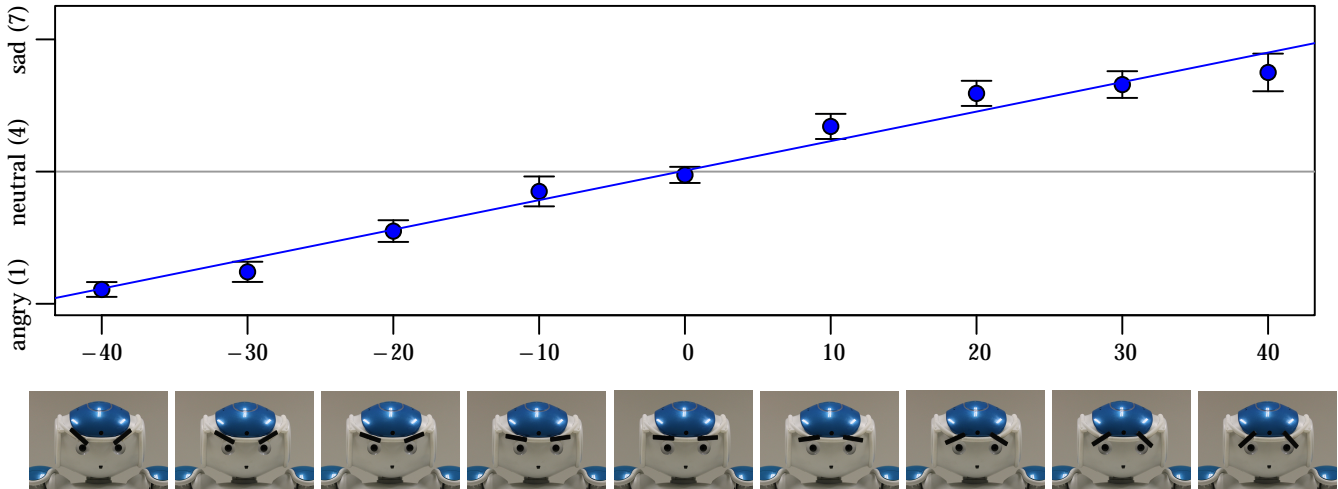
Fig. 6: Emotion expressed by NAO robot in function of the eyebrows' angle (in deg). Emotion was rated on a Likert scale from 1 (*anger*) to 7 (*sadness*). Dots represent the mean of the Likert scale value for each angle. Error bars indicate 95% confidence intervals. The blue line is the linear regression.

3.3 Experiment 3

This experiment focuses on the look of the eyebrows and raises two questions. As the eyebrows are intended for social interaction, we want to know which design is the most appreciated. However, we are also concerned by the prevaricate that could induce such design. Indeed, although we want to express emotions, we also want to be able to avoid expressing them at some moment: the neutral position. New shapes could however prevaricate the neutrality of the robot making it always look angry for example. We hypothesize that some eyebrow designs will be more appreciated than others. In particular we suppose that shape and size will have influences on the likeability of the robot. Additionally, shape and size of the eyebrows could also cancel robot's neutral emotion by inducing a bias toward anger or sadness.

Procedure

We recruited 40 participants (16 women and 24 men) with a mean age of 27.48 (*SD=8.14*) using Prolific Academic [1]. Participants had to be aged between 18 and 80 and could not have participated in a previous study on NAO eyebrows. The study, is similar to the previous experiment: it consisted of 6 questions composed

of two sub questions and lasted on average 4 minutes in total. Each participants received a compensation of 0.8€ after finishing the study. During the survey, 6 pictures were presented one by one in a randomized order. The pictures contained the face of NAO with different eyebrows designs, all in neutral position (see Figures 7a to 7f). In addition, NAO's eyes were turned off to avoid any bias that could be cause by the eyes color. For each pictures, participants had to rate the likeability of the eyebrows and the emotions of the robot trough two separated Likert scale. For the likeability rating [2], we asked: "Do you like the eyebrows?" on a 5-point Likert scale: not at all (1) - not really (2) - undecided (3) - somewhat (4) - very much (5). For the robot emotion, the question was similar to the question of experiment 2 (section 3.2).

Results and discussion

We first look at the factors (*shape* or *size*) that have influences on the eyebrow neutrality and likeability. Secondly we select the most appropriate eyebrows regarding the neutrality then the likeability.

A repeated-measures ANOVA with *shape* (bar, comma, circumplex) and *size* (small, big) as within-subjects fac-

Table 1: Results of experiment 2: one sample t-test with hypothesis: emotion score=4, for each angle.

angle [deg]	-40	-30	-20	-10	0	+10	+20	+30	+40
$t(39)$	-32.18	-20.10	-11.09	-2.68	-0.83	7.27	12.59	13.19	10.65
p value	<.0001	<.0001	<.0001	=.010	=.41	<.0001	<.0001	<.0001	<.0001

(a) small bar   (b) small comma   (c) small circumflex   (d) big bar   (e) big comma   (f) big circumflex
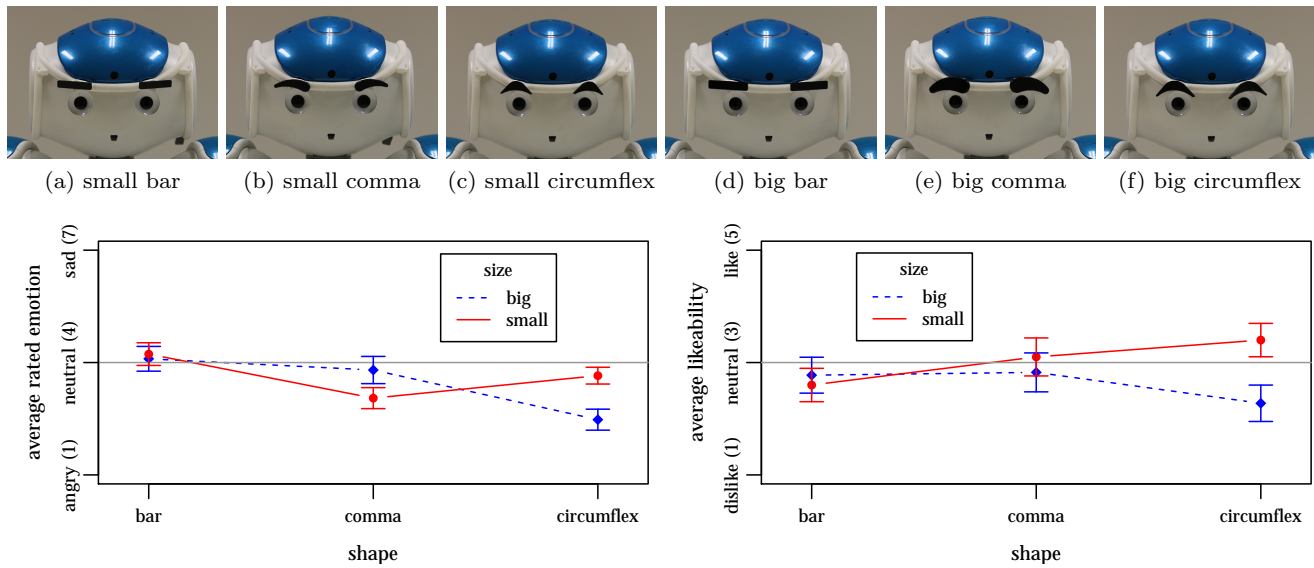
(g) Interaction plot of emotion rating for the 6 eyebrows designs. Error bars indicate 95% confidence intervals.

(h) Interaction plot of likeability rating for the 6 eyebrows designs. Error bars indicate 95% confidence intervals.

Fig. 7: Experiment 3 is performed on the eyebrows designs. A total of 6 solution are explored using 3 different shapes in two sizes: small (a-c) and big (d-e). Results are presented in Figures g and h.

tors was conducted. There was a significant main effect for *shape*, $F(2, 228) = 4.10$, $p = .017$. In general results where higher for the bar shapes (M=4.156, SD=0.96) than for the comma shape (M=3.42, SD=1.07) and the circumflex shape (M=3.06, SD=1.12). However there was no significant effect of *size*, $F(1, 228) = 1.44$, $p = .230$. In addition, there was a marginally significant interaction of *size* and *shape*, $F(2, 228) = 2.70$, $p = .069$. Similarly, a one-way within subjects ANOVA on likeability reported a significant effect of the *shape*, $F(2, 228) = 3.19$, $p = .042$. In general results seems to be higher for the comma shapes (M=2.96, SD=1.07) than for the circumflex shape (M=2.83, SD=0.98) and the bar shape (M=2.69, SD=0.98). There was no effect of the *size*, $F(1, 228) = 1.22$, $p = .270$, and a marginally significant interaction between the two factors, $F(2, 228) = 2.79$, $p = .063$.

Interaction plots are presented in Figure 7g for the rated emotion and in Figure 7h for the likeability. These results suggest that, when designing eyebrows, shape is an important concern. Surprisingly, and in opposition to our hypothesis, the size does not seem to have a direct influence. However size might have an interaction with the shape, suggesting that some shapes are better small, while other shapes are better big. Because pictures used in this experiment where focusing on the eyebrows, we believe that these results could be generalized to other social robots.

In the second part of these results, we select the most appropriate shape to be used as a reference in our next studies. We first select the shapes that are not biased in neutral position. One-sample t-tests performed on expressed emotion shows that three of the shapes are significantly different from the neutral value (4): shape *small comma*, $t(39) = -6.862$, $p < .0001$, shape *small circumflex*, $t(39) = -3.1631$, $p = .003$, and shape *big circumflex*, $t(39) = -11.00$, $p < .0001$. Theses shapes are therefore excluded. We then compare the likeability of the three remaining shapes (*small bar*, *big bar* and *big comma*). As t-test report no significant differences, $p > .1$, it suggest that theses three remaining shapes are equally adapted. As a personal choice, we propose to use the shape *big comma* as the design of reference. A possible limitation of this experiment comes from the definition of neutral position (horizontal). While the horizontal position is quite direct for bar and circumflex shapes, obtaining a horizontal position for the coma shape is not as straightforward, and could therefore conduct to different results.

3.4 Experiment 4

In this experiment, we want to show that it is possible for NAO to express emotion while performing other tasks. In consequence we hypothesize that participants watching the NAO robot performing neutral action such as pointing or waving will be able to decode the robot's

emotions through its eyebrows. In addition, we hypothesize that the type of action performed should not influence such emotion rating.

Procedure

For this study, 40 participants (19 women and 21 men) with a mean age of 28.7 (*SD=9.04*) were recruited using Prolific Academic [1]. To access the study, participants had to be aged between 18 and 80 years and could not have participated in a previous study on NAO eyebrows. The study was made using Limesurvey [9] and consisted of 6 questions presented in a randomized order. In total the survey lasted on average 3 minutes and its competition granted 0.6€ to each participants. For each question, participants had to watch a video of 7 seconds of the robot NAO performing an action, then rate the emotion expressed by the robot using the Likert scale presented in experiment 2 (section 3.2). In the videos, NAO performed either an *hello* or a *pointing* action. In the *hello* action, NAO waved its hand while saying "hello, my name is NAO". In the *pointing* action, NAO pointed on its right side with its arm and head, while saying "hey look!". For each action, the angle of the eyebrows where either corresponding to anger, neutral or sadness emotions. The eyes of the robot where colored in white in all conditions to avoid interference with the eyebrows.

Results and discussion

A one-way within subjects ANOVA was conducted on robot perceived emotion to compare the effect of the factor *action* and the factor *eyebrow angle*. There was
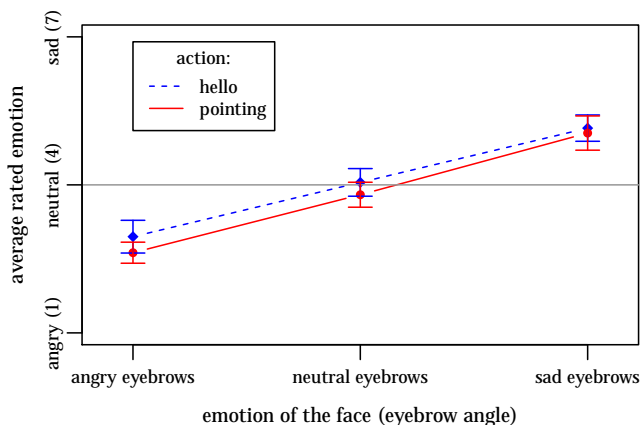

Fig. 8: Experiment 4: Interaction plot of the rated emotions for the videos. Error bars indicate 95% confidence intervals. We consider two factors: action performed and emotion of the face.

a significant main effect for *eyebrow angle*, $F(2, 228) = 27.62$, $p < .0001$. In general angry eyebrows ($M=2.78$, $SD=0.88$) were rated lower than neutral eyebrows ($M=3.92$, $SD=0.83$) and sad eyebrows ($M= 5.10, SD= 0.96$) on the 7-point Likert scale. These results are confirmed by one-sample t-tests showing that: 1) angry condition is significantly lower than neutral condition, $t(158) = -8.36$, $p <.0001$; 2) sad condition is significantly higher than neutral condition $t(158) = 8.23$, $p <.0001$. 3) Neutrality is conserved as the neutral condition is not significantly different from the neutral value (4 on the Likert scale), $t(79) = 0.8$, $p = .426$. In addition there was no significant effect of the *action*, $F(1, 228) = 0.530$, $p = .467$, and no interaction of the two factors, $F(2, 228) = 0.68$, $p = .507$. The interaction graph of the study is presented in Figure 8. This experiment supports our claim and confirms our hypothesis. Indeed, participants were able to recognize correctly the robot's emotion while it was performing other tasks.

4 Conclusion and future work

In this work we have proposed a unique and novel eyebrows device for the NAO robot as well as a platform independent method to map eyes colors and eyebrow orientation to emotions. The eyebrows device is easy to use and can be directly controlled through the Choreograph programming environment. In addition, four experiments explored different questions about the eyebrows. First, we showed the interest of the device as participants emotion recognition greatly increased with by its addition. Second, we confirmed the linear relation between angle and expressed emotion. Third, we investigated design criteria of eyebrows for NAO, and more generally for other robots. Fourth, we showed that with this device NAO robot was now able to express anger or sadness while performing other tasks.

In the future, we would like to see if it is possible to express other emotions with these eyebrows. Indeed, in this paper, the two eyebrows angles were always equal. But considering that left and right eyebrows can be controlled independently, we think that it could be possible to create other emotions, like disgust, where the face becomes not symmetric. Additionally, we would like to explore how small variation of the eyebrows around the neutral position could increase NAOs impression of aliveness.

References

1. (2015) Prolific academic. URL http://www.prolific.ac/

2. Bartneck C, Kulić D, Croft E, Zoghbi S (2009) Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. International journal of social robotics 1(1):71–81

3. Beck A, Cañamero L, Bard K, et al (2010) Towards an affect space for robots to display emotional body language. In: Ro-man, 2010 IEEE, pp 464–469

4. Cole J (1999) About face. MIT Press

5. Ekman P (1993) Facial expression and emotion. American psychologist 48(4):384

6. Ekman P (2007) Emotions revealed: Recognizing faces and feelings to improve communication and emotional life. Macmillan

7. Häring M, Bee N, André E (2011) Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots. In: RO-MAN, 2011 IEEE, pp 204–209

8. Johnson DO, Cuijpers RH, van der Pol D (2013) Imitating human emotions with artificial facial expressions. International Journal of Social Robotics 5(4):503–513

9. LimeSurvey Project Team / Carsten Schmitz (2015) LimeSurvey An Open Source survey tool. LimeSurvey Project, Hamburg, Germany, URL http://www.limesurvey.org

10. Mazzei D, Cominelli L, Lazzeri N, Zaraki A, De Rossi D (2014) I-clips brain, a hybrid cognitive system for social robots. In: Biomimetic and Biohybrid Systems, Springer, pp 213–224

11. Mehrabian A (1977) Nonverbal communication. Transaction Publishers

12. Van de Perre G, Van Damme M, Lefeber D, Vanderborght B (2015) Development of a generic method to generate upper-body emotional expressions for different social robots. Advanced Robotics 1(ahead-of-print):1–13

13. Pioggia G, Ahluwalia A, Carpi F, Marchetti A, Ferro M, Rocchia W, De Rossi D (2004) Face: Facial automaton for conveying emotions. Applied Bionics and Biomechanics 1(2):91–100

14. Posner J, Russell JA, Peterson BS (2005) The circumplex model of affect an integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and psychopathology 17

15. R Core Team (2015) R A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org

16. Russell JA (1978) Evidence of convergent validity on the dimensions of affect. Journal of personality and social psychology 36(10):1152

17. Saldien J, Goris K, Vanderborght B, Vanderfaeillie J, Lefeber D (2010) Expressing emotions with the social robot probo. International Journal of Social Robotics 2(4):377–389

18. Saldien J, Vanderborght B, Goris K, Van Damme M, Lefeber D (2014) A motion system for social and animated robots. International Journal of Advanced Robotic Systems 11

19. Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer Science & Business Media

Albert De Beir received in 2014 his Master degree in electromechanical engineering from the Vrije Universiteit Brussel as well as his Bachelor degree in Psychology from the Université Libre de Bruxelles. He is now pursuing a PhD at the Robotics and Multibody Mechanics Research Group. His research interest focus on the development of hardware and artificial intelligence for social robots.

Hoang-Long Cao received his B.Eng. in Mechatronics from Can Tho University in 2010, and M.Sc. in Electromechanical engineering from the Vrije Universiteit Brussel and Université Libre de Bruxelles in 2013. From September 2013, he has been working as a PhD researcher at the Multibody Mechanics and Robotics research group (VUB). His research interest is social robotics. Currently he is involved in DREAM project for the development of robot-enhanced therapy for children with autism spectrum disorders.

Pablo Gomez Esteban received, in March 2014 by the Rey Juan Carlos University of Madrid, his PhD for his thesis entitled Cooperation and Competition in Emotional Robot Societies with highest distinction. His research was addressed towards decision making processes within competitive and cooperative environments; the impact that affective factors have in such processes; and designing and implementing those models to be used in non-expensive platforms (like the AiSoy1 robot) within domestic environments. His research interests also lie on cognitive architectures for social robots and HRI.

Greet Van de Perre studied Mechanical Engineering at the Vrije Universiteit Brussel. After graduating in 2011, she started her PhD at the Robotics and Multibody Mechanics Research Group, funded by an FWO fellowship. Her research focuses on the development of a generic gesture system for social robots and the design of an actuated upper-torso for the social robot Probo.

Dirk Lefeber received a degree in Civil Engineering in 1979 and a PhD in Applied Sciences in 1986 from the Vrije Universiteit Brussel. Currently, he is full professor, former head of the department of Mechanical Engineering and head of the Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, which he founded in 1990. The research interests of the group are new actuators with adaptable compliance, dynamically balanced robots, robot assistants, rehabilitation robotics and multibody dynamics, all in the context of physical and cognitive human robot interaction with emphasis on improving the quality of life of human.

Bram Vanderborght focused his PhD research on the use of adaptable compliance of pneumatic artificial muscles in the dynamically balanced biped Lucy. In May-June 2006, he performed research on the humanoids robot HRP-2 at the Joint Japanese/French Robotics Laboratory (JRL) in AIST, Tsukuba (Japan) in the research 'Dynamically stepping over large obstacles by the humanoid robot HRP-2'. From October 2007-April 2010 he worked as post-doc researcher at the Italian Institute of Technology in Genova (Italy) on the humanoid robot iCub and compliant actuation. Since October 2009, he is appointed as professor at the Vrije Universiteit Brussel. He has an ERR starting grant on SPEA actuation concept. He is member of the Young Academy of the Royal Flemish Academy of Belgium for Science and the Arts. His research interests include cognitive and physical human robot interaction, social robots, humanoids, robot assisted therapy and rehabilitation/assistive robotics.

Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

TECHNICAL REPORT
Robot Low-Level Motor Control

Date: **20/03/2015**

Technical report lead partner: **Plymouth University**

Primary Author: **P. Baxter**

Revision: **2.0**

Contents

Summary

This technical report describes the first version of the low-level robot control system using YARP as the communications infrastructure. This system is designed to be extensible, and flexible to the requirements of the 'higher level' robot behavioural components. A demonstrator system has been constructed for the Nao, but the structure is intended to be applicable to other robot embodiments (i.e. specifically the Probo, assuming a similar level of partially abstracted control is possible).

Principal Contributors

The main authors of this document are as follows (in alphabetical order).

Paul Baxter, Plymouth University
Tony Belpaeme, Plymouth University
Emmanuel Senft, Plymouth University

Revision History

Version 1.0 (P.B. 01-12-2014)
Initial outline of control structures.

Version 2.0 (P.B. 20-03-2015)
Updated control structures after implementation and test application to Sandtray WoZ system.

1 Overview

This technical report describes the first version of the low-level robot control system using YARP as the communications infrastructure. This system is designed to be extensible, and flexible to the requirements of the 'higher level' robot behavioural components. A demonstrator system has been constructed for the Nao, but the structure is intended to be applicable to other robot embodiments (i.e. specifically the Probo, assuming a similar level of partially abstracted control is possible).

2 High Level Robot Control

The general high-level behaviour of the robot is defined by the intervention scripts (D1.1), and instantiated in a 'script manager' that handles the flow of phases within an intervention and exposes the current intervention state (cf *getInterventionState()*). This script will be interpreted through the autonomous controller (figure 1). This will facilitate the adaptation of the robot behaviour in two respects (outlined in D3.1): (i) adaptation of the robot behaviour to that of the child (if desired/appropriate: alignment/entrainment); (ii) the appropriate handling of child/environment states/behaviours that fall out of the script (child distraction, deviation from script, etc).

The autonomous controller itself is composed of a number of sub-systems, as defined in the DoW: reactive, attention, deliberative and self-monitoring (all within the context of expression and actuation). The definition, implementation and interaction of these sub-systems is the subject of WP6[1]. This technical report focuses rather on how to execute the desired action once it has been chosen.
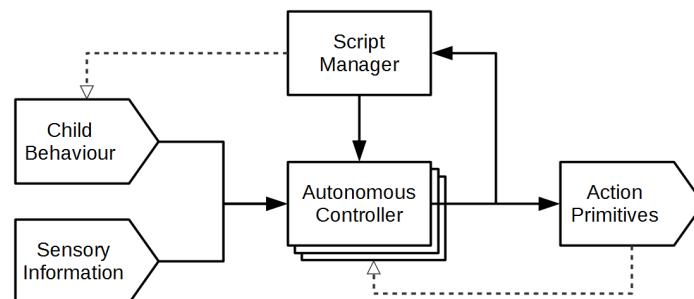

Figure 1: High level description of the robot control system; arrows denote information flow on YARP ports. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied. The intervention script provides context for child behaviour interpretation (wizard GUI interfaces not shown for clarity).

Once an action has been chosen, execution is required. We choose to handle this with a separate component, the Robot Interface, to provide a distinction between the robot neutral action primitives, and the robot specific commands necessary for actuation. This component also provides feedback functionality for the therapist interface (Wizard GUI, figure 2) and to the other autonomous cognitive subsystems (specifically the Deliberative and Expression and Actuation subsystems) for planning/learning purposes.

---

[1]This discussion begun at the DREAM M6 meeting (Amsterdam), refer to initial outline architecture plans from Pablo Gomez (VUB): preliminary deliverables D6.3 and D6.4 cover this.
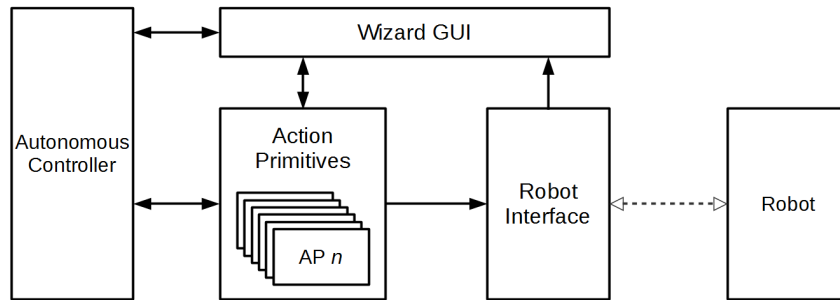
Figure 2: The Robot Interface coordinates motor control for the robot, and provides information to the Wizard GUI (and autonomous controller - not shown) regarding robot/behaviour state. Solid arrows denote information flow on YARP ports, dashed arrow shows communication using robot API (e.g. naoqi API for Nao robot). See figure 3 for processing steps within Robot Interface.

3   Low Level Robot Control

The low level motor control structure is shown in figure 3), which describes the execution of an action primitive once it has been selected (either manually or automatically) - i.e. the processing within the Robot Interface component (see figure 2). At each stage of processing a message is emitted on a YARP port, which can be picked up by the autonomous action selection mechanism (e.g. feedback to improve selection) and/or displayed on the wizard interface to provide both confirmation of action(s) performed and information should anything go wrong. The messages are described below (table 1). The YARP port defined for these messages (from the robotInterface component) is:

/cognitiveControl/robotInterface/motorFeedback:o

The strings are sent in a bottle from this port. In the demonstrator, the GUI is set to receive this information to display, however, the same mechanism has been used to provide information to the WoZ controller, and is logged to a log file in the sandtray WOZ system.

Table 1: Motor control execution states sent on YARP port, providing information on failures and success of requested action primitives. See figure 3 for control flow.

| State	Parameter	Description
CONNECTIONFAIL	-	Connection to the robot has failed
ISIMPOSSIBLE	-	This command is not possible with this robot
ISCANCELLED	-	This command is not allowed at the moment
ISWAITING	-	Now waiting to execute the requested command
ISINPROGRESS	-	Command is currently being excuted
ISFAILED	current state information	Command has failed to be executed
ISCOMPLETE	-	The robot has completed the requested action
CONNECTIONSUCCESS	-	(debug) connection to robot is operational
ISPOSSIBLE	-	(debug) the requested command is possible with this robot
NOTWAITING	-	(debug) waiting is not required to execute this action
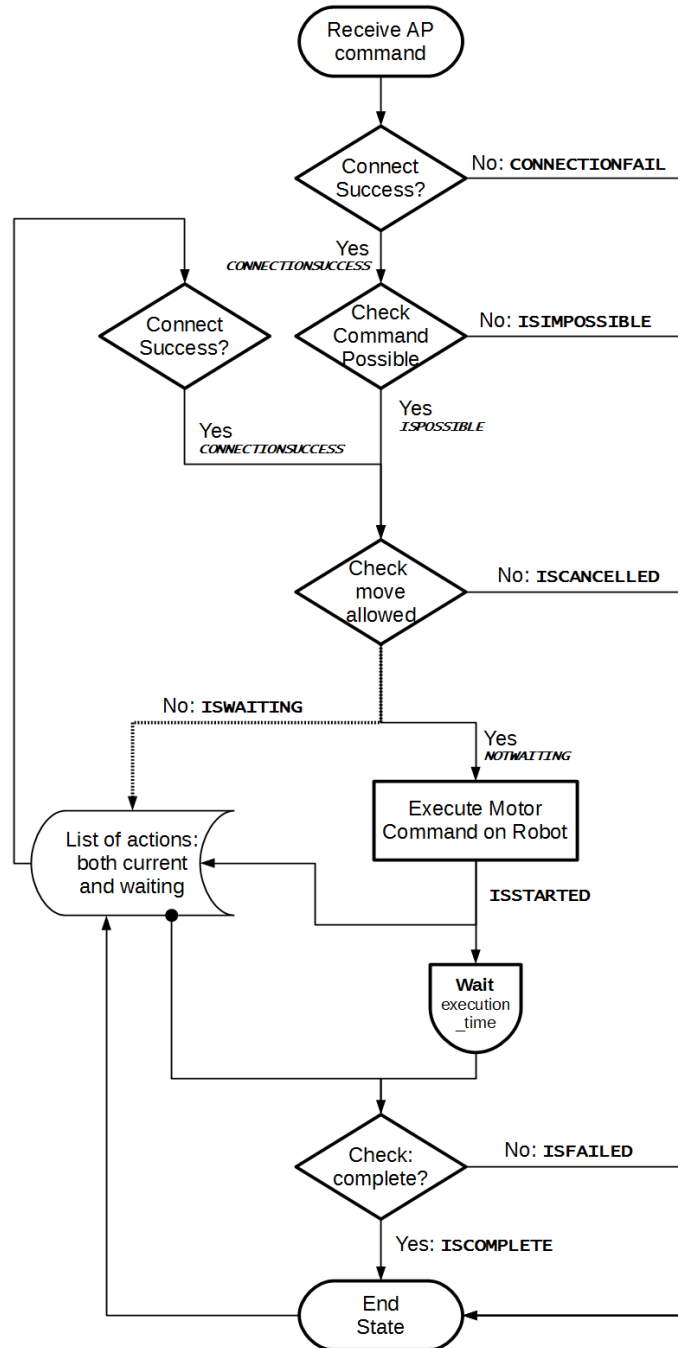
Figure 3: Low level motor control, showing feedback states (in bold, italicised shows additional debug feedback) from various stages emitted on a YARP port. Input is the request (from autonomous action selection or wizard interface) for execution of a single Action Primitive (AP). Dotted arrow denotes path if bool Queue is true, and on the first iteration only. The 'Check Command Possible' and 'Execute' operations are embodiment specific, with the remaining robot-platform specific (i.e. an API capable of supporting such functionality is assumed).

4 Demonstration System

A simple demonstrator system has been constructed to illustrate the low-level motor control of the robot (figure 4). In this system the "Robot Interface" is a YARP component that accesses functions from the naoqi API (i.e. YARP does not run directly on the robot). While the first demonstrator implementation was created with naoqi API version 1.22.1.46, the stable version to be used in the project now requires naoqi API version 2.1.2.17. The demonstrator simply shows the control for opening and closing the hand of the Nao, as controlled by two buttons in a GUI. Also shown in the GUI are the messages returned that could be used to inform autonomous control (see figure 3 for an overview).

The naoInterface (the nao-specific version of robotInterface) subsystem requires the use of naoqi API version 2.1.2.17, with the same version installed on the robot itself. Aldebaran recommend that the version of Choregraphe used matches the version used on the robot (compatibility between different versions of naoqi is not always straight forward, and not always fully documented, even between what seem to be minor sub-versions).
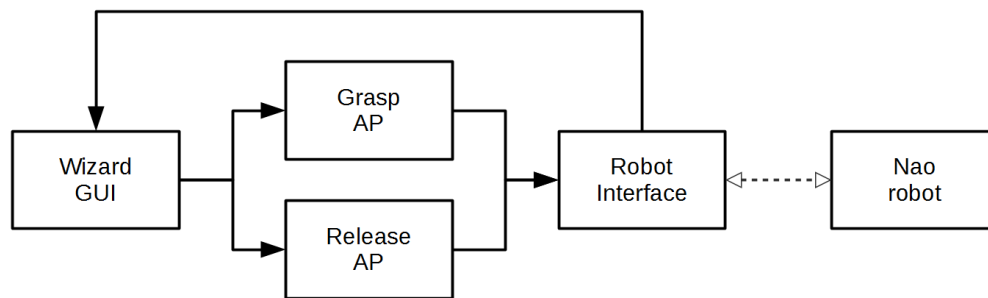

Figure 4: Basic system demonstration of low-level control described in section 3. The Wizard GUI has controls and displays feedback from the Robot Interface (see table 1).

Being based on naoqi, this example is specific to the Nao. Use with other robots will entail the use of the appropriate robot control API: a number of basic functionalities are assumed from these alternatives (as described above). However, the motor control procedures described above are not restricted to only one specific robot embodiment, and will be applicable to the Probo as well.

Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

TECHNICAL REPORT
Organisation of Cognitive Control and Robot Behaviour

Date: **02/02/2015**

Technical report lead partner: **Plymouth University**

Primary Author: **P. Baxter**

Revision: **2.2**

Contents

Summary

The purpose of this technical report is to summarise the motivations and constraints underlying the cognitive control structures, and to outline an organisation of these sub-systems. This is a proposal only; this document is intended to be a working one, to be updated as required during development. This version of the report is based primarily on the discussions that took place in Brussels (23/01/15).

Principal Contributors

The main authors of this document are as follows (in alphabetical order).

Paul Baxter, Plymouth University
Tony Belpaeme, Plymouth University
Hoang-Long Cao, VUB
Albert De Beir, VUB
Pablo Gomez, VUB
Emmanuel Senft, Plymouth University
Greet Van de Perre, VUB
Bram Vanderborght, VUB

Revision History

Version 1.0 (P.B. 21-01-2015)
Initial outline of ideas for the DREAM supervised autonomous robot control.

Version 2.0 (P.B. 26-01-2015)
Updated after discussions during Brussels meeting 23rd Jan 2015.

Version 2.1 (P.B. 02-02-2015)
Clarifications and updates following further discussion.

Version 2.2 (P.G. 27-02-2015)
Some modifications regarding priority system and Expression and Actuation subsystem functionality.

1   Aims and Constraints

An attempt is made in this section to formulate what the ideal (semi) autonomous system should conform to in terms of both clinical outcomes (i.e. the requests from the psychologists to improve the outcomes of individual children through robot-assisted therapy) and potential research (where this does not conflict with the clinical objectives).

The primary goal of the work in WP6 is to provide robot behaviour to facilitate the Robot-Assisted Therapy, see [1]. The main visible outcome of this should be the ability of the robot to execute the evaluation and therapeutic scripts as defined by the therapists. Whilst this must be achieved to fulfil the aims of the project, there are a number of areas in which there would be a role for behavioural adaptation, learning, and autonomous decision making. These should not however conflict in any way with the therapeutic goals for any given interaction session - indeed, it is necessary to vary the degree of shared control between the autonomous behaviour and the wizard supervisory control if this is more appropriate for a given child and/or circumstance.

Primary among these is the high probability that the interaction (due to the behaviour of the child for example) will deviate from the script. This must be handled in a manner consistent with the therapy, to not upset the child, and possibly (depending on the context) trying to re-engage the child with the script. A range of strategies will be required to deal with these situations, depending on the individual child (his/her characteristics) and the actual context for the departure from the script. This behaviour is likely to require flexible action selection, and will therefore require substantial research effort.

A second reason is that the robot is to demonstrate social behaviour in a supervised autonomous manner (with the requirement that the supervisor may over-rule this autonomous social behaviour if required). Social behaviour requires behaviour that is adaptive to the interaction partner in a range of interaction modalities (e.g. movement and speech). The autonomous behaviour of the robot must therefore be responsive to this, in a manner that is not, and indeed can not, be predetermined in the script.

Thirdly, given the range of intervention scripts that have been defined, there is also a possible need to modify the relative difficulty of the task (and/or interaction) given the specific characteristics and performance of the interacting child. This would, for example, involve varying the number and type of social behavioural cues used, the complexity of the required motor behaviours to complete the task, and/or the number of steps in the task.

The interfaces of the cognitive controller (WP6) with the rest of the DREAM integrated system (WP's 4 and 5) have already been defined. The intention in providing this overview document is to show how the subsystems of WP6 fit together to determine the behaviour of the robot in therapy interactions: the context in which each subsystem must operate is thereby defined. Initially, the skeleton of this system will be implemented in the most straightforward manner possible (with simplified code implementations of full component functionality for example) to check that the system fulfils all the requirements. This skeleton can then be filled in with more appropriate functionality over the course of the project.

2   Overall Organisation

A general high level description of the robot control system is shown is figure 1. This basically describes how the autonomous controller is informed by three external sources: the child behaviour description, sensory information, current intervention script state, and input from a therapist (e.g.

emergency stop, not shown in diagram). Combining these sources, the autonomous controller should trigger as an output the appropriate sequence of action primitives to be performed (as well as some feedback via the WoZ GUI), which then gets executed on the robot.
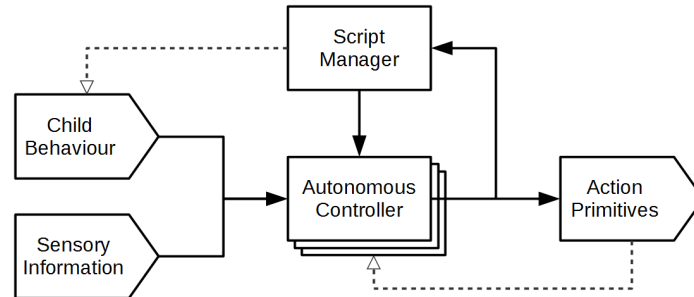

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied. The intervention script provides context for child behaviour interpretation.

The autonomous controller is composed of a number of sub-systems, as described in the DoW: Reactive, Attention, Deliberative, Self-Monitor and Expression and Actuation. These sub-systems interact, and must combine their suggested courses of actions to produce a coherent robot behaviour, in the context of constraints laid down by the therapist (for example, the script to be followed, types of behaviour not permissible for this particular child because of individual sensitivities, etc). An additional challenge is to ensure that the resulting system is independent of specific robot platform. As a result, we have formulated the following architecture describing how cognitive control informed by the therapy scripts is to be achieved (figure 2), an outcome of the WP6 meeting in Brussels (23/01/14). The following sections provide some further outline details of the main subsystems.
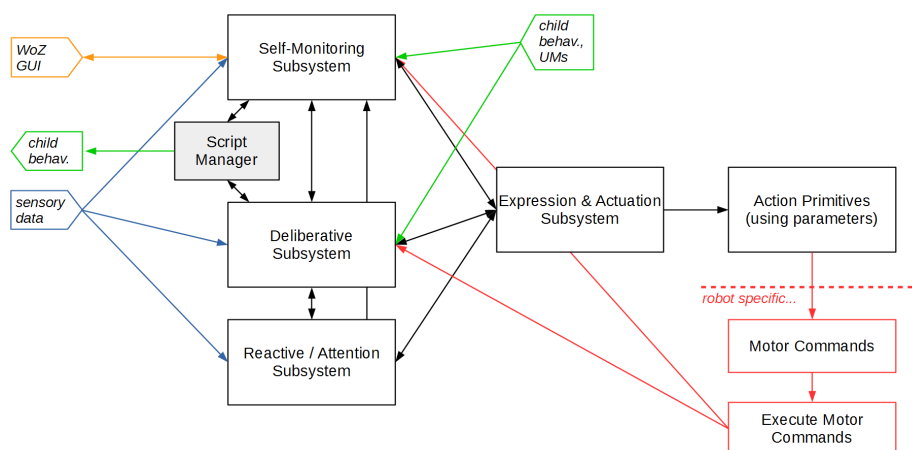

Figure 2: Description of the cognitive controller subsystems. The script manager is separate from, but tightly interacts with, the deliberative subsystem to enable the robot control system to generate appropriate social/interaction behaviour even in the absence of an explicit interaction script. UMs: User Models.

3   Reactive/Attention Subsystem

In the DoW, these are separated into two distinct subsystems. The reactive subsystem provides the general life-like behaviour of the robot (small motions, eye blinking, balancing, recovering from falls, 'pain' reactions, etc) in an as appropriate manner as possible (possibly requiring pilot studies to verify this). However, it should be possible to turn off these behaviours should the therapist deem it necessary for a particular child. This functionality is not envisaged to involve learning or adaptation. The attention subsystem is a "*combination of perceptual attention ... and attention emulation*". Making eventual use of saliency maps and habituation filters, this functionality will be guided by the deliberative subsystem.

We instead propose that these two subsystems be combined into a single component, due to the significantly overlapping technical systems required to fulfil the functions required. Both subsystems require access to features of the environment and interacting person(s) to respond appropriately (e.g. looking at a face or diverting attention to a loud noise somewhere in the environment). Managing this in a single component therefore seems a sensible choice so that functionality is not replicated. As planned in the DoW, it will be possible for the supervising therapist to switch off these functionalities if required for interaction with a particular child.

4   Deliberative Subsystem

A central aspect of the cognitive controller is the ability to follow intervention scripts as defined by the clinicians for both diagnosis and therapy. These scripts describe the high-level desired behaviour of the robot[1], and the expected reactions and behaviours of the child, in a defined order.

The decision was made to separate the script manager from the deliberative subsystem itself (fig 3). This decision was taken for a number of reasons. Firstly, it enables the cognitive control of the robot to be independent of the precise application domain - with the intention that the developments made would be more generally applicable within the field of social robotics, although the script-based behaviours remain a central part of the behaviour generation of the system. Secondly, it ensures that it would be possible to change the scripts in the future to alter their relative difficulty, by for example including further steps in the intervention, changing the type of intervention, or creating different activities, due to a modular design[2]. As a consequence of this, the deliberative subsystem is now primarily focussed on action selection considerations, making use of a range of algorithms and methodologies as will be explored in the coming years. Thirdly, this division of the script manager from the deliberative subsystem enables the system to generate coherent behaviour even if there is not a script active at a given moment. This could be useful for periods between the explicit intervention sessions for example, where the robot would then still be able to respond appropriately to environmental stimuli, if so desired by the therapists. These are consistent with the aims expressed within the WP6 DoW.

The script manager itself separates the logic necessary to manage progression through the script (by taking into account the available sensory feedback after actions for example) from the script itself. This makes it straightforward to add new scripts or modify existing scripts as required. This logic management could in the first instance be achieved using a Finite State Machine (FSM).

---

[1]These predefined robot behaviours differ from the the low-level motor control of the robot, as these may be mixed with other aspects of behaviour not specified explicitly in the high-level intervention script; e.g. the addition of attention to unexpected events in the environment.

[2]As noted above, these high-level scripts do not necessarily completely define the behaviour of the robot, and are distinct from any predefined robot motor control sequences that may be used, such as waving or nodding.
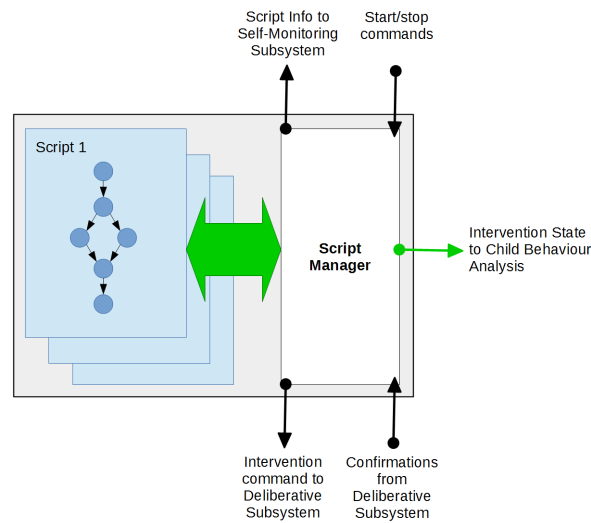
Figure 3: Overview of the script manager subsystem. The scripts are defined independently of the script manager, which is responsible for stepping through the script as appropriate and communicating with the other subsystems as required.

One possibility for the scripts is that each step in the script be defined as a 3-tuple of the form: *[existing_state, proposed_action, consequent_state]*. In this context, *existing_state* could be defined by default to be the *consequent_state* of the previous step. The *proposed_action* defines what action should be taken by the robot, and be one of the actions (or unique identifier thereof) defined in D1.2. The *consequent_state* defines what robot state should be expected (in terms of sensed state) if the *proposed_action* were successfully completed. This may be used by the script manager to determine if and when it is appropriate to move onto the next script step. These 3-tuples may initially be held in a plain text file to facilitate examination and modification by the clinical staff as required. This can be changed later to ease the process (for example by providing a drag-and-drop script construction GUI).

The deliberative subsystem is the primary locus of autonomous action selection in the cognitive controller (fig 2). This subsystem takes as input sensory data, child behaviour information, information on what step should be next executed from the therapy script, and higher-level direction from the wizard/self-monitoring subsystem. It then proposes what action should be taken next by the robot (this proposal is sent to the expression and actuation subsystem). In a normal script execution context, the deliberative subsystem is the primary driver of behaviour, which would typically propose the next script step.

There are however a number of circumstances in which this is not the most appropriate action to perform. For example, if the child is detected to have very low engagement with the task (as determined from the WP5 component/s, and/or information from WP4 sensory system saying the child is looking away for example), then it would be appropriate to attempt to re-engage the child with the robot/task prior to executing the next stage in the therapy script. In this case, the deliberative subsystem can choose to depart from the behaviour defined in the script, and instead propose a different behaviour.

5   Expression and Actuation Subsystem

The main functionality of this subsystem is to determine which combination of low-level actions the robot should execute next, and how these actions are to be performed. Suggestions for actions to take will come from three other subsystems: deliberative, reactive/attention, and self-monitoring, and the affective state generated by the deliberative subsystem, see left side of figure 4. Along with this, it is assumed that the supervising therapist, through the GUI, will determine (either beforehand or in real time) the aspects of robot behaviour that should be executed, from which relative priorities will be determined for the three subsystems. This covers for example whether external disturbances (a loud noise in the background, or the appearance of a new face) should be reacted to by the robot (by leaving the script for a while for example), or ignored (with the script rigidly adhered to). The Expression and Actuation subsystem will combine these sources of information in an appropriate manner, see Motion Mixer in figure 4, ensuring that the stability of the robot is maintained. For example, if a greeting wave is requested by the deliberative subsystem, and the reactive/attention subsystem wants to look at a face that has been detected, then the expression and actuation subsystem can combine the two by executing both (if the robot can remain stable by doing so). For a basic first step switches based on priority level could be used: i.e. if the script requests an action, execute it (and only it), but if there is no script action requested, then do what the reactive/attention subsystem proposes. However, the intention is to provide full behaviour mixing capabilities based on derived priorities from the therapists.

All this should be complemented by affective information, if this is available and appropriate to use. For example, the speed of motor execution could be related to arousal levels, or the choice of action sequence could be based on valence levels (if appropriate alternative sequences exist). This functionality will need to be switched on or off as required by the therapist based on child-specific considerations, and the relation to the therapy script (it may not appropriate to add emotional colouring to actions during the diagnosis procedure for example).

To approach such challenges, the first task should be to design a platform-independent representation of expressions. Different robots use the Facial Action Coding System (FACS) by Ekman and Friesen [2] to abstract away from the physical implementation of the robot face. FACS decomposes different human facial expressions in the activation of a series of Action Units (UA), which are the contraction or relaxation of one or more muscles. In a similar way, Body Action Units (BAU) will be defined together with a Body Action Coding System, where the different gestures are decomposed in the activation of BAUs. The BACS will point out the Action Units that need to be actuated for the generation of a desired gesture or body pose. This system avoids pre-programming of robot-dependent body poses and actions, which is relevant since humans are able to recognize actions and emotions from point light displays (so without body shape) [3].

The physical actuation of Action Units will depend on the morphology of the robot: a mapping will be needed between Action Units and physical actuators, this mapping will be specific to a robot platform and we will explore the possibility of learning this mapping. To translate this to the morphology of the robot, the Action Units need to be mapped to the degrees of freedom, and thus to the joints of the robot, see right side of figure 4.

A second task will be the categorisation of actions, comprised of temporal series of FACS and BACS, and the organisation in libraries that are accessible from the behaviour subsystems (Reactive, Attention and Deliberative). All actions for the different behaviours should be stored and expanded upon without the need to reprogram other subsystems.
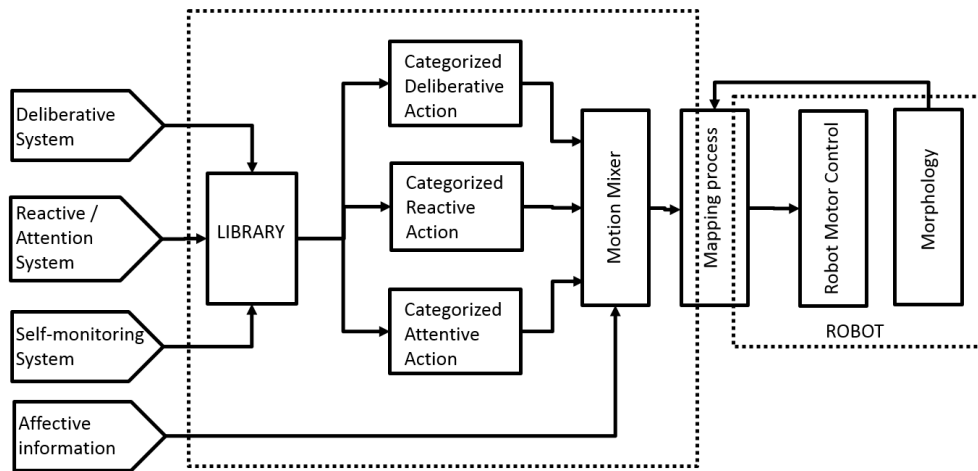
Figure 4: Overview of the Expression and Actuation subsystem. This subsystem receives inputs from several sources, categorizes them using the Library module and mixes them up to create a unique behavior. Such behavior is mapped into the joint configuration of the corresponding robot. This last process is done collaboratively between the subsystem and the robot.

6 Self-Monitoring Subsystem

The self-monitoring subsystem provides an oversight mechanism (or set of mechanisms) of the robot behaviour. It is intended to provide a check to prevent technical limits being exceeded (of the robot[3]), and to prevent any ethical boundaries being crossed. This subsystem should have some degree of autonomous behaviour, with the intention being that these checks be implemented in a set of predefined rules, with no role for learning within this subsystem.

During the discussions, it was proposed that the self-monitoring subsystem should also be integrated explicitly with the therapist GUI. In line with the principle of supervised autonomy established in the project, the therapist ("wizard") should be able to monitor the behaviour of the robot, and be able to intervene if necessary, either stopping the behaviour, modifying a behaviour, or setting an alternative behaviour. Having this oversight function go through the self-monitoring subsystem seems to be a reasonable solution. By specifying the required priorities for each subsystem depending on the needs of the therapy, and using the "alarm signals", the supervising therapist can stop the robot or modify its behaviour as desired.

Regarding both the autonomous oversight functions and the supervised actions, there are a number of issues that require exploration and further definition over the course of the project. One thing is how the robot should behave, and what feedback it should give to the child, should something go wrong. Possible alternatives are described in the DoW.

7 Action Primitives and Motor Execution

The behavioural functions of the action primitives required for completion of the therapy scripts have been defined. The execution of these is handled in a number of steps, as outlined in the "Robot

---

[3]This is mentioned here as it is listed in the DoW as a competence of the self-monitoring subsystem, however, this functionality is at least partially implemented in the low-level motor control system of the robot: see section 7.

Low-Level Motor Control" technical report. This provides an interface between the control system (handled in a Yarp-based system) and the API of the robot hardware (Naoqi in the case of the Nao). The purpose is both to provide a bridge between the two systems, and to provide information to behaviour planning and supervisory oversight regarding the progress of motor command execution, including why a fail occurs if it does. This can be used to inform future action selection for example (by providing feedback for learning).

In addition to this low-level control system, there is the possibility that hardware abstraction can be handled automatically: i.e. that motor commands at the joint level can be determined automatically for different robot embodiments, without having to manually encode each specific action.

8 Other aspects of the Cognitive Control System

8.1 User Models

One functionality that was not explicitly defined in the proposed architecture, WP6, or indeed elsewhere in the project, is some source of information on the child. This information could encompass personal identification and preference information that could be used in conversations (e.g. name, age, favourite colour, etc), and possibly also ASD diagnosis information (perhaps as emerging from the diagnosis interaction scripts).

These user models would enable, for example, inform learning mechanisms (within the deliberative subsystem for example) to link behaviours and outcomes with specific characteristics of individuals (indicated in figure 2). This information need only be uniquely identifiable rather than linked to a specific child - although the extent to which this can be done needs to be assessed in light of ethics considerations (cf. WP7 ethics manual draft, December 2014). Technically, in the first instance, a unique impersonal identifier may be used to represent an individual child. Where this information should reside, how it should be stored, etc, has not been decided. It would probably be useful however to coordinate this system with WP5, as the child behaviour interpretation methods may find such information useful too to be able to provide more personalised characterisations of engagement and performance for example.

References

[1] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, and B.ram Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, 2012.

[2] Ekman P and Friesen W. *Facial Action Coding System*. Consulting Psychologists Press, 1978.

[3] A. P. Atkinson, W. H. Dittrich, A. J. Gemmell, A. W. Young, et al. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception-London*, 33(6):717–746, 2004.

Period 2

Can Less be More? The Impact of Robot Social Behaviour on Human Learning

James Kennedy[1]   and   **Paul Baxter**[1]   and   **Tony Belpaeme**[1]

Abstract.   In a large number of human-robot interaction (HRI) studies, the aim is often to improve the social behaviour of a robot in order to provide a better interaction experience. Increasingly, companion robots are not being used merely as interaction partners, but to also help achieve a goal. One such goal is education, which encompasses many other factors such as behaviour change and motivation. In this paper we question whether robot social behaviour helps or hinders in this context, and challenge an often underlying assumption that robot social behaviour and task outcomes are only positively related. Drawing on both human-human interaction and human-robot interaction studies we hypothesise a curvilinear relationship between social robot behaviour and human task performance in the short-term, highlighting a possible trade-off between social cues and learning. However, we posit that this relationship is likely to change over time, with longer interaction periods favouring more social robots.

1   INTRODUCTION

Social human-robot interaction (HRI) commonly focuses on the experience and perception of human users when interacting with robots, for example [2]. The aim is often to improve the quality of the social interaction which takes place between humans and robots. Companion robots increasingly aim not just to merely interact with humans, but to also achieve some goal. These goals can include, for example, imparting knowledge [11], eliciting behaviour change [17] or collaborating on a task [3, 13]. Studies with these goal-oriented aims often still apply the same principles for social behaviour as those without goals - that of maximising human interaction and positive perception towards the robot. The implicit assumption is often that if the interaction is improved, or the human perception of the robot is improved, then the chance of goal attainment will be increased as well.

In this paper, we focus on learning. In this context, we take learning to be the acquisition and retention of novel information, and its reuse in a new situation. This definition covers 3 areas from each of the 'Cognitive Process' (remember, understand, apply) and 'Knowledge' (factual, conceptual, procedural) dimensions of learning according to the revised version of Bloom's taxonomy [14]. Learning outcomes can depend on many different elements of behaviour, such as motivation [20] and engagement [4], which will also be considered here.

The remainder of this paper is structured as follows. First, studies in which social robots assist humans in learning will be reviewed, with the intention of showing the complex variety of results obtained when relating learning to the social behaviour of the robot (Section 2). Human-human interactions are then considered and are used as a basis to create a hypothesis about the relationship of robot social behaviour and human performance in tasks over both the long and short-term (Section 3). This leads to a discussion of the implications for HRI design in such contexts (Section 4).

2   MIXED LEARNING RESULTS IN HRI

One area of great potential in HRI is in using robots for education. However, mixed results are often found when using social robots to teach or tutor humans. Despite regular reports of liking robots more than virtual avatars, or preferring more socially contingent robots over those with less social capability, the human performance in learning tasks doesn't always reflect these positive perceptions [11, 12, 17, 22]. Conversely, significant cognitive gains have been found when comparing robots to virtual avatars, with varied amounts of contingent behaviour [15, 16]. Similar effects have been seen in compliance when comparing agents of differing embodiments [1]. Whilst the varied context and content to be learned between these studies could account for many of the differences in results, we suggest that the relationship between social behaviour and learning performance may be more complex than typically assumed.

Commonly, when behavioural manipulations are carried out on one or two cues, such as in a study by Szafir *et al.* varying the gestures and vocal volume that a robot uses, there are clear benefits to the human in terms of performance in learning tasks [26]. However, these positive benefits may be lost, or even reversed when larger manipulations to the social behaviour of the robot are applied, as in [12]. While it may be reasonably assumed that the effect of multiple individual cues is additive, this does not seem to be in accordance with the empirical evidence. Indeed, the proposition that social cues are perceived by humans as a single percept [29] considers individual social cues as providing the context for the interpretation of other social cues (recursively), leading to non-trivial interactions and consequences when multiple social cues are applied. There is thus the possibility that making large manipulations in social behaviour by varying multiple social cues simultaneously does not elicit the benefits that varying each of these cues individually would, as suggested by the data.

Human expectations of sociality will play a large role in an interaction with a robot. It has been suggested that a discrepancy between categorical expectations and perceptual stimuli could account for negative cognitive reactions [19]. We posit that humans don't necessarily expect to interact with a robot exhibiting social behaviours and that the discrepancy between their expectation and the reality of the interaction could create a cognitive reaction which impedes learning. This might explain some results showing a lack of improvement when social presence of an agent is increased (such as when going from a virtual avatar to a robot, as in [10, 17]), or when social behaviour

---

[1] Centre for Robotics and Neural Systems, Cognition Institute, Plymouth University, United Kingdom
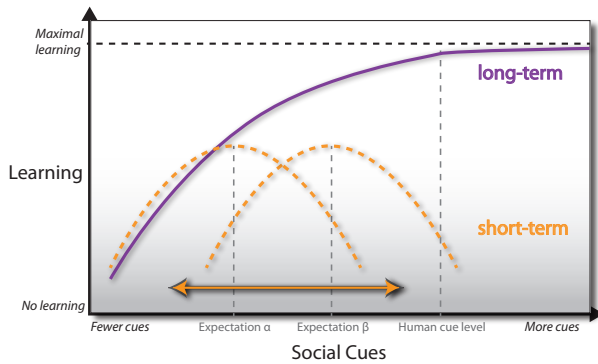 email: {james.kennedy, paul.baxter, tony.belpaeme}@plymouth.ac.uk

Figure 1: Hypothesised relationship between social behaviour (characterised by *immediacy* for example) as exhibited by a robot and its impact on the learning of a human in both the short and long-term. The position of the short-term curve is dependent on the humans' prior expectations of social behaviour (e.g. α is the expectation of fewer social cues from the robot than expectation β). Over time, these expectations normalise with reality, with increased use of social cues tending to lead to improved learning performance for the human interactant.

becomes more contingent, as in [12]. Expectation discrepancy would consequently lead to changes in the cognitive reaction over time as expectations change, and vary based on individuals, contexts, and so on; this is reflected in Figure 1 and will be expanded upon in Section 3.

Although there are many questions regarding learning in the context of HRI that remain unexplored, it would be useful to try and first create a testable hypothesis to attempt to explain why the results gathered so far are so varied. Whether this lies in social presence differences between virtual and physical robots, or in social behaviour manipulation between robot conditions, the main variable in all of the studies considered in this section is sociality. As such, we now consider how social behaviour might influence learning.

3 SOCIAL BEHAVIOUR AND LEARNING

In order to understand more about the nature of the relationship between social behaviour and learning, literature from human-human interaction (HHI) studies will now be introduced. Learning in the context of HHI has been under study for far longer than HRI, so longer-term research programmes have been carried out, and more data is consequently available.

When exploring the connection between learning and social behaviour in HHI literature, one behavioural measure repeatedly found to correlate with learning is 'immediacy'. Particularly applied to educational contexts, this concept has been long-established and validated across many cultures [18, 24] and age ranges [21]. Immediacy provides a single value definition of the social behaviour of a human in an interaction by characterising conduct in a range of verbal and non-verbal behavioural dimensions [23]. Immediacy could therefore prove a useful means of characterising robot social behaviour in HRI (as in [26]). Further, it has been shown that more immediate behaviours on the part of a human tutor increases cognitive learning gains [28]. However, the exact nature of the relationship between immediacy and cognitive learning gain is debated [5, 28].

Many HRI studies seem to implicitly assume a linear relationship between an increase in the number of social cues used or in social behaviour contingency and learning gains (or gains in related measures

such as engagement, compliance, etc). Upon reviewing the literature concerning immediacy between humans, this has sometimes found to be the case [5], but more recent work has shown that this relationship may in fact be curvilinear [6]. A curvilinear relationship could go some way to explaining the mixed results found so far in HRI studies considering task performance with respect to robot social behaviour; it is possible that some studies make the behaviour *too social* and fall into an area of negative returns.

It is hypothesised that the curvilinear nature of immediacy may have been the effect observed in the study by Kennedy *et al.* in which a 'social' robot led to less learning than a robot which was actively breaking social expectations [12]. Over the short term, the novelty of social behaviour displayed by a robot may cause this kind of curvilinear relationship as has been observed in relation to immediacy [6]. As alluded to in Section 2, humans have a set of expectations for the sociality of the robot in an interaction. We would suggest that the greater the discrepancy between these expectations and the actual robot behaviour, the more detrimental the effect on learning. Individuals will have varied expectations, which is manifested in different short-term curves (Figure 1): the short-term curve shifts such that its apex (translating to the greatest possible amount of learning in the time-frame) is at the point where the expected and actual level of social cues is most closely matched. Prior interactions and the range of expectations created could also change the shape of the short-term curve, making the apex flatter or more pronounced depending on the variety of previous experiences.

However, when considering the interaction over the longer-term, such novelty effects wear off as the human adapts to the robot and their expectations change [7, 8, 25]. In this case we suggest that substantial learning gains could be made as the robot behaviour approaches a 'human' level of social cues; having attained a reasonable matching of expectation to reality, the robot can leverage the advantages that social behaviour confers in interactions, as previously suggested [9, 26]. Beyond this level, improvement would still be found by adding more cues, but the rate of increase is much smaller as the cues will require more conscious effort to learn and interpret. These concepts are visualised in the long-term curve seen in Figure 1.

4 PERSPECTIVES

So far, we have challenged the assumption that social behaviour has a simple linear relationship with learning by providing conflicting examples from HRI literature and also by tying concepts of social behaviour to the measure of immediacy from HHI literature. Given the regular use of HHI behaviour in generating HRI hypotheses, the non-linear relationship between immediacy and learning is used to hypothesise a non-linear relationship for HRI, particularly in the short-term (Figure 1).

A series of controlled studies would be needed to verify whether these hypothesised curves are correct. One particular challenge with this is the measuring of social behaviour. It is unclear what it is to be 'more' or 'less' social, and how this should be measured. This is where we propose that *immediacy* could be used as a reasonable approximation. All factors in immediacy are judgements of different aspects of social behaviour, which are combined to provide a single number representing the overall 'immediacy' (i.e. sociality of social behaviour) of the interactant. This makes the testing of such a hypothesis possible as the social behaviour then becomes a single dimension for consideration.

Of course, there are many other issues (such as robotic platform and age of human) which would need to be explored in this context,

but with a single measure approximating sociality this would at least be possible. Providing an immediacy measure for robot behaviour makes it much easier to compare results between studies, allowing improved analysis of the impact of things such as task content and context, which are currently very difficult to disentangle when comparing results between studies. Literature from the field of Intelligent Tutoring Systems may be a useful starting point for future work to investigate specific aspects of learning activities due to their proven effectiveness across many contexts [27].

It should be noted that the aim of this paper is to highlight the potential directionality of the relationships involved between social cues and learning. There is not enough data available to represent the shape of the curves presented in Figure 1 with any great accuracy. The curves have been devised based on the few data points available from the literature, and following from concepts of immediacy and discrepancies of expectation, as explored in Sections 2 and 3.

5   CONCLUSION

We suggest that immediacy could be taken from the HHI literature to be validated and applied to HRI more extensively as it presents itself as an ideal means to facilitate comparison of highly varied social behaviour between studies. The large volume of immediacy literature in relation to learning and other contexts could also provide a firm theoretical basis for the generation and testing of hypotheses for HRI.

In this position paper we have shown through examples from HHI and HRI literature that the relationship between social behaviour and task outcome, specifically learning in the present work, for humans cannot be assumed to be linear. We hypothesise a model in which social behaviour not only has a non-linear relationship with learning, but also a relationship which changes over interaction time. Following the hypothesised model, we suggest that although in the short-term there may be some disadvantages for a robot to be maximally socially contingent, the benefits conferred by social behaviour as proposed by prior work will be seen in the long-term.

REFERENCES

[1] Wilma A Bainbridge, Justin Hart, Elizabeth S Kim, and Brian Scassellati, 'The effect of presence on human-robot interaction', in *Proc. RO-MAN'08*, pp. 701–706. IEEE, (2008).

[2] Mike Blow, Kerstin Dautenhahn, Andrew Appleby, Chrystopher L Nehaniv, and David C Lee, 'Perception of robot smiles and dimensions for human-robot interaction design', in *Proc. RO-MAN'06*, pp. 469–474. IEEE, (2006).

[3] Cynthia Breazeal, Nick DePalma, Jeff Orkin, Sonia Chernova, and Malte Jung, 'Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment', *Journal of Human-Robot Interaction*, **2**(1), 82–111, (2013).

[4] Robert M Carini, George D Kuh, and Stephen P Klein, 'Student engagement and student learning: Testing the linkages*', *Research in Higher Education*, **47**(1), 1–32, (2006).

[5] Laura J Christensen and Kent E Menzel, 'The linear relationship between student reports of teacher immediacy behaviors and perceptions of state motivation, and of cognitive, affective, and behavioral learning', *Communication Education*, **47**(1), 82–90, (1998).

[6] Jamie Comstock, Elisa Rowell, and John Waite Bowers, 'Food for thought: Teacher nonverbal immediacy, student learning, and curvilinearity', *Communication Education*, **44**(3), 251–266, (1995).

[7] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C Schultz, et al., 'Designing robots for long-term social interaction', in *Proc. IROS'05*, pp. 1338–1343. IEEE, (2005).

[8] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro, 'Interactive robots as social partners and peer tutors for children: A field trial', *Human-Computer Interaction*, **19**(1), 61–84, (2004).

[9] A. Kendon, 'Some functions of gaze-direction in social interaction.', *Acta Psychol (Amst)*, **26**(1), 22–63, (1967).

[10] James Kennedy, Paul Baxter, and Tony Belpaeme, 'Children comply with a robot's indirect requests', in *Proc. HRI'14*, (2014).

[11] James Kennedy, Paul Baxter, and Tony Belpaeme, 'Comparing robot embodiments in a guided discovery learning interaction with children', *International Journal of Social Robotics*, (2014).

[12] James Kennedy, Paul Baxter, and Tony Belpaeme, 'The robot who tried too hard: Social behaviour of a robot tutor can negatively affect child learning', in *Proc. HRI'15*, (2015).

[13] Hatice Kose-Bagci, Ester Ferrari, Kerstin Dautenhahn, Dag Sverre Syrdal, and Chrystopher L Nehaniv, 'Effects of embodiment and gestures on social interaction in drumming games with a humanoid robot', *Advanced Robotics*, **23**(14), 1951–1996, (2009).

[14] David R Krathwohl, 'A revision of Bloom's taxonomy: An overview', *Theory into practice*, **41**(4), 212–218, (2002).

[15] Daniel Leyzberg, Sam Spaulding, and Brian Scassellati, 'Personalizing robot tutors to individual learning differences', in *Proc. HRI'14*, pp. 423–430, (2014).

[16] Daniel Leyzberg, Samuel Spaulding, Mariya Toneva, and Brian Scassellati, 'The physical presence of a robot tutor increases cognitive learning gains', in *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, CogSci 2012, pp. 1882–1887, (2012).

[17] Rosemarijn Looije, Anna van der Zalm, Mark A Neerincx, and Robbert-Jan Beun, 'Help, I need some body: The effect of embodiment on playful learning', in *Proc. RO-MAN'12*, pp. 718–724. IEEE, (2012).

[18] James C McCroskey, Aino Sallinen, Joan M Fayer, Virginia P Richmond, and Robert A Barraclough, 'Nonverbal immediacy and cognitive learning: A cross-cultural investigation', *Communication Education*, **45**(3), 200–211, (1996).

[19] Roger K Moore, 'A Bayesian explanation of the 'uncanny valley' effect and related psychological phenomena', *Nature Scientific Reports*, **2**(864), (2012).

[20] Paul R Pintrich and Elisabeth van de Groot, 'Motivational and self-regulated learning components of classroom academic performance.', *Journal of educational psychology*, **82**(1), 33, (1990).

[21] Timothy G Plax, Patricia Kearney, James C McCroskey, and Virginia P Richmond, 'Power in the classroom VI: Verbal control strategies, nonverbal immediacy and affective learning', *Communication Education*, **35**(1), 43–55, (1986).

[22] Aaron Powers, Sara Kiesler, Susan Fussell, and Cristen Torrey, 'Comparing a computer agent with a humanoid robot', in *Proc. HRI'07*, pp. 145–152. IEEE, (2007).

[23] Virginia P Richmond, James C McCroskey, and Aaron D Johnson, 'Development of the Nonverbal Immediacy Scale (NIS): Measures of self- and other-perceived nonverbal immediacy', *Communication Quarterly*, **51**(4), 504–517, (2003).

[24] Vincent Santilli and Ann Neville Miller, 'The effects of gender and power distance on nonverbal immediacy in symmetrical and asymmetrical power conditions: A cross-cultural study of classrooms and friendships', *Journal of International and Intercultural Communication*, **4**(1), 3–22, (2011).

[25] Ja-Young Sung, Henrik I Christensen, and Rebecca E Grinter, 'Robots in the Wild: Understanding long-term use', in *Proc. HRI'09*, pp. 45–52. IEEE, (2009).

[26] Daniel Szafir and Bilge Mutlu, 'Pay attention!: Designing adaptive agents that monitor and improve user engagement', in *Proc. CHI'12*, pp. 11–20, New York, NY, USA, (2012). ACM.

[27] Kurt VanLehn, 'The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems', *Educational Psychologist*, **46**(4), 197–221, (2011).

[28] Paul L Witt, Lawrence R Wheeless, and Mike Allen, 'A meta-analytical review of the relationship between teacher immediacy and student learning', *Communication Monographs*, **71**(2), 184–207, (2004).

[29] Jamil Zaki, 'Cue integration a common framework for social cognition and physical perception', *Perspectives on Psychological Science*, **8**(3), 296–312, (2013).

Reaching and pointing gestures calculated by a generic gesture system for social robots

Greet Van de Perre*, Albert De Beir, Hoang-Long Cao, Pablo Gómez Esteban, Dirk Lefeber and Bram Vanderborght

Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium

Abstract

Since the implementation of gestures for a certain robot generally involves the use of specific information about it's 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. In this paper, we discuss the novel end-effector mode of the method, which can be used to calculate gestures whereby the position of the end-effector is important, for example for reaching for or pointing towards an object. The interesting and innovative feature of our method is its high degree of flexibility in both the possible configurations wherefore the method can be used, as in the gestures to be calculated. The method was validated on several configurations, including those of the robots ASIMO, NAO and Justin. In this paper, the working principles of the end-effector mode are discussed and a number of results are presented.

Keywords: Generic gesture system, pointing, gestures, upper body postures

1. Introduction

In today's robotics, motions are mostly preprogrammed off-line for a specific robot configuration [28][14][33], or generated by mapping motion capture data to the robot's configuration [21] [29] [7]. Since both techniques use specific information about the robot's morphology, these motions cannot be easily transferred to other robots. This issue is know as the correspondence problem [6][1]. As a result, when using a different robot platform, new joint trajectories need to be calculated and implemented. To offer another solution next to this time consuming methodology, we designed a generic method to generate gestures for different robots. The 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.

An alternative technique to generate gestures in a flexible way was proposed by Stanton et al. [27], by using neural networks to teleoperate a humanoid robot without an explicit kinematic modeling. However, this technique requires training while the method proposed here is very straightforward in use. In both [24] and [19], a gesture framework initially developed for virtual agents is applied on a humanoid robot. In [24], the speech and gesture production model developed for the virtual agent MAX is used to generate gestures for the ASIMO robot. 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 [8]. Similarly, in [19], gestures are described independently of the embodiment by specifying features as the hand shape, wrist position and palm orientation. The specifications for the hand shape and palm orientation are used to calculate values for the wrist joint and fingers. However, the angles for the shoulder and elbow joints are selected from a predetermined table listing joint values for all possible wrist positions. So although the gestures are described independently of the robot configuration, mapping these gestures to the robot requires hard coded joint information. Specifically for manipulation tasks, [18] presented a semi-general approach for

---

*Corresponding author
Email address: `Greet.Van.de.Perre@vub.ac.be` (Greet Van de Perre)

Figure 1: In the state of the art, gestures are implemented for a specific robot. We propose to use a generic method to generate gestures for different 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 for manipulation and pointing. Robots: (a) WE-4RII [15], (b) KOBIAN [33], (c) NAO [3], (d) ASIMO [23], (e) Myon [10], (f) HRP-2 [12].

generating natural arm motions for human figures. In their inverse kinematics algorithm which is based on neurophysiological findings, the problem of finding joint angles for the arm is decoupled from finding those from the wrist. The sensorimotor transformation model of [26] is used to determine the arm posture, while the wrist angles are found by assuming a spherical wrist and using orientation inverse kinematics.

The interesting and innovative aspect of the method described here is its flexibility; a maximum degree of flexibility was pursued for both the desired robot configuration as for the targeted body motion. The resulting framework allows calculating different types of gestures, including emotional expressions and pointing gestures, for a random robot configuration that can be modelled as at least one arm, a body and/or a head. Since for different types of gestures, different features are important, our method was designed to work in two modes (figure 1). 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. This paper focuses on the end-effector mode. The working principles and results of the block mode were presented in detail in a previous publication [31] and are briefly repeated in the next subsection to provide a better understanding of the global method.

1.1. Block mode

In the block mode, the method uses a set of emotional expressions, stored in a database and maps them to a selected configuration. To ensure a good overall posture, it is not sufficient to only impose the pose of the end effector, since inverse kinematics for robots with a different configuration and different relative arm lengths could result in unrecognisable global postures. Therefore, the orientation of every joint complex the robot has in common with a human needs to be imposed. To do this, we use a simplified model of the rotational possibilities of a human, which we called the base model. This model consist of four chains, namely a body, a head, and a left and right arm. Each chain consists of one or more joint blocks. The head consists of 1 block, while the body chain contains 3 blocks, each consisting of 3 joints. The arm chain consists of four blocks; the clavicle block (2 joints), elbow block (1 joint) and the shoulder and wrist block (3
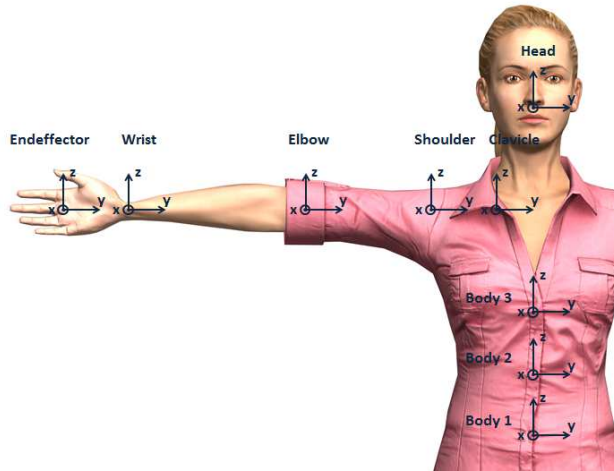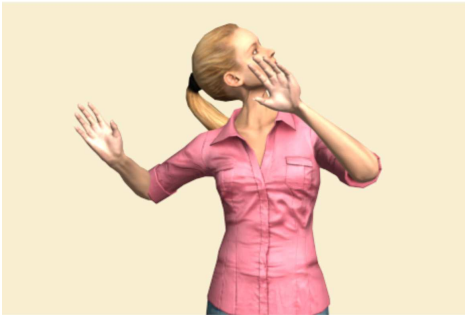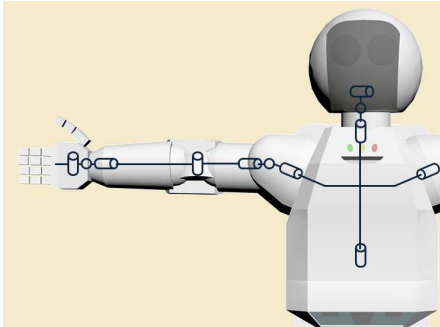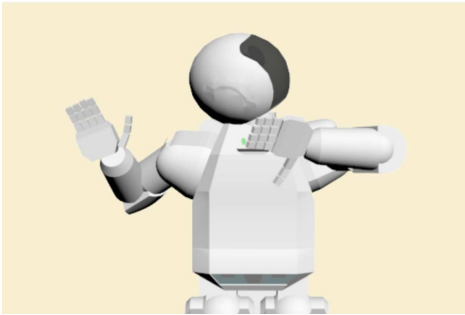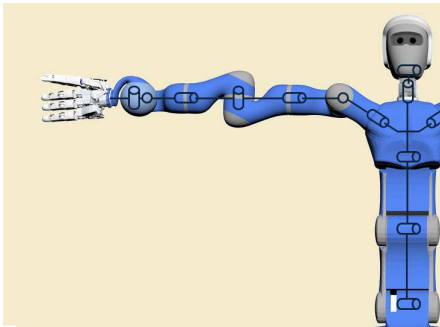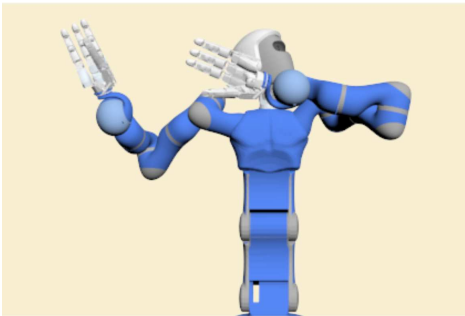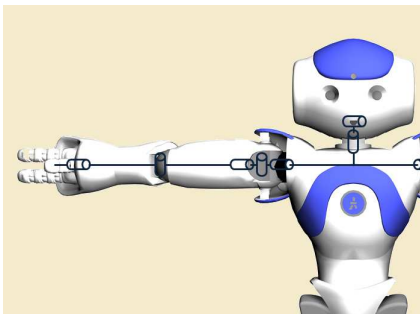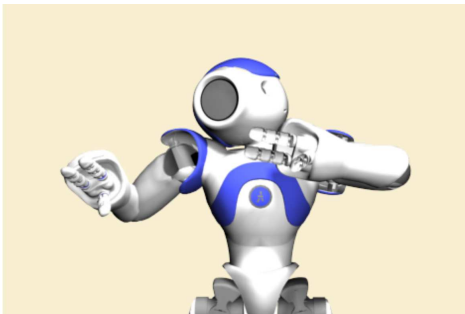
Figure 2: A reference frame was assigned to each block. For the body 1 block, the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle's reference axes are the body 3 - embedded axes. For all other blocks of the arm, the axes are the embedded axes of the previous block.

joints each). A standard reference frame was defined, whereby the x-axis is located in the walking direction and the z-axis is pointing upwards, and subsequently, a reference frame was assigned to each joint block (see figure 2). The target gestures are stored quantitatively in the database by specifying the orientation of every joint block. Information concerning the morphology of a robot or model to be used is specified by inputting its Denavit-Hartenberg (DH) parameters into the program. The different joints of the robot are grouped into the chains and blocks of the human base frame, and the rotational information from the database is mapped to the configuration to calculate a set of joint angles corresponding to the desired gesture. Table 1 shows a calculated posture for different robot configurations. The first row shows the base model with the targeted gesture, in this case, the emotional expression of disgust. The remaining of the first column shows the different joint configurations for the robots ASIMO [11], Justin [22] and NAO [9], while the second column shows the mapped posture for that configuration. The end posture is clearly recognizable, although differences in the calculated posture resulting from the different configurations can be detected. A clear example is the different placement of NAO's right wrist compared to the other models. NAO's wrist only contains the joint corresponding to the pronation and supination. Especially the absence of a joint corresponding to the flexion/extension of the wrist results in an altered placement. Furthermore, since all robots listed in the table lack the presence of the clavicle block, the complete left arm is placed a bit lower compared to the target posture shown by the human model. More information about the block mode can be found in [31].

1.2. End-effector mode

This paper will focus on the novel develloped end-effector mode of the method, which is used for end-effector dependent movements. In some situations, for example when reaching for an object, the position of the end-effector is important and specified by the user. This situation is called the *place-at* condition, whereof the working principle is covered in section 2. 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 needs to check rapidly if this point is in reach of the robot. In order to do this, it uses an approximation of the robot's workspace. Section 3 covers how this approximate workspace is determined. If the desired point is in range of the robot, a suitable trajectory towards this point needs to be calculated. This is discussed in section 4. For pointing towards an object, several end-effector poses are possible to achieve a pointing gesture to the specified target. The methodology of how a certain pose is chosen for the pointing condition is discussed in section 5. In section 6, some results of the method are listed. The paper is concluded by a short summary and a perspective of the future work in section 7.

Table 1: 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 disgust for that configuration.

Configuration		Calculated posture
Base model		
Config 2: ASIMO		
Config 3: Justin		
Config 4: NAO		

2. *Place-at* condition

2.1. 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 the hand itself. A set of joint angles corresponding to this 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 DH-parameters and rotational information 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) - 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 [25]:

$$\dot{q} = J_A^\dagger(q)\left(\dot{x}_d + K\left(x_d - x_e\right)\right) + \left(I - J_A^\dagger(q)J_A(q)\right)\dot{q}_0 \tag{1}$$

Here, $J_A^\dagger(q)$ is the Moore-Penrose pseudo inverse of the analytical jacobian $J_A(q)$. Since 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 $\left(I - J_A^\dagger(q)J_A(q)\right)$ 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 2.2):

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q}\right)^T \tag{2}$$

with k_0 a positive weight factor. The desired joint angles q are calculated by integrating $\dot{q}$ with the Runge-Kutta algorithm [2].

2.2. 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 [4], jerk [34], angular dispacement (MAD) [20] or torque [30] [13] [4]. Another possibility is to use the joint range availability (JRA) criterion [16]. Here, the algorithm will try to find an optimal humanlike posture by keeping the joints close to their central position, away from their limits [17]:

$$JRA = \sum_{i=1}^{n} w_{0,i} \frac{\left(q_i - q_{ci}\right)^2}{\left(q_{max,i} - q_{min,i}\right)^2} \tag{3}$$

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. [5] 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°, 80°and 10°, 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:

$$w = \sum_{i=1}^{n} w_{0,i} \frac{\left(q_i - q_{mi}\right)^2}{\left(q_{max,i} - q_{min,i}\right)^2} \tag{4}$$

The minimum posture angles q_{mi} used in our method are listed in table 2.

Table 2: Minimum values for the joint cost functions. The angles are defined in the reference frames connected to the human base model (see figure 2) and relative to the standard T-pose.

Block	BAU	Description	Min angle (°)
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
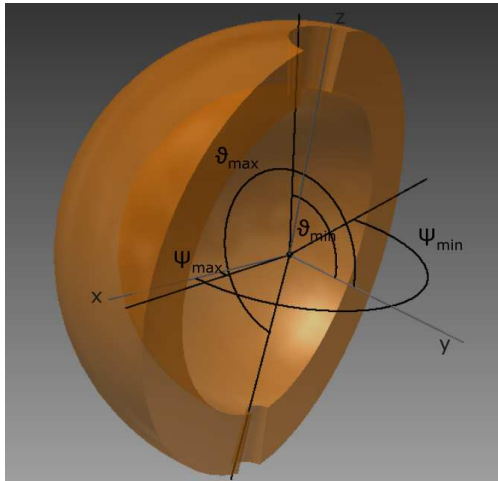
Figure 3: Example of an approximated workspace.

3. Range of the robot

3.1. 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 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 3 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 θ and azimuthal angle ϕ 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} is the maximum distance of the end-effector of the chain with respect to the shoulder base frame (see figure 4). 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 todays robotics and in any case, the range of the corresponding joint angles is limited, resulting in a negligible contribution to the workspace
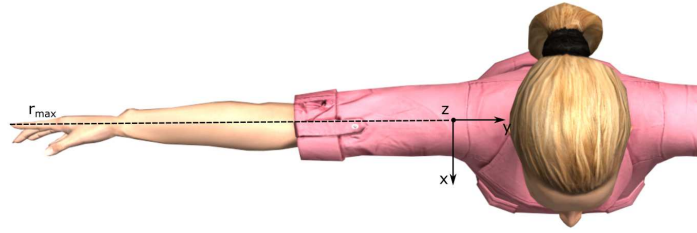
Figure 4: Determination of the maximum radius r_{max}: the maximum reachable distance of the end effector, measured from the shoulder base frame origin.
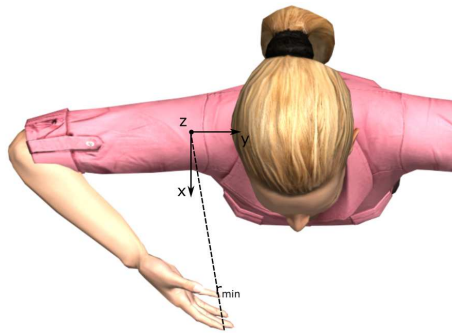
Figure 5: Determination of the minimum radius r_{min}: the elbow joint is placed in maximum flexion while the other joint angles correspond to the T-pose angles. The distance between the shoulder base frame and end-effector corresponds to r_{min}.
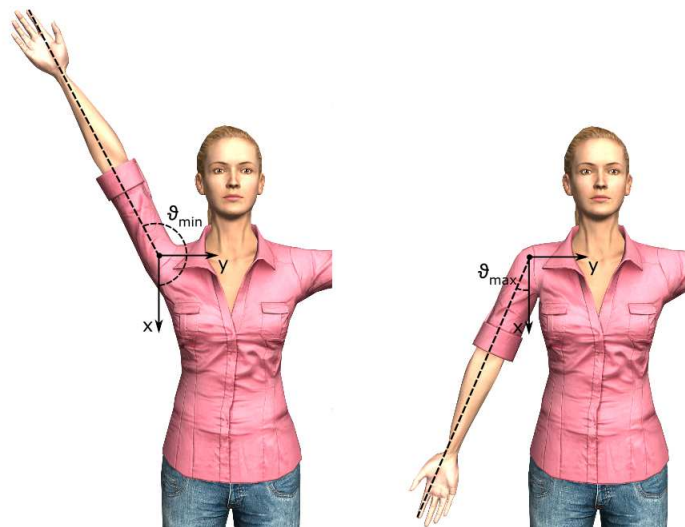
Figure 6: Calculation of the minimum and maximum value for θ; the arm is placed in respectively, maximum abduction and maximum adduction and the angle formed by the end-effector is calculated.
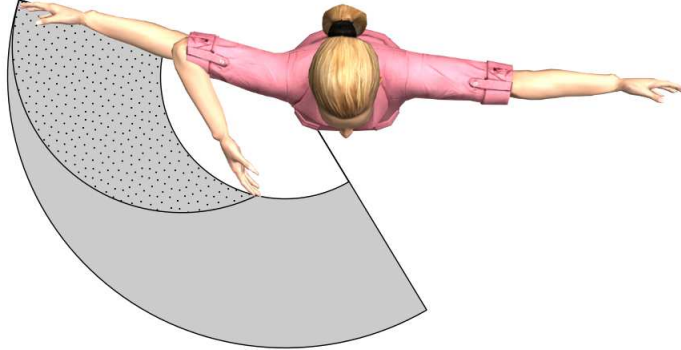
Figure 7: View of the calculated workspace in the horizontal plane crossing the shoulder base frame. The area covered by a circle with the centre point located in the elbow base frame and radius equal to the length of the lower arm (dotted surface) needs to be subtracted from the workspace (grey surface). N.b: this is the calculated workspace used in the pointing-condition, where the end-effector is the finger.

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 5). To specify the minimum and maximum polar angle θ, 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 6). In a similar way, the minimum and maximum values for the azimuthal angle ϕ 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 it 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 7). 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, θ and ϕ, also the length of the lower-arm is calculated and used in the determination of the range.

Figure 8 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 result from not taking into account the configurations involving elbow flexion for maximum shoulder flexion. The eliminated circle around the elbow is clearly visible in the right bottom corner of the $xy-$ cross section (left of figure 8). However, some blue dots are visible in this region. They origin from non-human like postures and do not contribute to proper natural trajectories.

3.2. Evaluation of specified end-effector positions

Since only four variables are used to describe the approximate workspace, it is very fast to evaluate if a certain end-effector position lies within the possible range of the robot. Therefore, the parameters r, θ and ψ corresponding to the specified 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 θ and ψ 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 lies in the hollow sphere-part, the method checks if the
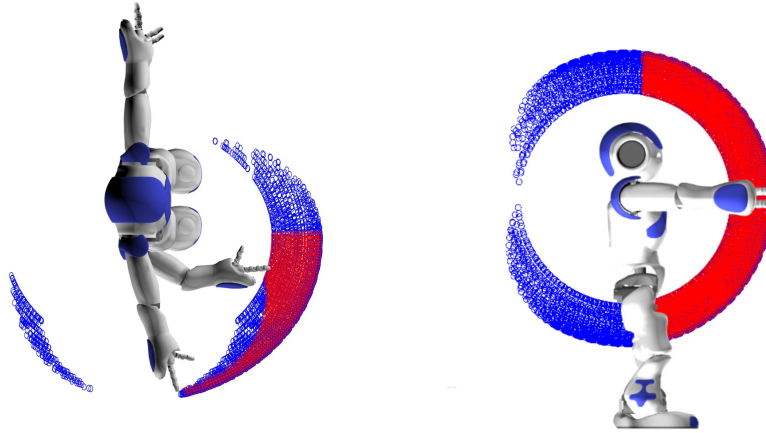
Figure 8: Calculating an approximate workspace. Blue dots indicate the real workspace, red dots the approximation. Left: xy-cross section passing the shoulder base frame. Uncovered blue dots in the top right result from not taking into account the configurations involving elbow flexion for maximum horizontal 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. Uncovered blue dots result from not taking into account the configurations involving elbow flexion for maximum shoulder flexion

position is located inside the sphere centred around the elbow. In order to do this, the desired position is rewritten in the elbow base frame, and its norm is compared to the lower-arm length.

4. Trajectory generation

When the desired end-effector position is located in the workspace of the robot, a trajectory towards this position needs to be calculated. A 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 out the reachable range of the robot. Therefore, a set of points on the trajectory are checked to lie in the workspace. In case one of these points fall out the range, an alternative trajectory is calculated. This trajectory consists of a circular arc connecting the start and end position of the end-effector. The exact shape of the path, 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.

Figure 9 summarises how a place-at gesture is calculated: firstly, the desired end-effector position is calculated in the selected arm base frame. 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 2.

5. *Pointing* condition

In the pointing condition, the pointing position is specified by the user. 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 mode, the end-effector is the index finger, in contrast to the hand itself, as used in the place-at condition. 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. To calculate the different possible postures, the end-effector is gradually virtually extended and the pointing position is imposed on the virtual end-effector. For every virtual length, the optimal configuration is calculated using the algorithm discussed in section 2. 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 10 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.
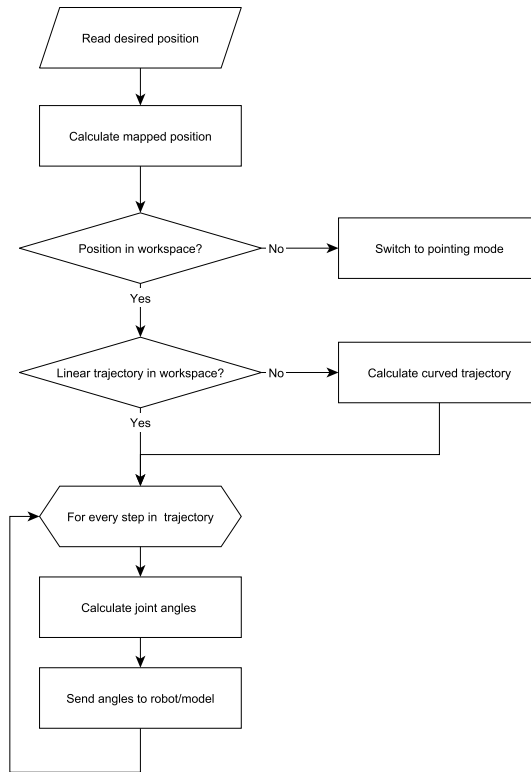
Figure 9: Schematic representation of the work flow for the place-at condition.

A simplified diagram of the complete work flow of the program is visualized in figure 11. The method firstly verifies which mode the user would like to use. When using the block mode, the orientation information for the desired gestures is loaded from the database and mapped to the selected configuration. When using the end-effector mode, the method checks which condition is enabled. When a pointing gesture towards a specific position is desired, the optimal end posture according to the principle of minimal deviation from the neutral posture is firstly determined. However, in case of a place-at condition, a suitable trajectory is calculated directly to the mapped end-effector position, provided that it 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.

6. Results

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 12. The arm consists of a 2 DOF clavicle, 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist (virtual model comes from the RocketBox Libraries [32]). 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 13 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, 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.
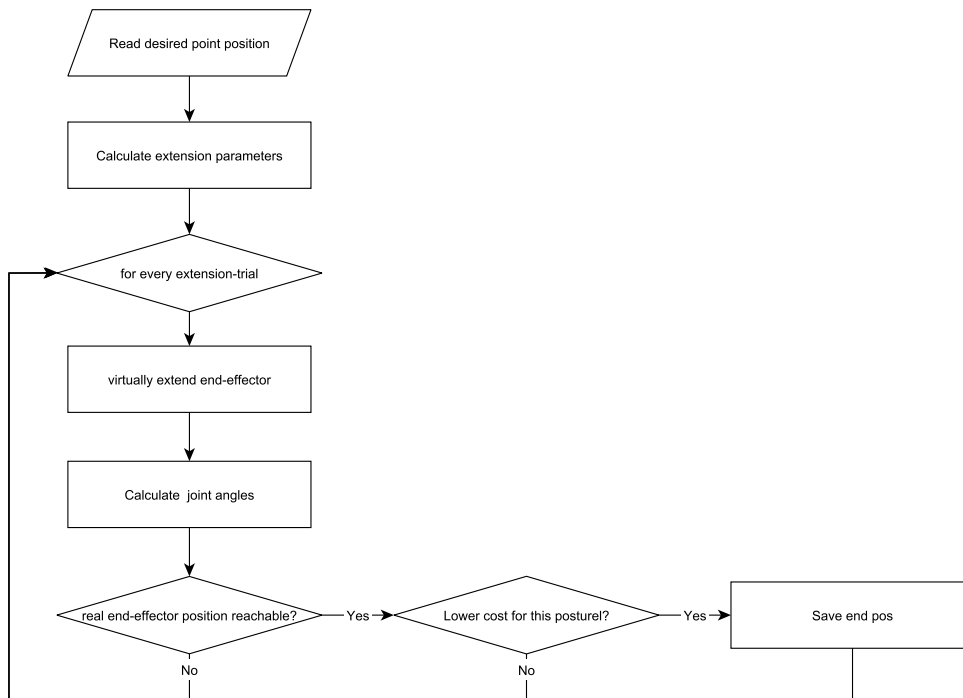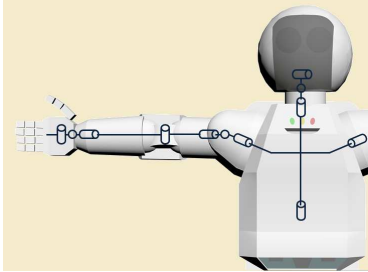
Figure 10: Schematic representation of the work flow for the pointing condition.

6.2. Results for the Pointing condition

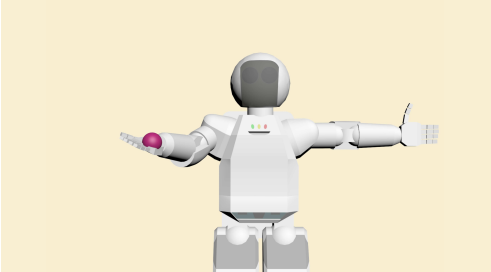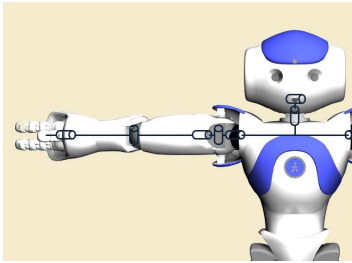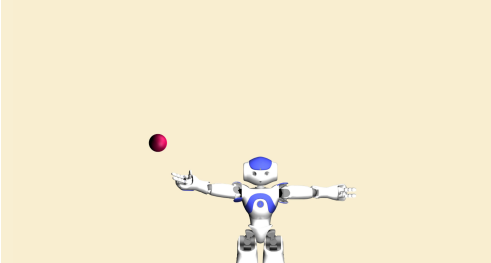As discussed in section 5, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function. In this section, we will discuss a pointing gesture to the position $(60, -20, 30)$ cm performed by the robot NAO [9] with the T-pose as the starting posture. Figure 14 shows the calculated end posture for the different iteration steps. The end-effector is virtually extended between a minimum and maximum value. The minimal 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 14 (a) shows the calculated end posture for the minimum virtual extension whereby the pointing position is visualized by a sphere. Figure 14 (d) visualizes the end posture for the maximum virtual extension, while Figure 14 (b) and Figure 14 (c) correspond to two intermediate values of extension. The cost function selected posture (d) as the optimal end posture.

6.3. Place-at condition imposed on different configurations

Table 3 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 [11]. 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 [9]. NAO is considerably smaller than the previous models, and as a result, the maximum 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.

11
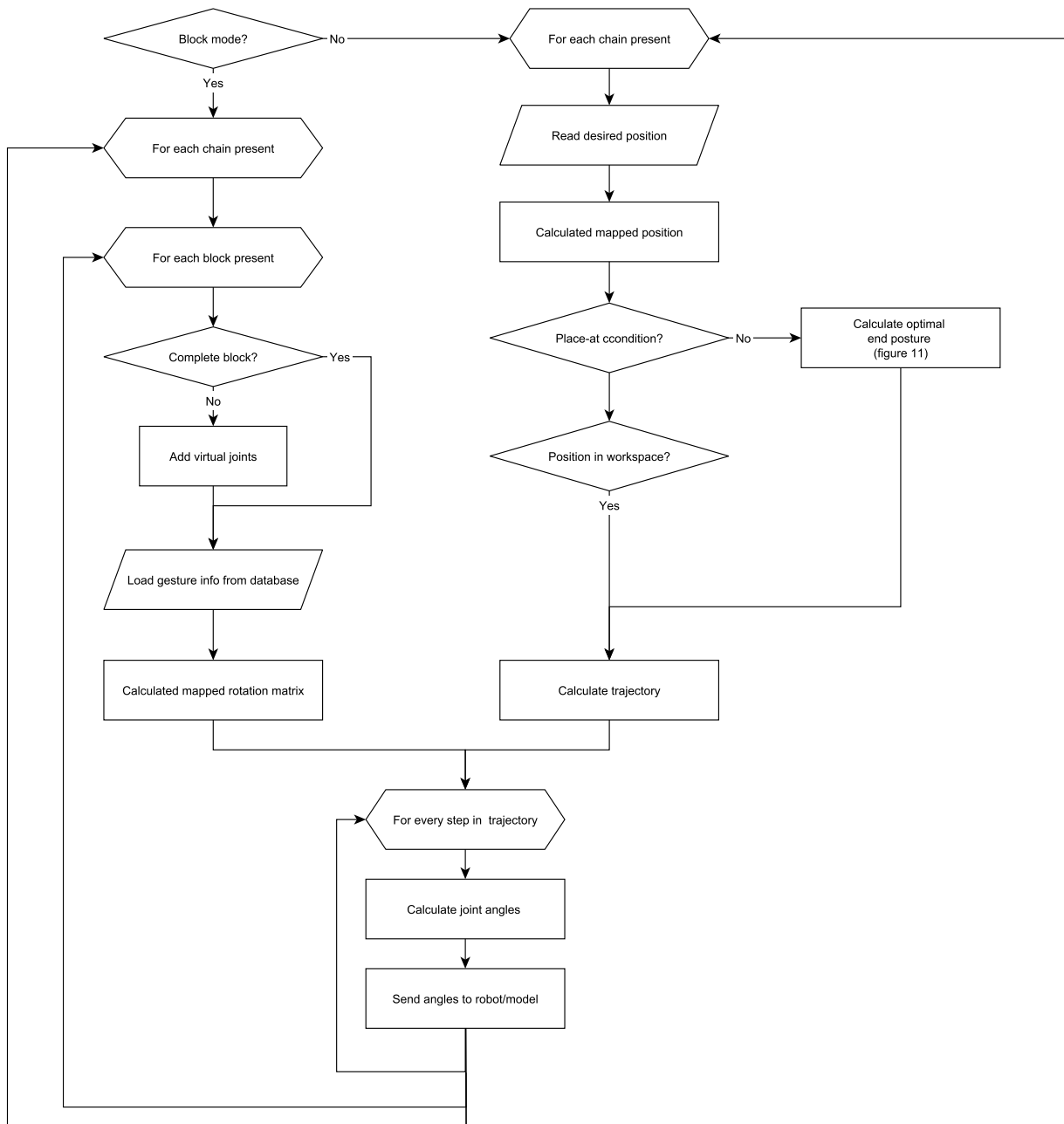
Figure 11: Simplified work flow of the complete method.
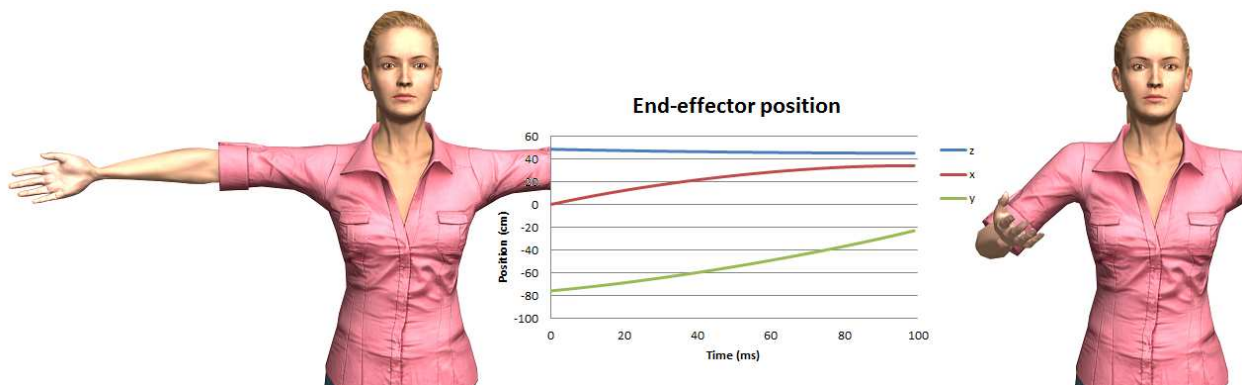
Figure 12: Calculated trajectory for a place-at task for the right hand with a start position of $(0, 76, 48)$ cm and end position of $(34, -23, 45)$ cm. Left: start pose. Right: end pose. Middle: plot of the calculated end-effector position with respect to the time.
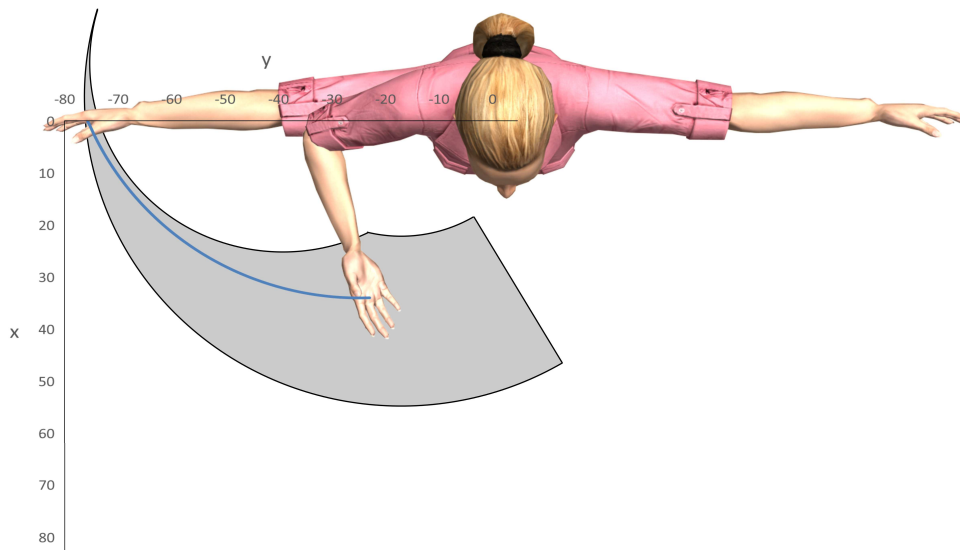
Figure 13: Top view of the calculated trajectory for a place-at task for the right hand with a start position of $(0, 76, 48)$ cm and end position of $(34, -23, 45)$ cm, superposed on the xy-cross section of the right arm workspace.

(a)   (b)   (c)   (d)
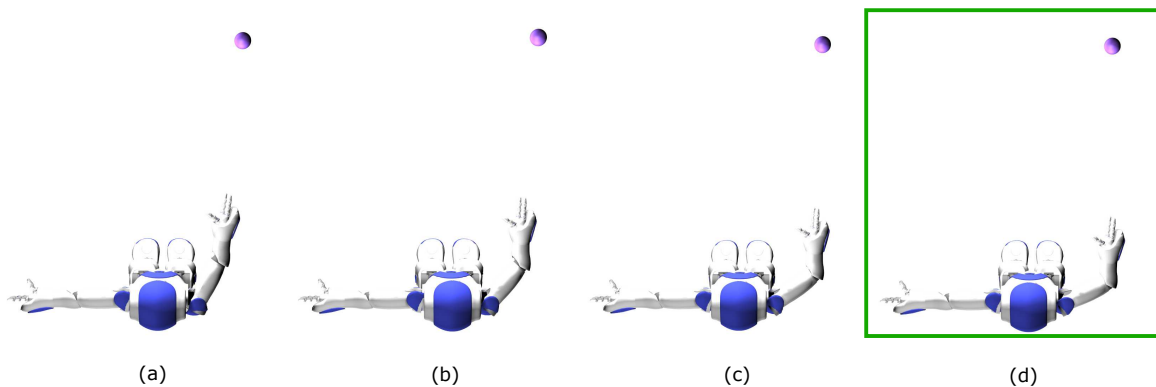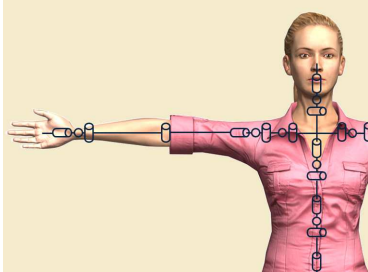
Figure 14: In the pointing condition, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function.(a) 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 3: 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
Config 1: 9 DOF arm		
Config 2: ASIMO		
Config 4: NAO		

7. Conclusions and future work

This paper discussed the novel end-effector mode of a generic method for the generation of gestures. 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 morphological information of a robot chosen by the user. The end-effector mode is used for gestures whereby the position of the end-effector is crucial. This 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 method was validated on a set of configurations, including those of the robots NAO, ASIMO and Justin. The output is here visualized using the virtual model of the robots. Current work includes implementing the method on real robots. For gestures whereby the overall posture of the arm is important, such as for emotional expressions, the block mode is used. This mode was discussed in a previous publication [31]. Future work includes mixing the two working modes, to allow combining different types of gestures. In the current implementation, when using the end-effector mode for one arm, the joint angles of the other chains are kept according to the last imposed posture. But when mixing the two modes, it will be possible to perform, for example, a pointing movement (calculated by the end-effector mode) while expressing happiness with the remaining chains (calculated by the block mode).

8. Acknowledgments

[1] Alissandrakis, A., Nehaniv, C. L., Dautenhahn, K., 2002. 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.

[2] Ascher, U. M., Petzold, L. R., 1998. Computer methods for ordinary differential equations and differential-algebraic equations. Vol. 61. Siam.

[3] Belpaeme, T., Baxter, P. E., Read, R., Wood, R., Cuayáhuitl, H., Kiefer, B., Racioppa, S., Kruijff-Korbayová, I., Athanasopoulos, G., Enescu, V., et al., 2012. Multimodal child-robot interaction: Building social bonds. Journal of Human-Robot Interaction 1 (2), 33–53.

[4] Biess, A., Liebermann, D. G., Flash, T., 2007. 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.

[5] Cruse, H., Brüwer, M., Dean, J., 1993. 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.

[6] Dautenhahn, K., Nehaniv, C. L., 2002. The correspondence problem. MIT Press.

[7] Do, M., Azad, P., Asfour, T., Dillmann, R., 2008. Imitation of human motion on a humanoid robot using non-linear optimization. In: 8th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids. pp. 545–552.

[8] Gienger, M., Janssen, H., Goerick, C., 2005. Task-oriented whole body motion for humanoid robots. In: Humanoid Robots, 2005 5th IEEE-RAS International Conference on. IEEE, pp. 238–244.

[9] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B., 2009. Mechatronic design of nao humanoid. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, pp. 769–774.

[10] Hild, M., Siedel, T., Benckendorff, C., Thiele, C., Spranger, M., 2012. Myon, a new humanoid. In: Language Grounding in Robots. Springer, pp. 25–44.

[11] Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T., May 1998. The development of honda humanoid robot. In: IEEE International Conference on Robotics and Automation (ICRA 1998). Vol. 2. pp. 1321–1326.

[12] Hirukawaa, H., Kanehiroa, F., Kanekoa, K., Kajitaa, S., Fujiwaraa, K., Kawaia, Y., Tomitaa, F., Hiraia, S., Taniea, K., Isozumib, T., Akachib, K., Kawasakib, T., Otab, S., Yokoyamac, K., Handac, H., Fukased, Y., ichiro Maedad, J., Nakamurae, Y., Tachie, S., Inoue, H., October 2004. Humanoid robotics platforms developed in HRP. Robotics and Autonomous Systems 48 (4), 165–175.

[13] Hollerbach, J. M., Suh, K. C., 1987. Redundancy resolution of manipulators through torque optimization. Robotics and Automation, IEEE Journal of 3 (4), 308–316.

[14] Ido, J., Matsumoto, Y., Ogasawara, T., Nisimura, R., 2006. Humanoid with interaction ability using vision and speech information. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,. pp. 1316–1321.

[15] Itoh, K., Miwa, H., Matsumoto, M., Zecca, M., Takanobu, H., Roccella, S., Carrozza, M., Dario, P., Takanishi, A., November 2004. Various emotional expressions with emotion expression humanoid robot WE-4RII. In: IEEE Technical Exhibition Based Conference on Robotics and Automation. pp. 35–36.

[16] Jung, E. S., Kee, D., Chung, M. K., 1995. Upper body reach posture prediction for ergonomic evaluation models. International Journal of Industrial Ergonomics 16 (2), 95–107.

[17] Klein, C. A., Blaho, B. E., 1987. Dexterity measures for the design and control of kinematically redundant manipulators. The International Journal of Robotics Research 6 (2), 72–83.

[18] Koga, Y., Kondo, K., Kuffner, J., Latombe, J.-C., 1994. Planning motions with intentions. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. ACM, pp. 395–408.

[19] Le, Q. A., Hanoune, S., Pelachaud, C., 2011. Design and implementation of an expressive gesture model for a humanoid robot. In: Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on. IEEE, pp. 134–140.

[20] Marler, R. T., Yang, J., Arora, J. S., Abdel-Malek, K., 2005. Study of bi-criterion upper body posture prediction using pareto optimal sets. In: IASTED International Conference on Modeling, Simulation, and Optimization, Oranjestad, Aruba, Canada.

[21] Matsui, D., Minato, T., MacDorman, K., Ishiguro, H., 2005. Generating natural motion in an android by mapping human motion. In: IROS. pp. 3301–3308.

[22] Ott, C., Eiberger, O., Friedl, W., Bauml, B., Hillenbrand, U., Borst, C., Albu-Schaffer, A., Brunner, B., Hirschmuller, H., Kielhofer, S., et al., 2006. A humanoid two-arm system for dexterous manipulation. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on. IEEE, pp. 276–283.

[23] Salem, M., Kopp, S., Wachsmuth, I., Joublin, F., 2009. Towards meaningful robot gesture. In: Human Centered Robot Systems. Springer, pp. 173–182.

[24] Salem, M., Kopp, S., Wachsmuth, I., Joublin, F., 2010. 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).

[25] Sciavicco, L., 2009. Robotics: modelling, planning and control. Springer.

[26] Soechting, J. F., Flanders, M., 1989. Errors in pointing are due to approximations in sensorimotor transformations. Journal of Neurophysiology 62 (2), 595–608.

[27] Stanton, C., Bogdanovych, A., Ratanasena, E., 2012. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In: Proc. Australasian Conference on Robotics and Automation.

[28] Sugiyama, O., Kanda, T., Imai, M., Ishiguro, H., Hagita, N., 2007. Natural deictic communication with humanoid robots. In: IROS 2007. pp. 1441–1448.

[29] Tapus, A., Peca, A., Aly, A., Pop, C., Jisa, L., Pintea, S., Rusu, A. S., David, D. O., 2012. Children with autism social engagement in interaction with nao, an imitative robot. a series of single case experiments. Interaction studies 13 (3), 315–347.

[30] Uno, Y., Kawato, M., Suzuki, R., 1989. Formation and control of optimal trajectory in human multijoint arm movement. Biological cybernetics 61 (2), 89–101.

[31] Van de Perre, G., Van Damme, M., Lefeber, D., Vanderborght, B., 2015. Development of a generic method to generate upper-body emotional expressions for different social robots. Advanced Robotics 29 (9), 59–609.

[32] Website, http://www. rocketbox-libraries.com.

[33] Zecca, M., Mizoguchi, Y., Endo, K., Iida, F., Kawabata, Y., Endo, N., Itoh, K., Takanishi, A., 2009. 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. pp. 381–386.

[34] Zhao, J., Xie, B., Song, C., 2014. Generating human-like movements for robotic arms. Mechanism and Machine Theory 81, 107–128.

Socially Contingent Humanoid Robot Head Behaviour Results in Increased Charity Donations

Paul Wills, Paul Baxter, James Kennedy, Emmanuel Senft, Tony Belpaeme*
Centre for Robotics and Neural Systems
The Cognition Institute
Plymouth University, Plymouth, U.K.
*tony.belpaeme@plymouth.ac.uk

Abstract—The role of robot social behaviour in changing people's behaviour is an interesting and yet still open question, with the general assumption that social behaviour is beneficial. In this study, we examine the effect of socially contingent robot behaviours on a charity collection task. Manipulating only behavioural cues (maintaining the same verbal content), we show that when the robot exhibits contingent behaviours consistent with those observable in humans, this results in a 32% increase in money collected over a non-reactive robot. These results suggest that apparent social agency on the part of the robot, even when subtle behavioural cues are used, can result in behavioural change on the part of the interacting human.

Index Terms—Charity; Experimental; Robot behavior design; Quantitative field study

I. Introduction

Prior work has suggested that highly contingent robot behaviours in complex interaction scenarios leads to an impression of autonomy and life-like attributes [1]. There has been a further suggestion that non-verbal robot behaviours can lead to significantly increased persuasiveness [2], with facial expressions, and particularly gaze, found to be particularly important in human-human interactions [3]. In this contribution we assess the extent to which robot socially-contingent behaviour can alter the behaviour of interacting humans.

The domain of charity collection provides an ideal testcase for such explorations, in part due to the clear means of assessing the difference in contributions per condition (and of course the social contribution made through the money collected). There has been some limited prior work involving charity collecting robots, most notably the iCharibot [4]: in this study involving a wheeled mobile robot with simplified face displayed on a screen, interactivity was found to increase donation amounts over only attracting behaviours and a passive benchmark. The present study differs from, and extends, this prior work in a number of ways. Firstly, we employ a static robot, comprised only of shoulders and head: with a retro-projected face, however, we have the capability to implement a wide range of facial animations, including gaze behaviours (figure 1). Secondly, rather than manipulate interaction content as well as robot behaviour as in [4], we focus only on manipulating the robot's socially-contingent behaviour. In this way, we seek to assess the specific contribution of the robot behaviour on people's charity donation behaviour.
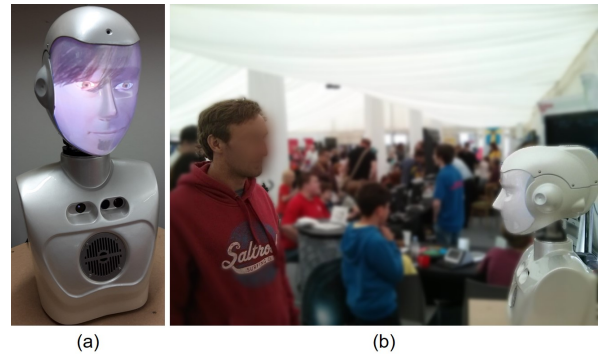
Fig. 1. (a) The robot used in the experiment with a retro-projected face; (b) the robot set up near the entrance of the event space to maximise potential interactions with the public.

The charity chosen was related to support for people with autism spectrum disorders, and their families[1]. The content of the robot's speech was based upon promotional literature from this charity. This content was the same in both experimental conditions, thus ensuring the only difference was the robot behaviour.

II. Charity Collection at a Public Event

A two-condition (contingent vs. non-contingent) study was run to explore the primary hypothesis: a robot that uses socially contingent behaviours will collect more money (for charity) than a robot without these competencies.

The study took place at a public event on a University campus, aimed at public engagement, over two days. The robot was placed adjacent to the main entrance to the event site (figure 1) to maximise potential interaction opportunities with members of the public. An experimenter was present to supervise the robot, but played no role in attracting attention to the robot or in the interactions between the robot and the public.

The robot platform used was a SociBot mini humanoid head on a pan-tilt-roll neck with a range of cameras and RGBD detectors (Engineered Arts Ltd.) and a retro-projected face system [5] which facilitates facial animation responsiveness. The robot was placed on a pedestal so as to appear at head

[1]National Autistic Society (UK): http://www.autism.org.uk/

height for the average adult. A collection bucket was placed in front of the pedestal; a separate collection bucket was used for each condition. In both conditions a set of scripted speech was used which was triggered at various points during the condition. The script consisted of information on the charity (e.g. why the money was being collected, etc), and verbal encouragement to donate. We reiterate that the speech used was the same in both conditions.

The contingent robot behaviour made use of a range of facial and gaze cues depending on who was in the environment, and how many people there were. For example, the robot would turn to look at people as they came into view, switching gaze if multiple people were present (and making use of short 0.25s glances). This was combined with blinking, eyebrow movement and pupil dilation, and reciprocal smiling if this was detected in the interacting people.

The non-contingent robot behaviour consisted of the robot uttering the scripted speech at predefined intervals, with no movement (either motor or projected), other than the lips (synchronised with the string spoken).

In order to balance exposure of each condition to the public, and given variable attendance through the day, each robot behaviour condition was alternated throughout the day, in 15 minute periods. At the end of each period, the robot would signal the experimenter (using a phrase such as "I feel sleepy"), at which point the experimenter would switch collection buckets. The robot behaviour controller switched automatically. Given the 13 hour experiment length, this meant that each condition was run on 26 separate occasions.

III. Results and Discussion

The metric with which the primary hypothesis is tested is the amount of money collected in each condition, with a total of £61.41 obtained. The results show that there was a 32.2% increase in monies collected for the contingent condition over the non-contingent condition (figure 2). This therefore suggests that the hypothesis is supported.

This result is consistent with other work: one study demonstrated that the presence of eyes on charity collection buckets increased donation rates over control non-eye images, particularly during quiet periods [6], a result that has been replicated [7]. While in the present study both conditions had eyes, the addition of contingent behaviours (to both eyes and facial features) is suggested to increase the impact of these eyes, by perhaps increasing the sense of social agency, thus increasing the effect. This suggestion is supported by human-human interaction data, which showed that mutual eye contact increased charity donations [8].

Subjectively however, a number of members of the public who engaged with the robot reported that the robot looked "creepy" or "scary", particularly in the contingent condition, where the robot attempted to make eye contact. This suggests that the mere addition of a human-like competence is not desirable [9], and that further refinement of behaviour is necessary to make it appropriate (e.g. the addition of suitable gaze-aversion strategies).
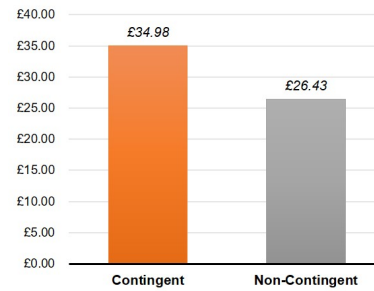
Fig. 2. Comparison of money collected in the two conditions, demonstrating a 32.2% increase for the socially contingent condition over the control.

There are a number of limitations of the study which can be rectified in future experiment replication. For example, one of these is related to data collection. The number of people who donated, and the individual amounts donated, were not recorded (for technical reasons). This makes it difficult to assess the extent of the differences in donation behaviour between the two conditions, apart from the overall donation amount, in a manner similar to prior work [4].

Nevertheless, we have demonstrated here the basis for further investigation into the role of head-based socially contingent behaviours on the donation behaviours of casual passers-by in a public space, as afforded by the retro-projected face. This suggests the positive role that apparent social agency can play on modifying the pro-social behaviour of humans.

References

[1] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, "How contingent should a lifelike robot be? The relationship between contingency and complexity," *Connection Science*, vol. 19, no. 2, pp. 143–162, 2007.

[2] V. Chidambaram, Y.-H. Chiang, and B. Mutlu, "Designing Persuasive Robots: How Robots Might Persuade People Using Vocal and Nonverbal Cues," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI '12)*, Boston, MA, USA, 2012, pp. 293–300.

[3] N. J. Emery, "The eyes have it: the neuroethology, function and evolution of social gaze." *Neuroscience and Biobehavioral Reviews*, vol. 24, no. 6, pp. 581–604, 2000.

[4] M. Sarabia, T. L. Mau, H. Soh, S. Naruse, and C. Poon, "iCharibot: design and fields trials of a fundraising robot," in *International Conference on Social Robotics*, Bristol, U.K., 2013, pp. 412–421.

[5] F. Delaunay, J. de Greeff, and T. Belpaeme, "A study of a retro-projected robotic face and its effectiveness for gaze reading by humans," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI '10)*, Osaka, Japan, 2010, pp. 39–44.

[6] K. L. Powell, G. Roberts, and D. Nettle, "Eye images increase charitable donations: Evidence from an opportunistic field experiment in a supermarket," *Ethology*, vol. 118, no. 11, pp. 1096–1101, 2012.

[7] M. Ekström, "Do watching eyes affect charitable giving? Evidence from a field experiment," *Experimental Economics*, vol. 15, no. 3, pp. 530–546, 2012.

[8] R. Bull and E. Gibson-Robinson, "The influences of eye-gaze, style of dress, and locality on the amounts of money donated to a charity," *Human Relations*, vol. 34, no. 10, pp. 895–905, 1981.

[9] J. Kennedy, P. Baxter, and T. Belpaeme, "The Robot Who Tried Too Hard: Social Behaviour of a Robot Tutor Can Negatively Affect Child Learning," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI '15)*. Portland, OR, USA: ACM Press, 2015, pp. 67–74.

Period 3

Generic method for generating blended gestures and mood expressions for social robots

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber and Bram Vanderborght

Abstract Gesturing is an important modality in human-robot interaction. Up to date, gestures are often implemented for a specific robot configuration and therefore not easily transferable to other robots. To cope with this issue, we presented a generic method to calculate gestures for social robots. The method was designed to work in two modes to allow the calculation of different types of gestures. In this paper, we present the new developments of the method. We discuss how the two working modes can be combined to generate blended emotional expressions and deictic gestures. In certain situations, it is desirable to express an emotional condition through an ongoing functional behavior. Therefore, 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 implementations were validated on different configurations, including those of NAO, Justin and ASIMO.

Keywords Generic gesture system · pointing · gestures · upper body postures · mood expression

1 Introduction

Body language is a crucial feature in human communication. Facial expressions, body posture and gestures all convey information about a person's internal state, and contribute to the overall effectiveness of communication. It has been shown that these features also benefit the interaction between humans and robots by ensuring a more flu-

Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber and Bram Vanderborght
Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium
Corresponding author: Greet Van de Perre E-mail: Greet.Van.de.Perre@vub.ac.be

ent and natural communication (Park et al 2011; Scheutz et al 2007; Salem et al 2013; Breazeal et al 2005). Different research teams have implemented gestures in robots, in the light of investigating different aspects of communication. Since these gestures are generally preprogrammed off-line for a specific robot configuration (Sugiyama et al 2007; Ido et al 2006; Zecca et al 2009), or generated by mapping motion capture data to the robot's configuration (Matsui et al 2005; Tapus et al 2012; Do et al 2008), they are robot-specific and not easily transferable to other robots. To offer a solution to this issue, which is known as the correspondence problem (Dautenhahn and Nehaniv 2002; Alissandrakis et al 2002), we developed a generic method to generate gestures for social robots. By storing target gestures independently of a configuration and calculating a mapping based on a random configuration chosen by the user, gestures can be calculated for different robots.

Since for different types of gestures, different features are important, our method was designed to work in two modes (figure 1). The *block mode* is used to calculate gestures whereby the overall arm placement is crucial, like for emotional expressions (Van de Perre et al 2015). 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 (Van de Perre et al 2016). The working principles and results of the block and end-effector mode were presented in detail in previous publications. In this paper, we focus on how these two 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.

2 Related work

Different attempts are made to ease the animation of social robots. Balit et al (2016) 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 technique to teleoperate a humanoid robot without an explicit kinematic modeling by using neural networks was proposed by Stanton et al (2012). Mühlig et al (2012) ease 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 (2007), 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 (see section 3.1). 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 (Terlemez et al 2014). In Koga et al (1994), 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 Soechting and Flanders (1989) is used to determine the arm posture, while the wrist angles are found by assuming a spherical wrist and using orientation inverse kinematics. In both Salem et al (2010) and Le et al (2011), a gesture framework initially developed for virtual agents is applied on a humanoid robot. In Salem et al (2010), 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 (Gienger et al 2005). Similarly, in Le et al (2011), 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. An interesting feature of this framework however, is the possibility of generating affective motions by modulating a neutral behavior using a set of expressivity parameters. In that way, it is possible to convey an emotional state through an ongoing functional behavior. This is indeed an important feature in human communication. Lots of research has been performed on how an emotional state is reflected in the motions generated by a human. Wallbott (1998) found that both the quantity as the quality of the emotion influences the generated body movements. A number of studies investigated the effect of different emotions on human gait (Montepare et al 1987; Crane and Gross 2007), while other focussed on addressing affect to static body postures (James 1932; Coulson 2004; Atkinson et al 2004). A number of researchers focussed on affective arm movements (Pollick et al 2001) and whole body motion (De Meijer 1989; Castellano et al 2007), whereof a number of researches were directed to the effect of affect on dance (Dittrich et al 1996; Castellano et al 2007). Using the knowledge obtained by these numerous researches, it is possible to create behaviors conveying emotional information by modifying neutral motion patterns (see section 5).

As mentioned above, our developed system uses two separate modules to calculate different types of gestures. Next to modulating a certain neutral gesture into affective motions, it is also possible to combine different types of gestures into one blended gesture. 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. How the modes are combined to generate such a blended body motion is handled in section 4. But first, to get a better understanding of the method, the working principles of the two modes are briefly repeated in section 3. In situations where it is desirable to express an emotional condition not by using explicit bodily expressions, it can be expressed by modifying an ongoing functional behavior. How this is implemented in the method is handled in section 5. A number of re-
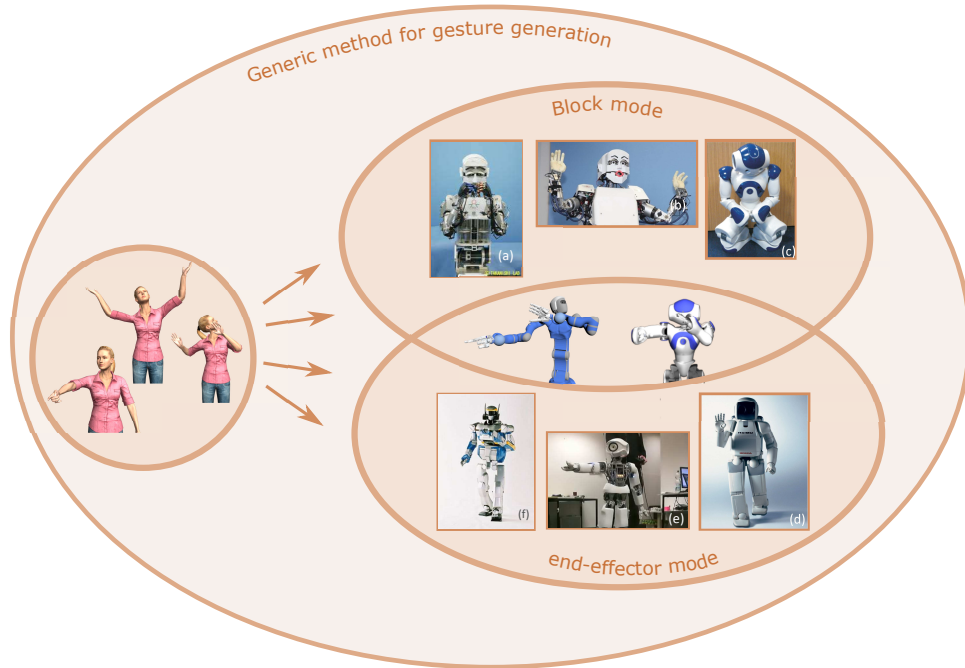
Fig. 1 In the state of the art, gestures are implemented for a specific robot. We propose to use a generic method to generate gestures for different robot. The method uses a human base model to store target gestures independently of any configuration in a database, and to calculate a mapping at runtime, 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 for deictic gestures. This paper focusses on how the two modes can be combined to generate blended emotional and deictic gestures, and how information concerning the emotional state can be used to modulate functional behaviors into affective motions. Robots: (a) WE-4RII (Itoh et al 2004), (b) KOBIAN (Zecca et al 2009), (c) NAO (Belpaeme et al 2012), (d) ASIMO (Salem et al 2009), (e) Myon (Hild et al 2012), (f) HRP-2 (Hirukawaa et al 2004).

sults are discussed throughout section 4 and 5. We conclude this paper by a summary and some details about current developments in section 6.

3 Working principles of the method

To ensure a generic method usable for different kind of robots, the framework was developed without using any kind of robot morphology. Instead, a simplified model of the rotational possibilities of a human is used; the base model. Firstly, a set of Body Action Units (BAU's) was defined, based on the human terms of motion. The defined BAU's are listed in Table 1. The units were grouped into different *blocks*, corresponding to one human joint complex, such as the shoulder or the wrist. These blocks can subsequently be grouped into three body parts, namely the head, body and arm, which we refer to as *chains*. In that way, our human base model was defined. A standard reference frame was defined, whereby the x-axis is located in the walking direction and the z-axis is pointing upwards, and subsequently, a reference frame was assigned to each joint block (see figure 2). 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 information and the configuration's Denavit-Hartenberg (DH) parameters into the program, whereby the different joints of the robot are grouped into the chains and blocks of the human base frame. As such, the method can be used for any robot that consists at least of one arm, a body, or a head.

3.1 Block mode

The block mode is used for gestures whereby the overall placement of the arms is important, such as for emotional expressions. In this mode, the method uses a set of target gestures stored in a database and maps them to a selected configuration. To ensure a good overall posture, it is not sufficient to only impose the pose of the end effector, since inverse kinematics for robots with a different configuration and different relative arm lengths could result in unrecognisable global postures. Therefore, the orientation of every joint complex the robot has in common with a human needs to be imposed. Hence, the target gestures are stored

Table 1 The Body Action Coding System

Chain	Block	BAU	Description
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
	Shoulder	9	Flexion/extension of shoulder
		10	Abduction/adduction of shoulder
		11	Inward/outward medial rotation
	Elbow	12	Flexion/extension of elbow
	Wrist	13	Pronation/supination of elbow
		14	Flexion/extension of wrist
		15	Abduction/adduction of wrist

Fig. 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.

in the database by specifying the orientation of every joint block i of the base model using the orthopaedic angles (Kadaba et al 1990) 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). To make a robot or model perform a selected expression, a mapped rotation matrix for every present joint block is calculated by combining the information from the database and the morphological data specified by the user:

$$R_i = {}^{b,i}R_{st} \cdot R_{i,des} \cdot {}^{st}R_{e,i} \qquad (1)$$

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 $i+1$.

These mapped matrices serve as input for an inverse kinematics algorithm to calculate the necessary joint angles to make the specified robot configuration perform the desired expression. Using the Runge-Kutta algorithm (Ascher and Petzold 1998), for every block, the angle values q are obtained from their derivatives $\dot{q}$, calculated by the following algorithm (Sciavicco 2009):

$$\dot{q} = J_A^\dagger(q)\left(\dot{x}_d + K\left(x_d - x_e\right)\right) + \left(I - J_A^\dagger(q)J_A(q)\right)\dot{q}_0 \quad (2)$$

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 $J_A(q)$. Since only rotational information is imposed, $J_A(q)$ is reduced to its rotational part only. x_e is the current end effector pose; i.e. the current $zyx-$Eulerangles, and K a positive definite gain matrix. Since the different blocks are treated separately, no redundancy is present, causing the second term $\left(I - J_A^\dagger(q)J_A(q)\right)\dot{q}_0$ to be zero.

3.2 End-effector mode

The end-effector mode is used for gestures whereby the position of the end-effector is crucial, like for deictic gestures. In some situations, for example when reaching for an object, the position of the right and/or left hand is important and specified by the user. This situation is called the *place-at* condition. The specified position then serves as a basis to calculate the necessary end-effector position for the selected chain, which is used as input for the same inverse kinematics algorithm as used in the block mode (equation 2). While in the block mode, a constraint is imposed on the end-effector of every block and the inverse kinematics algorithm is used to calculated the joint angles of every block separately, in the end-effector mode a constraint is imposed on the end-effector of the chain, and the algorithm is used to calculate the joint angles of the chain as a whole. Since in the end-effector mode the position is specified, the desired end effector pose x_d is limited to positional information only, reducing $J_A(q)$ to its translational part only. 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 second term of equation 2 will differ from zero, activating the influence of $\dot{q}_0$ on the calculated joint speeds. $\dot{q}_0$ introduces the cost function $w(q)$:

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^T \tag{3}$$

with k_0 a positive weight factor. For the cost function w, we decided to work with a slightly adapted form of the joint range availability (JRA) criterion (Jung et al 1995), whereby an optimal human like posture is calculated by keeping the joints close to a selected set of minimum posture angles (see our previous publication Van de Perre et al 2016):

$$w = \sum_{i=1}^{n} w_{0,i} \frac{(q_i - q_{mi})^2}{(q_{max,i} - q_{min,i})^2} \tag{4}$$

Here, q_i is the current value of joint i and q_{mi} the minimum posture angle for that joint. $q_{max,i}$ en $q_{min,i}$ are the maximum and minimum joint limits, and $w_{0,i}$ a weight factor for joint i.

The *pointing* condition functions in a same way as the *place-at* condition, apart from the fact that by specifying a desired pointing position, 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. When pointing to an object, the index finger is directed towards the object, implying that for a certain position of the end-effector, the orientation needs to be 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. To calculate the different possible postures, the end-effector is gradually virtually extended and the pointing position is imposed on the virtual end-effector. For every virtual length, the optimal configuration is calculated. From the resulting collection of postures, the cost function finally selects the optimal result by comparing the total cost of every configuration.

Before calculating a trajectory to the desired end-effector position, the possibility of reaching this position by the current configuration is checked by using an approximate calculation of the workspace. If the desired end-effector position is indeed located in the workspace of the robot, a suitable trajectory towards this position is be calculated. In case of a reaching gesture towards a position located outside the workspace, the pointing-condition can be activated, and a trajectory towards a suitable posture for a pointing gesture is calculated instead.

4 Blended gestures

4.1 Priority levels

During natural communication, humans use and combine different types of gestures. By combining the two modes of our method presented above, 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 gestures 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. Firstly, for an emotional expression, the priority level for each chain is set on the basic value. For a deictic gesture, the priority level of the corresponding chain overrules the level of that chain for an emotional expression. In combination with a deictic gesture, the user can enable the gaze to be directed towards the point of interest. The necessary joint angles

to reach the desired head orientation are calculated using the block mode. For the head block, gazing has a higher priority level than the calculation of the necessary joint angles for an emotional expression. For every separate chain, the priority levels determine which calculation principle has to be used for the current iteration; that of the block mode, or of the end-effector mode, and thus, which constraints are loaded for the different chains: orientational information for every block composing the chain, or the desired end-effector position for the complete chain. Therefore, when gestures with different priority levels are selected with the mode mixer enabled, the imposed end-effector conditions originating from the different gestures result in a blended end posture. Figure 3 schematically summarizes how the motion mixer and the priority levels determine the imposed constraints, while figure 4 visualizes the work flow of one iteration, depending on the priority levels.

4.2 Examples of blended gestures

Figure 5 illustrates the calculation of a blended gesture for the robot NAO. Figure 5(a) shows the joint configuration of the robot, while figure 5(b) displays the calculated end posture for the emotional expression of disgust, calculated by the block mode. In figure 5 (c), the calculated end-posture for a combination of the expression for disgust and a pointing gesture with the right arm is shown. 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 when de mode mixer is enabled. NAO has an actuated head, and left and right arm. Therefore, the three corresponding chains are considered. For the expression of disgust, all present chains have the basic priority level. However, the priority of the pointing gesture for the right arm is higher than the basic level. Therefore, for the right arm chain, the end-effector mode is activated, whereby the end-effector condition is determined by the desired pointing position (see subsection 3.2). For both the head chain and the left arm chain, the block mode is activated and 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 (see section 3.1).

Figure 6 shows a blended gestures calculated for the robot Justin. As visualized in figure 6(a), the robot consists of an actuated head, body, right and left arm. Figure 6(b) shows the calculated end posture for the emotional expression of fear for

this configuration. In figure 6(c), 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. When ignoring the gaze, the same conclusions regarding the priority levels can be drawn as for the previous example. However, since the priority of gazing towards a specified position for the head overrules that of the emotional expression, there will be a difference in the imposed end-effector condition for the head chain when gazing is taken into account. For the left arm, the head and the body chain, the block mode is activated. For the arm and body, again the mapped rotation matrices corresponding to the desired emotional expression are imposed as end-effector condition. For the head chain, however, the necessary rotation matrix to obtain the desired gazing direction is imposed. As in the previous example, the correct pointing gesture for the right arm chain is calculated by the end-effector mode.

5 Mood expressions

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 line with Xu et al (2013b), we then speak of mood expressions. Amaya et al (1996) 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 Pelachaud (2009), six parameters, namely spatial extent, temporal extent, fluidity, power, overall activation and repetition were used to modify behavior animations. This behavior expressivity model was validated on the virtual agent Greta, allowing it to communicate expressive content while maintaining the semantic value of the used gestures. Yamaguchi et al (2006) investigated how the emotions joy, sadness, angriness and fear can be expressed by modifying basic motions using a set of adjectival expressions, each operating one modification parameter. Three
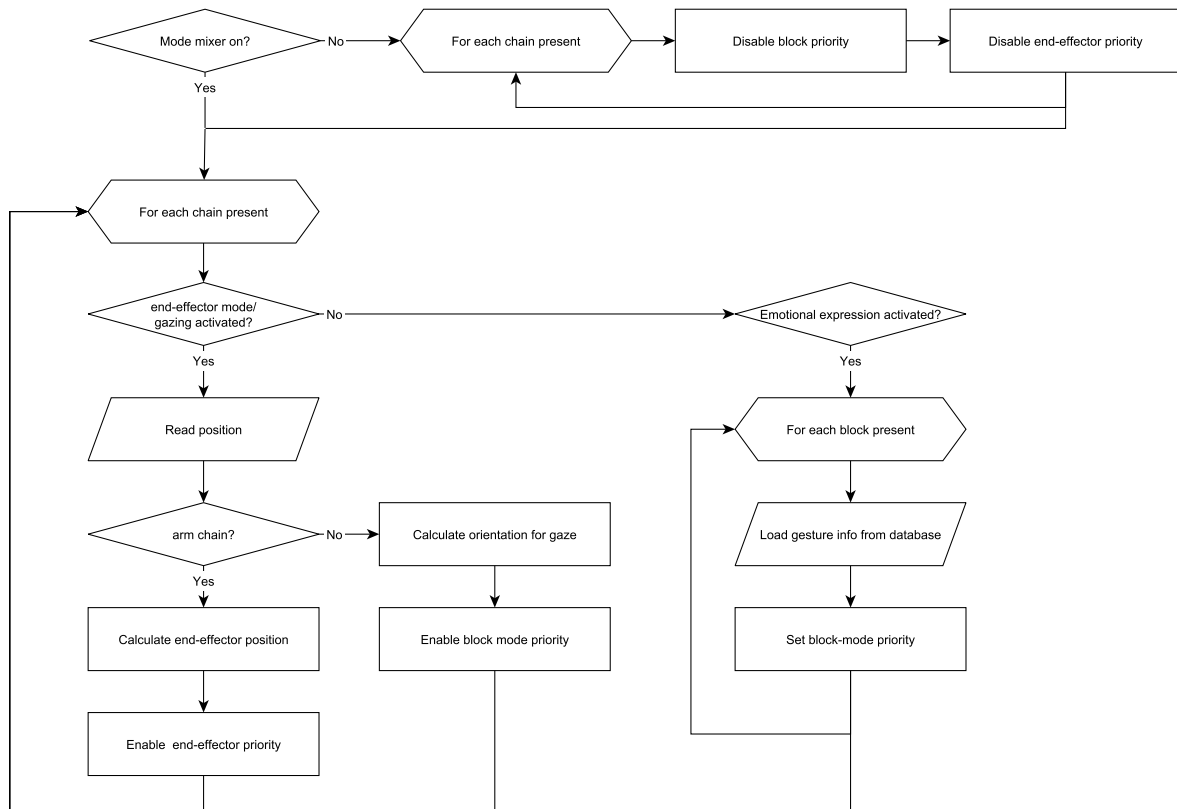
Fig. 3 Schematic representation of how the end-effector constraints are determined by the motion mixer and the priority levels.
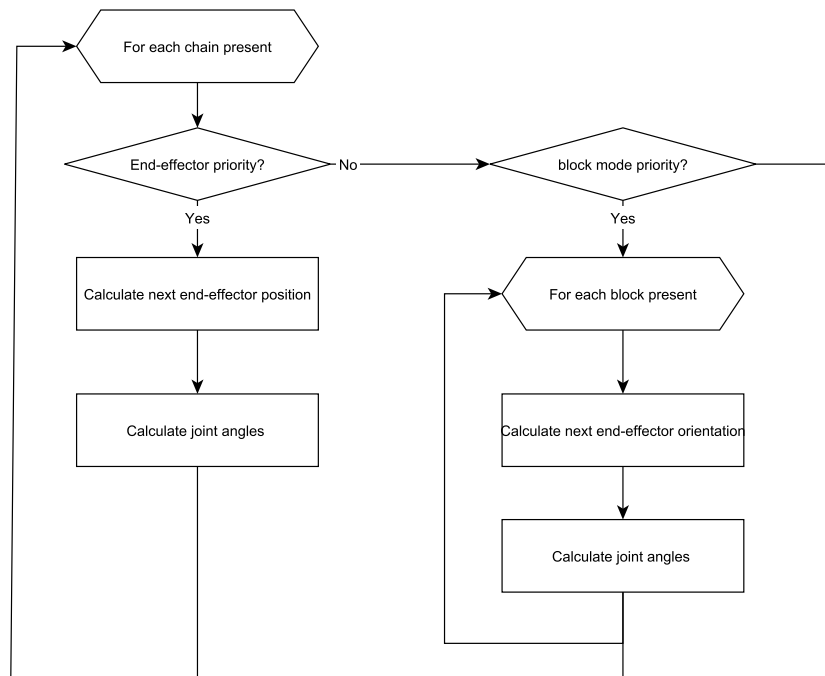
Fig. 4 Work flow of one iteration, depending on the priority levels
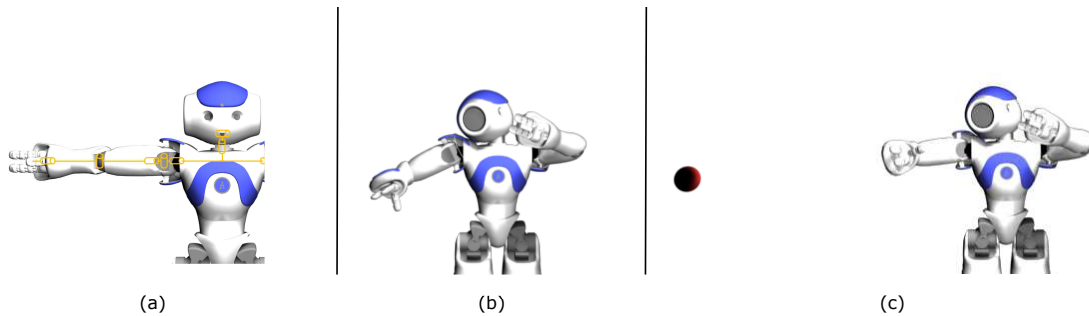
Fig. 5 Example illustrating the calculation of a blended gesture for NAO. (a) Joint configuration of the robot. (b) Calculated end posture for the emotional expression of disgust. (c) Calculated end posture for a combination of a pointing gesture with the right arm, and the emotional expression of disgust.
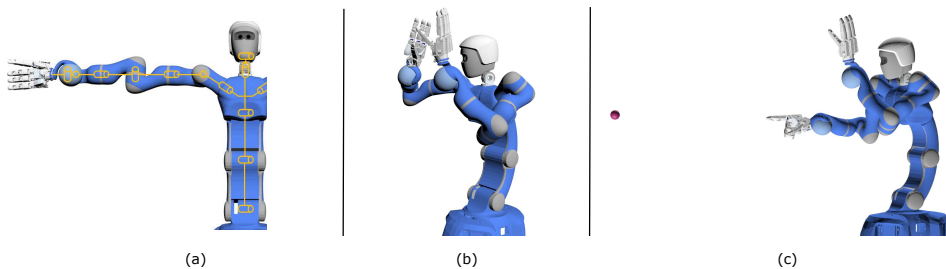
Fig. 6 Example illustrating the calculation of a blended gesture for Justin. (a) Joint configuration of the robot. (b) Calculated end posture for the emotional expression of fear. (c) Calculated end posture for a combination of a pointing gesture with the right arm, accompanied by gazing towards the pointing location, and the emotional expression of disgust.

parameters, namely the amplitude, the position and the speed, appeared to be useful to express emotions, whereby the amount of the modification can be adjusted to convey a stronger or weaker emotion. Another model was proposed by Lin et al (2009). They found that three style parameters, namely the stiffness, speed and spacial extent of the motion, can effectively generate emotional animations from an initial neutral motion. Xu et al (2013b) proposed a method for bodily mood expression, whereby a set of pose and motion parameters modulate the appearance of an ongoing functional behavior. The behavior model was applied on a pointing and waving behavior, and validated on the robot NAO. 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 (Xu et al 2013c). 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.

5.1 Generating affective gestures by influencing the motion speed

In both Xu et al (2013a) and Yamaguchi et al (2006), it was experimentally confirmed that the motion speed influences the perceived level of both valence and arousal; a fast motion is associated with a hight arousal and valence, while a slow motion is attributed to low arousal and valence values. By considering the two dimensional emotion space of valance and arousal, based on the circumplex model of affect (Posner et al 2005), we obtained an appropriate speed scaling factor for each emotion (see figure 7). 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, in order to add affectional content to it.

5.2 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 (2013a) 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 posters are related to states with a low valence (see figure
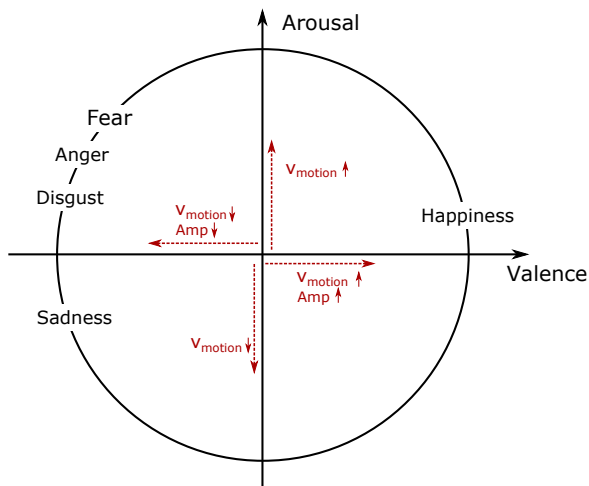
Fig. 7 Dependency of the modification factors *motion speed (v_{motion})* and *Amplitude (Amp)* on the valence and arousal value, depicted on the circumplex model of affect (Posner et al (2005)).

7). As discussed in section 3, the necessary joint angles to reach a desired posture are calculated by the inverse kinematics algorithm of equation 2 with as cost function w, a slightly adapted form of the joint range availability criterion (see equation 4). In that way, an optimal humanlike posture is calculated by keeping the joints q 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 1). 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 7) 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}*:

$$q_{ai} = q_{ai,min} + val * (q_{ai,max} - q_{ai,min}) \qquad (5)$$

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. Therefore, for BAU 10, a value of 90° 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° 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° for BAU 10, and 80° for BAU 13.

5.3 Example: deictic gesture during different states of affect

Figure 9 illustrates the results of the two subsections discussed above. A right-arm reaching gesture towards a specified position was generated for a human model during a happy, fearful and sad state. The reaching gesture is accompanied by gazing towards the imposed hand position. Figure 8 visualizes the joint configuration of the used model. The relevant features for this example are the head chain, consisting of 3 DOFs, and the right arm, consisting of 9 DOFs. As shown in the figure, the arm joints are grouped into the different blocks of the chain, resulting in a 2 DOF clavicle, 3 DOF shoulder, 1 DOF elbow and 3 DOF wrist. Since a reaching gesture is considered here, the position of the right hand palm was imposed, and joint trajectories for the different states could be generated by the end-effector mode. For every affective state, a set of postures is shown on a time line in figure 9. As discussed in subsection 5.1, the total timing of the gesture is influenced by the speed factor, of which the current 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 $0.75s$. For the same pointing gesture performed during a sad state, a low speed factor and long duration ($1.5s$) is calculated, while for the fearful state, the values are located somewhere in between (duration of $1.0s$). Next to the influence of the motion speed factor, the influence of the amplitude modification factor is also clearly visible. Since the amplitude of the posture should increase with higher valence values, an open posture is expected for the happy mood, while for the sad mood, a small, closed posture is expected. When comparing the end postures in figure 9, important differences in amplitude can indeed be distinguished. Since the position of the hand palm is imposed, the amplitude of the pos-

ture is mainly determined by the exact position of the elbow. The joint responsible for the abduction/adduction of the shoulder will therefore be the most defining. As expected, an open end posture is calculated for the happy mood, whereby the elbow is located far away from the body. For the sad mood, in contrast, the elbow is placed close to the body, generating a closed posture as expected, while for the fearful state, an intermediate posture is obtained since the value of the valence is indeed situated between those for happiness and sadness. From figure 8, it can be concluded that the fourth arm joint generates the motion related to BAU 10. The calculated angle values corresponding to this joint during the different emotional states are plotted in figure 10. To ease the comparison, the duration of each pointing gesture was rescaled to 1 second. Each joint trajectory is accompanied by a dotted line representing the affective posture angles for the corresponding affective state. As discussed above, the affective posture angle decreases for increasing values of valence. The calculated joint trajectories follow the same trend: for happiness, the reached joint angles are small, while for the sad mood, the angles reach higher values. Since the cost function to guide the angles to the active affective posture angle is only a secondary objective function, it is not expected that the end joint angle reaches this value. The cost function pushes the angle of arm joint 4 to the direction of the affective posture angle, while realizing the primary goal: generating a reaching gesture towards the imposed position. Figure 11 visualizes the calculated trajectories for arm joint 6, the joint responsible for the flexion/extension of the elbow joint (BAU 13). This joint will also influence the amplitude of the posture, however in a much lesser degree since the position of the hand is imposed by reaching gesture. A slight difference in end joint angles, following the trend of the affective posture angles can indeed be observed.

Another example of generated affective gesture is shown in figure 13. A pointing gesture to a position out of reach of the robot was calculated for the robot Justin during the same three affective states. Figure 13 shows two views of the calculated end posture for every state. As visualized in figure 12, the robot consists of an actuated head of 2 DOF's, a 3 DOF body, and a right and left arm consisting of 7 DOF. For this configuration, the joint responsible for the abduction/adduction of the shoulder (BAU 10) is arm joint 2, while motion of joint 4 results in flexion/extension of the elbow (BAU 13). As for the previous example, the calculated joint trajectories of these BAU's are plotted on a rescaled time-line (figure 14 and 15).
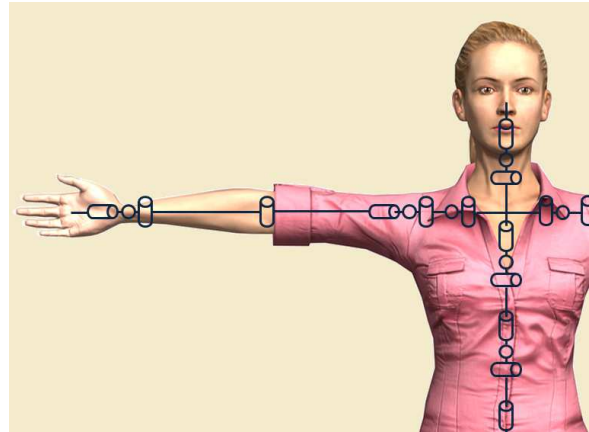
Fig. 8 Joint configuration of the model used in figure 9. Relevant for this example are the fully actuated head, consisting of 3 DOF, and the right arm, consisting of 9 DOF.

The major contribution to the differences in pose amplitude again arises from BAU 10; the reached end values for the corresponding joint clearly follow the trend of of the affective posture angles. However, this time also the difference in reached angles for BAU 13 is remarkably larger.

6 Conclusions and current work

In this paper, we presented the new developments of our generic method to generate gestures for social robots. The method was designed to work in two modes, to allow the calculation of different types of gestures. 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, such as deictic gestures. The working principles of both modes were discussed in previous publications. During human communication, different types of gestures are used and combined. In this paper we discussed how the two modes 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 in a blended end posture. In some situations however, it is desirable 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 to modu-
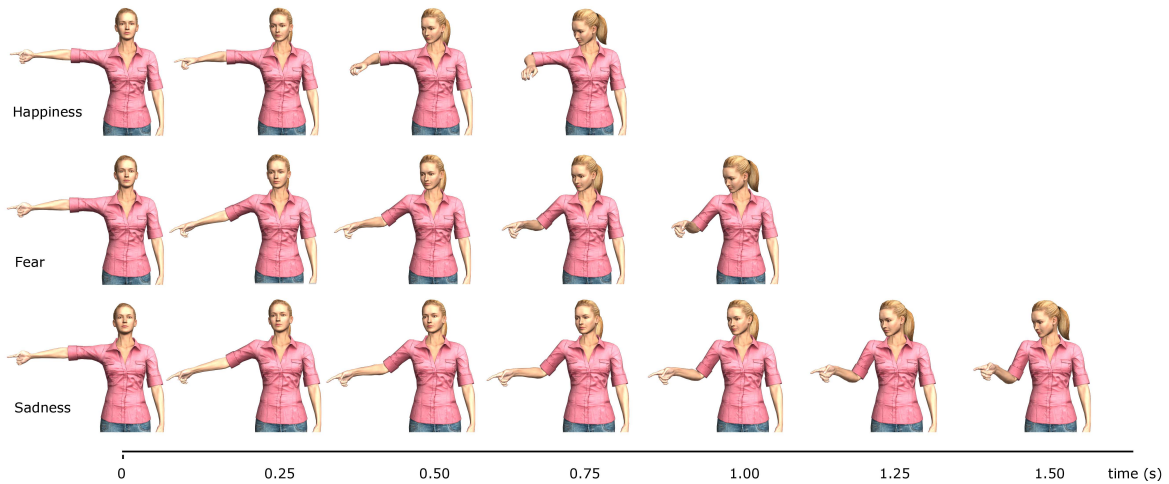
Fig. 9 Example illustrating mood expressions during a functional behavior for a human model. A reaching gesture was calculated during different affective states: happiness, fear and sadness. A set of postures for every affective state is shown on a time line to illustrate the effect of the motion speed modification factor on the calculated gesture. The effect of the amplitude modification factor is clearly visible when comparing the end postures for every mood.
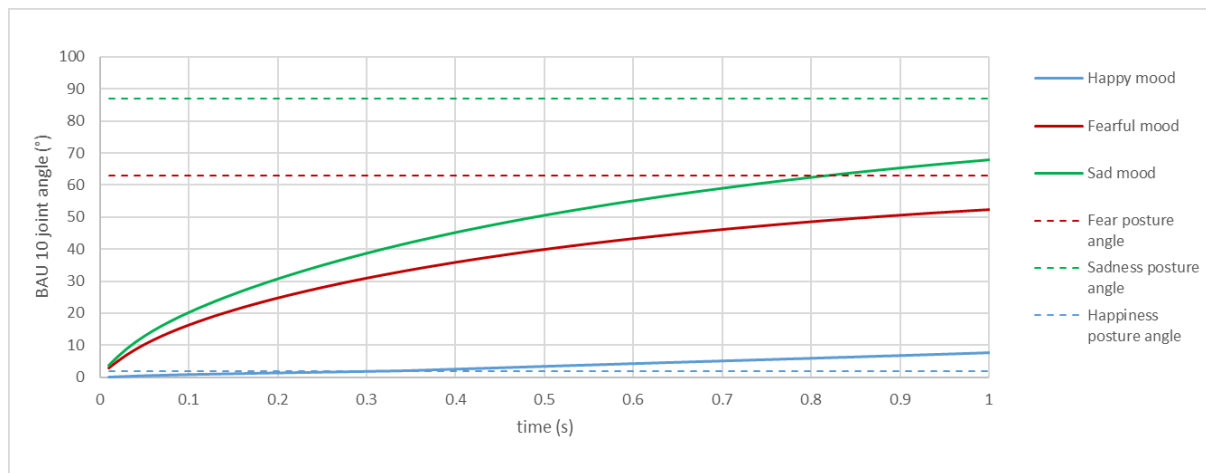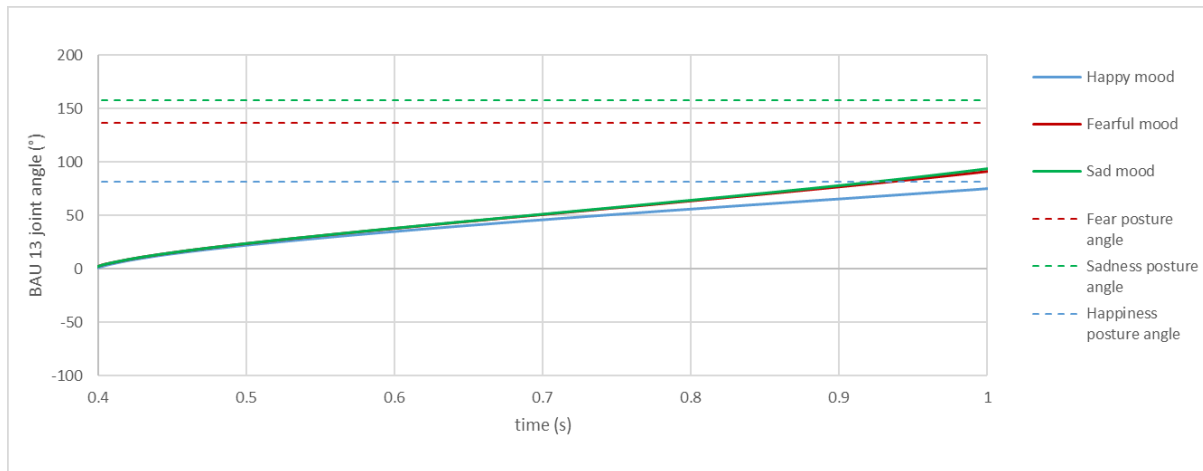

Fig. 10 Example illustrating mood expressions during a functional behavior for a human model: joint trajectories for BAU 10 for a reaching gesture during a happy, fearful and sad state. The dotted line shows the affective posture angle for BAU 10 for the corresponding mood.
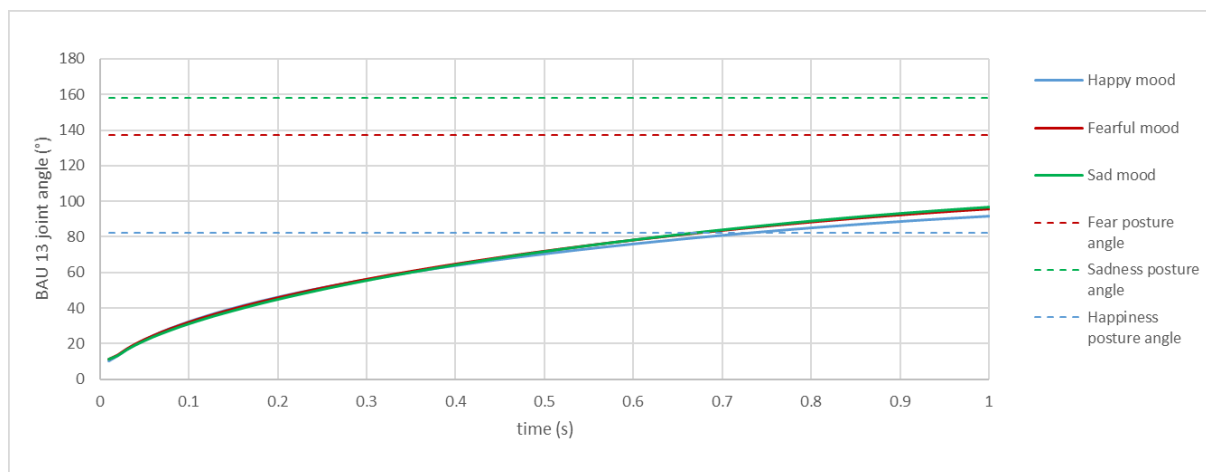
Fig. 11 Example illustrating mood expressions during a functional behavior for a human model: joint trajectories for BAU 13 for a reaching gesture during a happy, fearful and sad state. The dotted line shows the affective posture angle for BAU 13 for the corresponding mood.
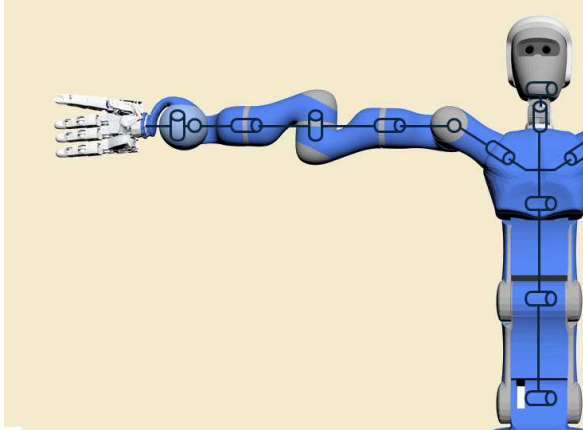
Fig. 12 Joint configuration of the robot Justin used in figure 13. Relevant for this example are the actuated head consisting of 2 DOF, and the right arm, consisting of 7DOF.

late a pointing or reaching gesture into an affective gesture by influencing the motion speed and amplitude of the posture. The new implementations were validated on different robot configurations and presented throughout this paper. Current work includes evaluating the method on the physical model of different robots.

7 Acknowledgments

References

Alissandrakis A, Nehaniv CL, Dautenhahn K (2002) 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

Amaya K, Bruderlin A, Calvert T (1996) Emotion from motion. In: Graphics interface, Toronto, Canada, vol 96, pp 222–229

Ascher UM, Petzold LR (1998) Computer methods for ordinary differential equations and differential-algebraic equations, vol 61. Siam

Atkinson AP, Dittrich WH, Gemmell AJ, Young AW, et al (2004) Emotion perception from dynamic and static body expressions in point-light and full-light displays. PERCEPTION-LONDON- 33:717–746

Azad P, Asfour T, Dillmann R (2007) Toward an unified representation for imitation of human

motion on humanoids. In: Robotics and Automation, 2007 IEEE International Conference on, IEEE, pp 2558–2563

Balit E, Vaufreydaz D, Reignier P (2016) Integrating animation artists into the animation design of social robots. In: ACM/IEEE Human-Robot Interaction 2016

Belpaeme T, Baxter PE, Read R, Wood R, Cuayáhuitl H, Kiefer B, Racioppa S, Kruijff-Korbayová I, Athanasopoulos G, Enescu V, et al (2012) Multimodal child-robot interaction: Building social bonds. Journal of Human-Robot Interaction 1(2):33–53

Breazeal C, Kidd CD, Thomaz AL, Hoffman G, Berlin M (2005) Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp 708–713

Castellano G, Villalba SD, Camurri A (2007) Recognising human emotions from body movement and gesture dynamics. In: Affective computing and intelligent interaction, Springer, pp 71–82

Coulson M (2004) Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. Journal of nonverbal behavior 28(2):117–139

Crane E, Gross M (2007) Motion capture and emotion: Affect detection in whole body movement. In: Affective computing and intelligent interaction, Springer, pp 95–101

Dautenhahn K, Nehaniv CL (2002) The correspondence problem. MIT Press

De Meijer M (1989) The contribution of general features of body movement to the attribution of emotions. Journal of Nonverbal behavior 13(4):247–268

Dittrich WH, Troscianko T, Lea SE, Morgan D (1996) Perception of emotion from dynamic point-light displays represented in dance. Perception 25(6):727–738

Do M, Azad P, Asfour T, Dillmann R (2008) Imitation of human motion on a humanoid robot using non-linear optimization. In: 8th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids, pp 545–552

Gienger M, Janssen H, Goerick C (2005) Task-oriented whole body motion for humanoid robots. In: Humanoid Robots, 2005 5th IEEE-RAS International Conference on, IEEE, pp 238–244

Hild M, Siedel T, Benckendorff C, Thiele C, Spranger M (2012) Myon, a new humanoid. In: Language Grounding in Robots, Springer, pp 25–44
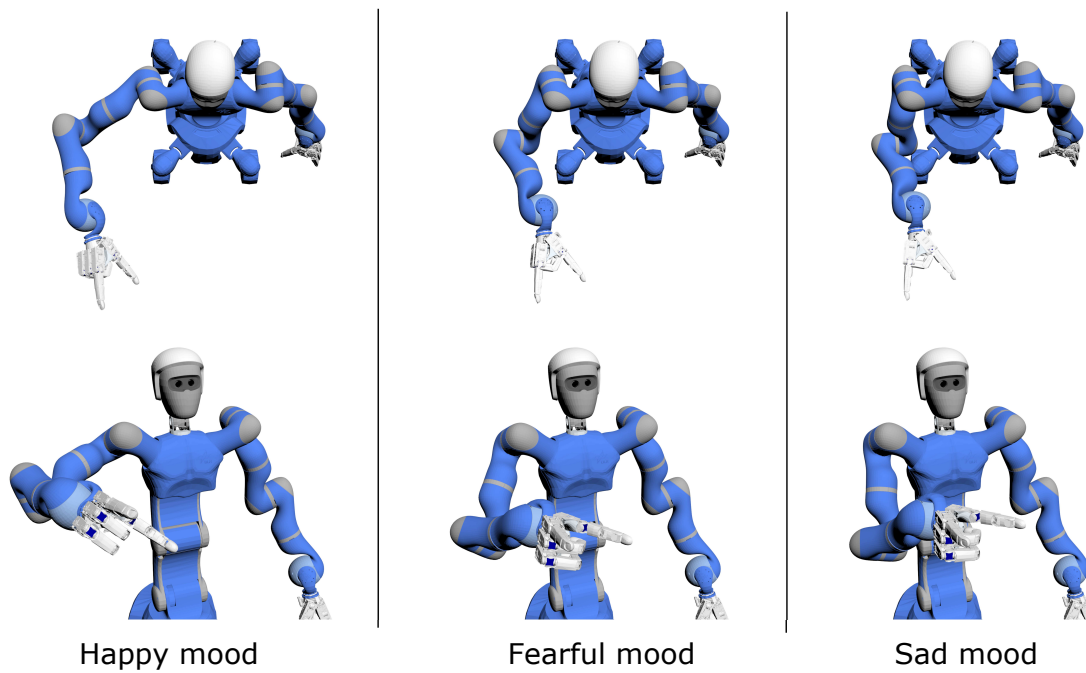
Fig. 13 Example illustrating mood expressions during a functional behavior. A pointing gesture was calculated during different affective states: happiness, fear and sadness. (a) End posture for pointing gesture during happy mood (b) End posture for pointing gesture during fearful mood (c) End posture for pointing gesture during sad mood
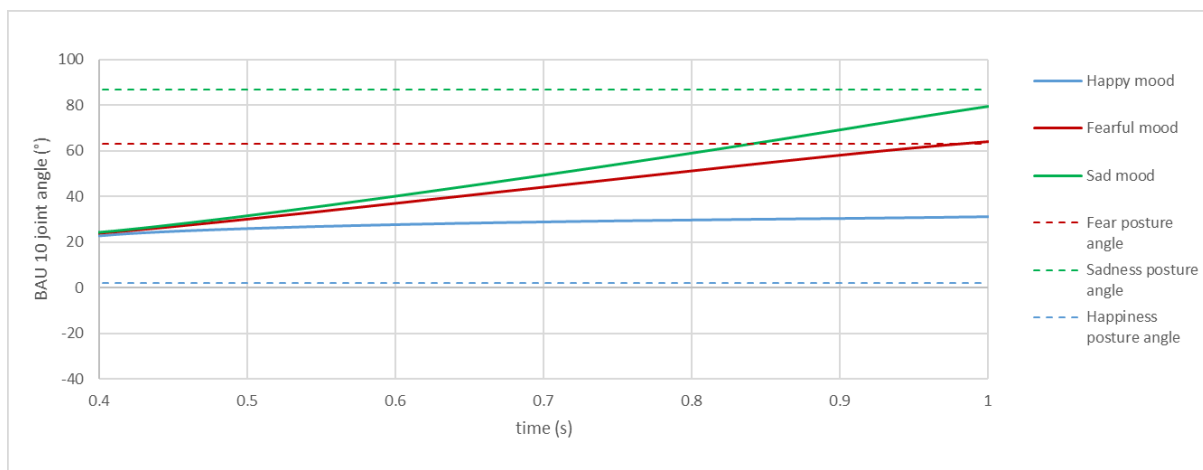
Fig. 14 Example illustrating mood expressions during a functional behavior for a the robot Justin: joint trajectories for BAU 10 for a pointing gesture during a happy, fearful and sad state. The dotted line shows the affective posture angle for BAU 10 for the corresponding mood.

Hirukawaa H, Kanehiroa F, Kanekoa K, Kajitaa S, Fujiwaraa K, Kawaia Y, Tomitaa F, Hiraia S, Taniea K, Isozumib T, Akachib K, Kawasakib T, Otab S, Yokoyamac K, Handac H, Fukased Y, ichiro Maedad J, Nakamurae Y, Tachie S, Inoue H (2004) Humanoid robotics platforms developed in HRP. Robotics and Autonomous Systems 48(4):165–175

Ido J, Matsumoto Y, Ogasawara T, Nisimura R (2006) Humanoid with interaction ability using vision and speech information. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2006), pp 1316–1321

Itoh K, Miwa H, Matsumoto M, Zecca M, Takanobu H, Roccella S, Carrozza M, Dario P, Takanishi A (2004) Various emotional expressions with emotion expression humanoid robot WE-4RII. In: IEEE Technical Exhibition Based Conference on Robotics and Automation, pp 35–36

James WT (1932) A study of the expression of bodily posture. The Journal of General Psychology 7(2):405–437

Jung ES, Kee D, Chung MK (1995) Upper body reach posture prediction for ergonomic evaluation models. International Journal of Industrial

Fig. 15 Example illustrating mood expressions during a functional behavior for the robot Justin: joint trajectories for BAU 13 for a pointing gesture during a happy, fearful and sad state. The dotted line shows the affective posture angle for BAU 13 for the corresponding mood.

Ergonomics 16(2):95–107

Kadaba MP, Ramakrishnan H, Wootten M (1990) Measurement of lower extremity kinematics during level walking. Journal of Orthopaedic Research 8(3):383–392

Koga Y, Kondo K, Kuffner J, Latombe JC (1994) Planning motions with intentions. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, ACM, pp 395–408

Le QA, Hanoune S, Pelachaud C (2011) Design and implementation of an expressive gesture model for a humanoid robot. In: 11th IEEE-RAS International Conference on Humanoid Robots, IEEE, pp 134–140

Lin YH, Liu CY, Lee HW, Huang SL, Li TY (2009) Evaluating emotive character animations created with procedural animation. In: Intelligent Virtual Agents, Springer, pp 308–315

Matsui D, Minato T, MacDorman K, Ishiguro H (2005) Generating natural motion in an android by mapping human motion. In: IROS, pp 3301–3308

Montepare JM, Goldstein SB, Clausen A (1987) The identification of emotions from gait information. Journal of Nonverbal Behavior 11(1):33–42

Mühlig M, Gienger M, Steil JJ (2012) Interactive imitation learning of object movement skills. Autonomous Robots 32(2):97–114

Park E, Kim KJ, del Pobil AP (2011) The effects of robot body gesture and gender in human-robot interaction. Human-Computer Interaction 6:91–96

Pelachaud C (2009) Studies on gesture expressivity for a virtual agent. Speech Communication 51(7):630–639

Van de Perre G, Van Damme M, Lefeber D, Vanderborght B (2015) Development of a generic method to generate upper-body emotional expressions for different social robots. Advanced Robotics 29(9):59–609

Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D, Vanderborght B (2016) Reaching an pointing gestures calculated by a generic gesture system for social robots. Robotics and Autonomous Systems (In review)

Pollick FE, Paterson HM, Bruderlin A, Sanford AJ (2001) Perceiving affect from arm movement. Cognition 82(2):B51–B61

Posner J, Russell J, Peterson B (2005) The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and psychopathology 17:715–734

Salem M, Kopp S, Wachsmuth I, Joublin F (2009) Towards meaningful robot gesture. In: Human Centered Robot Systems, Springer, pp 173–182

Salem M, Kopp S, Wachsmuth I, Joublin F (2010) 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)

Salem M, Eyssel F, Rohlfing K, Kopp S, Joublin F (2013) To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. Int Journal of Social Robotics pp pp. 1–11

Scheutz M, Schermerhorn P, Kramer J, Anderson D (2007) First steps toward natural human-like hri. Autonomous Robots 22(4):411–423

Sciavicco L (2009) Robotics: modelling, planning and control. Springer

Soechting JF, Flanders M (1989) Errors in pointing are due to approximations in sensorimotor transformations. Journal of Neurophysiology 62(2):595–608

Stanton C, Bogdanovych A, Ratanasena E (2012) Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In: Proc. Australasian Conference on Robotics and Automation

Sugiyama O, Kanda T, Imai M, Ishiguro H, Hagita N (2007) Natural deictic communication with humanoid robots. In: IROS 2007, pp 1441–1448

Tapus A, Peca A, Aly A, Pop C, Jisa L, Pintea S, Rusu AS, David DO (2012) Children with autism social engagement in interaction with nao, an imitative robot. a series of single case experiments. Interaction studies 13(3):315–347

Terlemez O, Ulbrich S, Mandery C, Do M, Vahrenkamp N, Asfour T (2014) 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, IEEE, pp 894–901

Wallbott HG (1998) Bodily expression of emotion. European journal of social psychology 28(6):879–896

Xu J, Broekens J, Hindriks K, Neerincx MA (2013a) Bodily mood expression: Recognize moods from functional behaviors of humanoid robots. In: Proceedings of International Conference on Social Robotics (ICSR), pp 511–520

Xu J, Broekens J, Hindriks K, Neerincx MA (2013b) Mood expression through parameterized functional behavior of robots. In: IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN 2013), pp 533–540

Xu J, Broekens J, Hindriks K, Neerincx MA (2013c) The relative importance and interrelations between behavior parameters for robots' mood expression. In: Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, IEEE, pp 558–563

Yamaguchi A, Yano Y, Doki S, Okuma S (2006) A study of emotional motion description by motion modification and adjectival expressions. In: IEEE Conference on Cybernetics and Intelligent Systems 2006, pp 1–6

Zecca M, Mizoguchi Y, Endo K, Iida F, Kawabata Y, Endo N, Itoh K, Takanishi A (2009) 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, pp 381–386

Generating gestures for different robot morphologies through one generic gesture system: validation on physical robots

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber and Bram Vanderborght

Abstract

To overcome the difficulties in transferring joint trajectories to different robots, we proposed the use of a generic system to calculate gestures for social robots. The developed method 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 previous work, the different modalities were validated on the virtual model of different robots. In this paper, we present the innovations made to the method to be able to use it on physical robots. 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.

1 Introduction

The correspondence problem is a well known issue in robot animation; joint trajectories calculated for a certain robot cannot be easily transferred to others because of differences in morphology. Therefore, when working with different robot platforms, new joint trajectories have to be calculated for every desired gesture, taking into account the specific morphology of the robot under consideration. To address this problem, we proposed a generic method to generate gestures for social robots. The framework of the method is constructed without using information concerning a specific robot configuration. Instead, a reference model containing the rotational possibilities of a human, the human base model, was used to set up the framework of the method. By storing target gestures independently of a configuration and calculating a mapping based on a random configuration chosen by the user, gestures can be calculated for different robots.

Different attempts are made to ease the animation of social robots. Balit et al. (2016) 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. Since the mapping of captured human motion data to a desired robot configuration is a critical issue in imitation, a number of approaches to overcome this problem were suggested in this area. Stanton et al. (2012) proposed a technique to teleoperate a humanoid robot without explicit kinematic modeling, by using neural networks. Mühlig et al. (2012) ease the correspondence problem between a human tutor and robot in imitation learning by representing

Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium

Corresponding author:
Greet Van de Perre, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium
Email: Greet.Van.de.Perre@vub.ac.be

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. (2007), 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 in the block mode of our method (see section 2.1). In Koga et al. (1994), 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. To determine the arm posture, the sensorimotor transformation model of Soechting and Flanders (1989) is used, while the wrist angles are found by assuming a spherical wrist and using orientation inverse kinematics. In Salem et al. (2010), a gesture framework initially developed for a virtual agent is applied on a humanoid robot. Using the speech and gesture production model initially developed for the virtual agent MAX, speech-accompanying gestures are generated for the ASIMO robot. 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 (Gienger et al., 2005). Similarly, in Le et al. (2011), 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.

The interesting aspect of our method, is that the framework is very generic, allowing it to be used for any social robot that consists of at least one arm, a body, or a head. Furthermore, no learning process is required, and only a limited amount of morphological information of the selected robot is needed as input for the method to calculate a variety of gestures. 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 (Van de Perre et al., 2015). 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 (Van de Perre et al., 2016b). For these types 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 (Van de Perre et al., 2016a). The different functionalities of the method were always validated on the virtual model of different robots, including NAO, Justin and ASIMO. In this paper, we focus on the necessary innovations made for the method to be used on physical robots. The method was validated on the robots NAO, Pepper and Romeo and a set of experimental results are presented.

To get a better understanding of the method, the working principles of the two modes are briefly repeated in section 2. Section 3 discusses the necessary amendments to the method to guarantee a smooth performance of the calculated gestures by physical robots. The experimental results of validating the method on NAO, Pepper and Romeo are discussed in section 4. We conclude this paper by a summary and conclusion in section 5.

2 Working principles of the method

To ensure a generic method, usable for different kind of robots, the framework was developed without using any kind of robot morphology. Instead, a model of the rotational possibilities of a human, which we called the human base model, served as reference for the method. Firstly, a set of Body Action Units (BAU's) was defined, based on the human terms of motion. The defined BAU's are listed in Table 1. The units were grouped into different *blocks*, corresponding to one human joint complex, such as the shoulder or the wrist. These blocks were subsequently grouped into three body parts, namely the head, body and arm, which we refer to as *chains*. In that way, our human base model was defined. A standard reference frame was defined, whereby the x-axis is located in the walking direction and the z-axis is pointing upwards, and subsequently, a reference frame was assigned to each joint block (see figure 1). To generate gestures for a certain robot or model, the method uses the Denavit-Hartenberg (DH) parameters (Sciavicco, 2009) of the configuration as input, whereby the different joints of the robot are grouped into the chains and blocks of the human base model. As such, the method can be used for any robot that consists at least of one arm, a body, or a head.

Table 1. The Body Action Coding System

Chain	Block	BAU	Description
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
	Shoulder	9	Flexion/extension of shoulder
		10	Abduction/adduction of shoulder
		11	Inward/outward medial rotation
	Elbow	12	Flexion/extension of elbow
	Wrist	13	Pronation/supination of elbow
		14	Flexion/extension of wrist
		15	Abduction/adduction of wrist

Figure 1. 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.

2.1 Block mode

The block mode (Van de Perre et al., 2015) is used for gestures whereby the overall placement of the arms is important, such as for emotional expressions. In this mode, the method uses a set of target gestures stored in a database and maps them to a selected configuration. To ensure a good overall posture, it is not sufficient to only impose the pose of the end effector, since inverse kinematics for robots with a different configuration and different relative arm lengths could result in unrecognisable global postures. Therefore, the orientation of every joint complex the robot has in common with a human needs to be considered. Hence, the target gestures are stored in the database by specifying the orientation of every joint block i of the base model using the orthopaedic angles (Kadaba et al., 1990) of frame $i+1$ (the base frame of block $i+1$) with respect to frame i (the base frame of block i) (see figure 1). To make a robot or model perform a selected expression, a mapped rotation matrix for every present joint block is calculated by combining the information from the database and the morphological data specified by the user:

$$R_i = {}^{b,i}R_{st} \cdot R_{i,des} \cdot {}^{st}R_{e,i} \qquad (1)$$

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 $i+1$.

These mapped matrices serve as input for an inverse kinematics algorithm to calculate the necessary joint angles to make the specified robot configuration

perform the desired expression. Using the Runge-Kutta algorithm (Ascher and Petzold, 1998), for every block, the angle values q are obtained from their derivatives $\dot{q}$, calculated using the following closed loop inverse kinematics (CLIK) algorithm (Sciavicco, 2009):

$$\dot{q} = J_A^\dagger(q)\left(\dot{x_d} + K\left(x_d - x_e\right)\right) + \left(I - J_A^\dagger(q)J_A(q)\right)\dot{q_0} \tag{2}$$

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 $J_A(q)$. Since only rotational information is imposed, $J_A(q)$ is reduced to its rotational part only. x_e is the current end effector pose; i.e. the current $zyx-$Eulerangles, and K a positive definite gain matrix. Since the different blocks are treated separately, no redundancy is present, causing the second term $\left(I - J_A^\dagger(q)J_A(q)\right)\dot{q_0}$ to be zero.

2.2 End-effector mode

2.2.1 Place-at condition The end-effector mode (Van de Perre et al., 2016b) is used for gestures whereby the position of the end-effector is crucial, like for deictic gestures. In some situations, for example when reaching for an object, the position of the right and/or left hand is important and specified by the user. This situation is called the *place-at* condition. The specified position then serves as a basis to calculate the necessary end-effector position for the selected chain, which is used as input for the same inverse kinematics algorithm as used in the block mode (equation 2). While in the block mode, a constraint is imposed on the end-effector of every block and the inverse kinematics algorithm is used to calculated the joint angles of every block separately, in the end-effector mode a constraint is imposed on the end-effector of the chain, and the algorithm is used to calculate the joint angles of the chain as a whole. Since in the end-effector mode the position is specified, the desired end effector pose x_d is limited to positional information only, reducing $J_A(q)$ to its translational part only. 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 second term of equation 2 will differ from zero,

activating the influence of $\dot{q_0}$ on the calculated joint speeds. $\dot{q_0}$ introduces the cost function $w(q)$:

$$\dot{q_0} = k_0 \left(\frac{\partial w(q)}{\partial q}\right)^T \tag{3}$$

with k_0 a positive weight factor. For the cost function w, we decided to work with a slightly adapted form of the joint range availability (JRA) criterion (Klein and Blaho, 1987), whereby an optimal human like posture is calculated by keeping the joints close to a set of minimum posture angles q_{mi} (see our previous publication (Van de Perre et al., 2016b)):

$$w = \sum_{i=1}^{n} w_{0,i} \frac{(q_i - q_{mi})^2}{(q_{max,i} - q_{min,i})^2} \tag{4}$$

Here, q_i is the current value of joint i and q_{mi} the minimum posture angle for that joint. $q_{max,i}$ and $q_{min,i}$ are the maximum and minimum joint limits, and $w_{0,i}$ a weight factor for joint i.

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. For such cases, we provided the possibility of expressing an emotional state through an ongoing reaching or pointing gesture by modulating it, using a certain set of characteristic performance parameters. Based on the results of the expressivity models proposed in Xu et al. (2013b), Amaya et al. (1996), Pelachaud (2009), Yamaguchi et al. (2006) and Lin et al. (2009), we decided to focus on two modification parameters: the motion speed and the posture amplitude. The posture amplitude refers to the spatial extent; the amount of space occupied by the body. The Body Action Units mostly influencing the openness of a posture are BAU 10 and 13; namely the units corresponding to the abduction/adduction of the shoulder and the flexion/extension of the elbow joint (see table 1). The posture amplitude is dependent on the valence value of the current affective state; open postures are coupled with affective states with high valence, while closed, low amplitude posters are related to states with a low valence (Xu et al., 2013a). By making the minimum posture angles q_{mi} used in equation 4 dependent of the valence value, the openness of the posture will depend on the current state of affect. Therefore, for the joints corresponding to BAU's 10 and 13, a linear function of the valence is provided instead of the fixed minimum posture angle, which we, in this case, call the *affective posture angle* q_{ai}:

$$q_{ai} = q_{ai,min} + val * (q_{ai,max} - q_{ai,min}) \tag{5}$$

Table 2. Minimum and maximum affective posture angles

BAU	min value (°)	max value (°)
10	90	0
13	170	80

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 1. The maximum value $q_{ai,max}$ of the affective posture angle, on the other hand, corresponds to the value associated to the maximum valence value; the value generating an open posture with high amplitude. The selected values for both BAU's are listed in table 2.

2.2.2 Pointing condition The *pointing* condition uses the same calculation principles as the *place-at* condition. However, in contrast to the specified hand position in the place-at condition, no direct constraint is imposed on the end-effector by specifying a desired pointing position. The pointing constraint can be fulfilled by a series of configurations with a specific combination of end-effector position and orientation. To calculate the optimal end posture, the end-effector is gradually virtually extended. For every virtual length, the pointing position is imposed on the current virtual end-effector and the corresponding posture is determined using equation 2. From the resulting collection of postures, the cost function (equation 4) finally selects the optimal result by comparing the total cost of every configuration.

2.3 Trajectory

After verifying if the desired end-effector position is located in the range of the robot, a suitable trajectory towards this point is is generated by calculating intermediate key frames. Also for the block mode, a trajectory between the starting pose and the desired end pose is determined by calculating the joint angles corresponding to a set of intermediate end-effector poses:

$$x_d(t_k) = f(x_{e,t_{start}}, x_{d,t_{end}}, t)$$

$$\text{with} \quad \begin{cases} x_d(t_{start}) = x_{e,t_{start}} \\ x_d(t_{end}) = x_{d,t_{end}} \end{cases} \tag{6}$$

Here, $x_d(t_k)$ is the desired pose at time t_k. Depending on the activated mode, this is a position or orientation and applied to the end-effector of a block or chain. $x_{e,t_{start}}$ is the actual end-effector starting pose of the block or chain under consideration, and $x_{d,t_{end}}$ the desired end pose. The exact nature of the trajectory function f is dependent on the activated mode and the location of the start and end pose in the workspace.

The total duration of the gesture t_{end} for an emotional expression is specified in the gesture database. For a gesture calculated by the end-effector mode, t_{end} is dependent on a speed factor, which on its turn, is dependent on the valence and arousal value of the current state of affect. As such, also the speed of the gesture serves a a modification parameter to convey an emotional state through the ongoing functional behavior.

2.4 Blended gestures

Humans use and combine different types of gestures during natural communication. By combining the two working modes of our method, blended emotional expressions and deictic gestures can be generated. 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. This means that when starting a new gesture, previously started gestures will be aborted. When enabling the mode mixer on the other hand, the priority levels determine for every separate chain which calculation principle has to be used for the current iteration; that of the block mode or of the end-effector mode, and thus, which constraints are loaded for the different chains: orientational information for every block composing the chain, or the desired end-effector position for the complete chain. Therefore, when gestures with different priority levels are selected with the mode mixer enabled, the imposed end-effector conditions originating from the different gestures result in a blended end posture (see Van de Perre et al. (2016a)).

3 Adjusting the method for physical robots

3.1 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 smoothly. Different strategies have been used in literature to implement joint limitation in existing algorithms. A well know method to avoid mechanical joint limits for redundant manipulators is the Gradient Projection Method (GPM) introduced by Liegeois (1977). 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 or affective posture (see section 2.2). 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 guaranty 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 (2012). 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:

$$q_i = \beta_i(z_i) \qquad (7)$$

whereby the domain of β 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:

$$q_i = \frac{q_{max,i} - q_{min,i}}{\pi} \, tan^{-1}(z_i) + \frac{q_{max,i} + q_{min,i}}{2} \quad (8)$$

By expressing the kinematic equations in terms of the fictive variables z_i, and calculating the real joint values q_i by equation 7, 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 2, whereafter they are transferred to the transformed joint space:

$$\dot{z} = d\beta^{-1}(z)\dot{\theta} \qquad (9)$$

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 7. Using this technique, the joint limits cannot be violated. However, when reaching a boundary, the derivative of the corresponding function β_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:

$$d\beta^\dagger_{ii} = \begin{cases} \frac{1}{\frac{\partial \beta_i}{\partial z_i}} & \text{if } \frac{\partial \beta_i}{\partial z_i} \geq \epsilon \\ 0 & \text{else} \end{cases} \qquad (10)$$

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 9 to calculate the transformed joint derivatives, an extended formula is used:

$$\dot{z} = d\beta^\dagger(z)\dot{\theta} + \left(I - d\beta^\dagger(z)J_A^\dagger J_A d\beta(z)\right) y \qquad (11)$$

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

$$\dot{z} = \dot{z}_1 + \dot{z}_2$$
$$\text{with } \begin{cases} \dot{z}_1 = d\beta^\dagger(z)\dot{\theta} \\ \dot{z}_2 = \left(I - d\beta^\dagger(z)J_A^\dagger J_A d\beta(z)\right) y \end{cases} \qquad (12)$$

3.1.1 Determination of the optimal task vector y for our method In Drexler and Harmati (2012), a linear function of z is proposed for the task vector y, which is only activated in case of low singular values:

$$y_i = \begin{cases} -m_i z_i & \text{if } d\beta_i < \epsilon \\ 0 & \text{else} \end{cases} \qquad (13)$$

with m_i a suitable weight factor for joint i. By defining the task vector using the same parameter ϵ as used as truncation bound for the calculation of the pseudoinverse $d\beta^\dagger$, the second part of equation 11 ($\dot{z}_2$), 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 vica versa.

The illustrating example in Drexler and Harmati (2012) 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 ϵ 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 11. An illustrating example of this is shown in the following figures. Consider the joint configuration shown in figure 2. 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 section 2.1. 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 3(a), while figure 3(b) 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 11, $\dot{z}_1$, pushes the joint angle value θ from its neutral 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 were $\frac{\partial \beta_1}{\partial z_1} = \epsilon$ is depicted by a red dot on figure 3(b). 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 $\dot{z}_2$, responsible for preventing the joint from crossing the joint limits. 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 ϵ. 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 drives z to a high negative value,
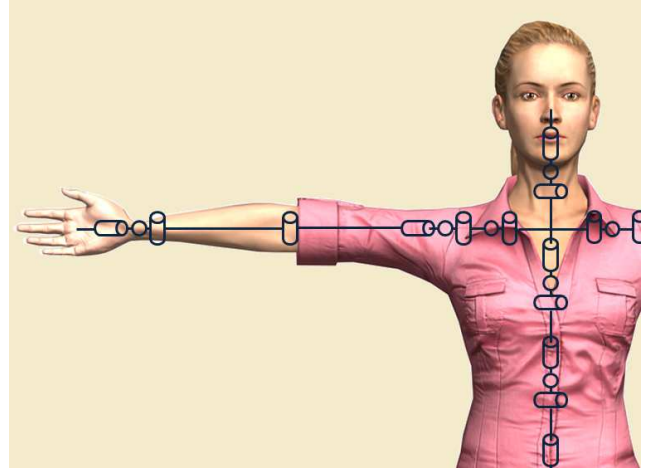
Figure 2. Joint configuration of the model used for the example in figure 3, 4 and 5. Relevant for this example is the 9 DOF right arm, consisting of a clavicle block (2 joints), a shoulder block (3 joints), an elbow block (1 joint) and a wrist block (3 joints).

projecting the joint angle θ from its upper boundary to its lower boundary. This results in a direct switch of activation from $\dot{z}_1$ to $\dot{z}_2$ to prevent the joint angle θ from crossing its lower limit value. The algorithm doesn't manage to lower the value of z enough to reactivate $\dot{z}_1$. 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 ϵ. Figure 4 (a) shows the calculated joint trajectory for the first clavicle joint for exactly the same configuration, desired gestures and parameters except for ϵ, which is now set to 10^{-3}. The corresponding transformed joint trajectories and the contributions of $\dot{z}_1$ and $\dot{z}_2$ are visualized in figure 4 (b). The initial trajectory of z is similar as for the previous example, however, the since ϵ is smaller, the term $\dot{z}_2$ is activated considerably sooner. This point is again denoted by a red dot. The joint angles θ 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, $\dot{z}_2$ will attempt to lower the value of z, until the $\frac{\partial \beta_1}{\partial z_1}$ again equals ϵ. Then, $\dot{z}_1$ 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 $\dot{z}_1$ again lies at time $t = 1.54\ s$ and is denoted by a purple dot on figure 4. From there,
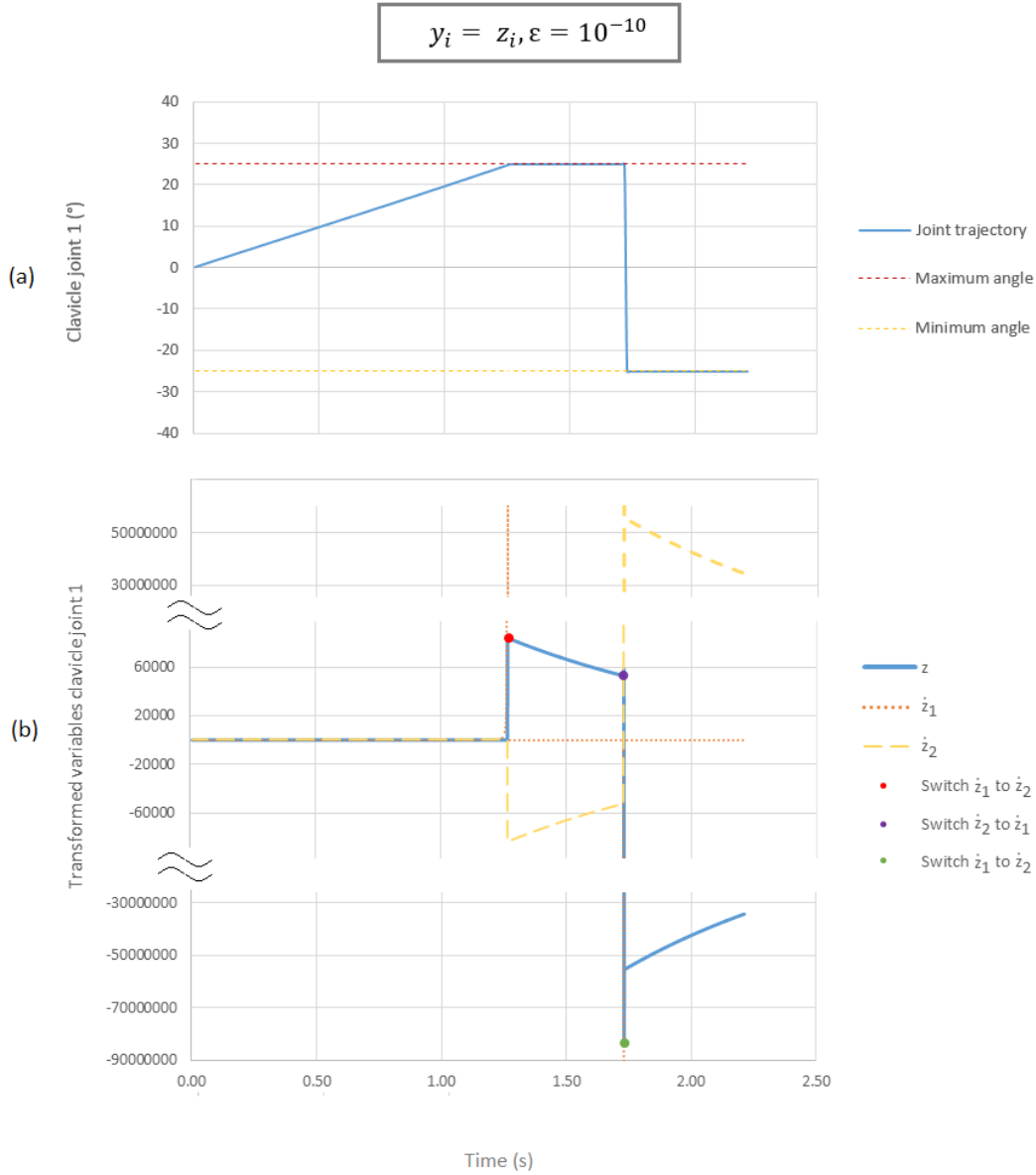
Figure 3. Calculated trajectory for the first clavicle joint of the configuration visualized in figure 2 for the execution of the emotional expression for sadness, followed by that for happiness. $y_i = -z_i$ and $\epsilon = 10^{-10}$. (a) Trajectory for the real joint angle θ. (b) Trajectory for the transformed joint variable z, accompanied by the contributions of $\dot{z}_1$ and $\dot{z}_2$.

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 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 on a change in end-effector constraint when locked around a joint limit. As could be noted in figure 4, 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 8) was used to serve as transformation β, its differential is proportional to the inverse of z^2:

$$\frac{\partial \beta_i(z_i)}{\partial z_i} = \frac{q_{max,i} - q_{min,i}}{\pi} \frac{1}{1 + z_i^2} \qquad (14)$$

$\dot{z}_1$ is therefore proportional to z^2, while $\dot{z}_2$ only to z when selecting equation 13 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
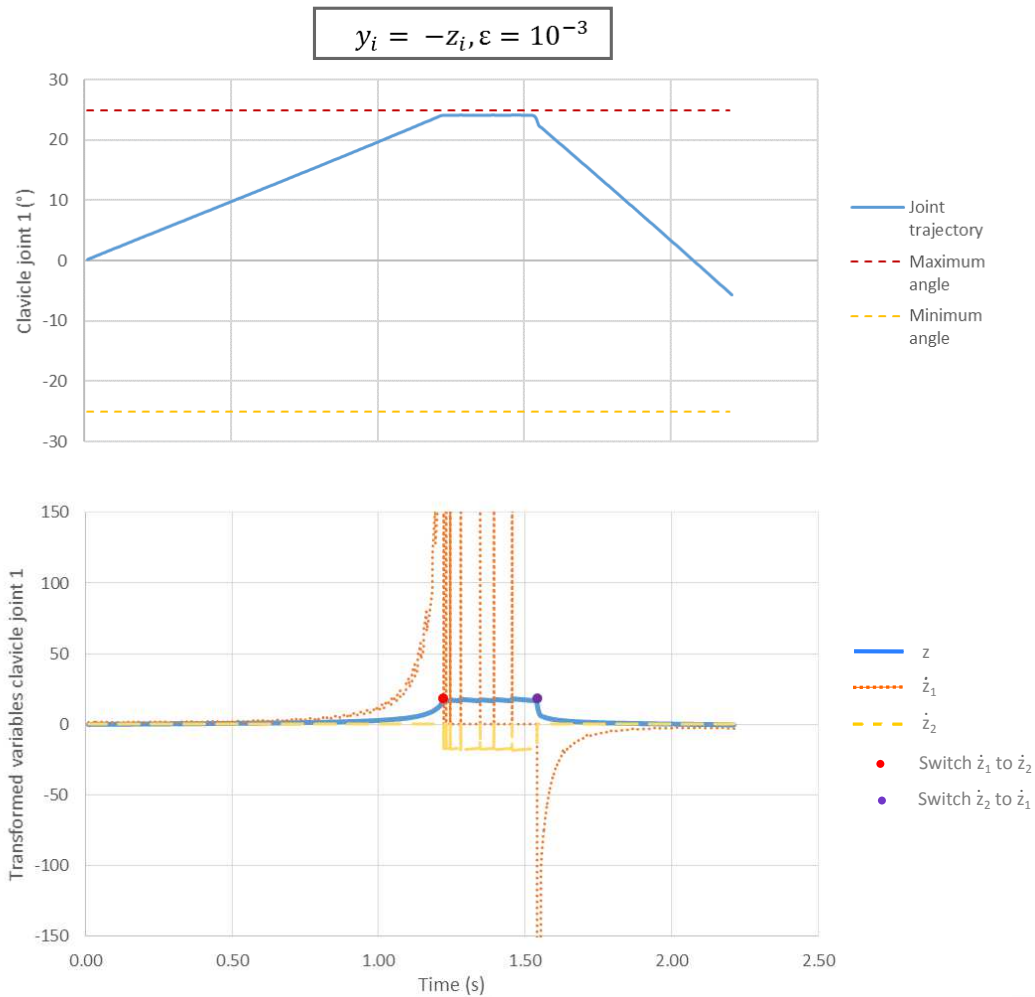
Figure 4. Calculated trajectories for the first clavicle joint of the configuration visualized in figure 2 for the execution of the emotional expression for sadness, followed by that for happiness. $y_i = -z_i$ and $\epsilon = 10^{-3}$. (a) Trajectory for the real joint angle θ. (b) Trajectory for the transformed joint variable z, accompanied by the contributions of $\dot{z}_1$ and $\dot{z}_2$.

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 4. To solve this issue, the following function can be used for y instead of equation 13:

$$y_i = \begin{cases} -sign(z_i)k_i z_i^2 & \text{if } d\beta_i < \epsilon \\ 0 & \text{else} \end{cases} \quad (15)$$

In figure 5, 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.

3.2 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 2, $\dot{q}_{calc}$, is included in the algorithm:

$$\dot{q} = \begin{cases} \dot{q}_{calc} & \text{if } -\dot{q}_{max} \leq \dot{q}_{calc} \leq \dot{q}_{max} \\ sign(\dot{q}_{calc})\dot{q}_{max} & \text{else} \end{cases}$$
$$(16)$$

As discussed in subsection 2.3, the time span in which an emotional expression should be finished is specified in the database. For pointing and reaching,
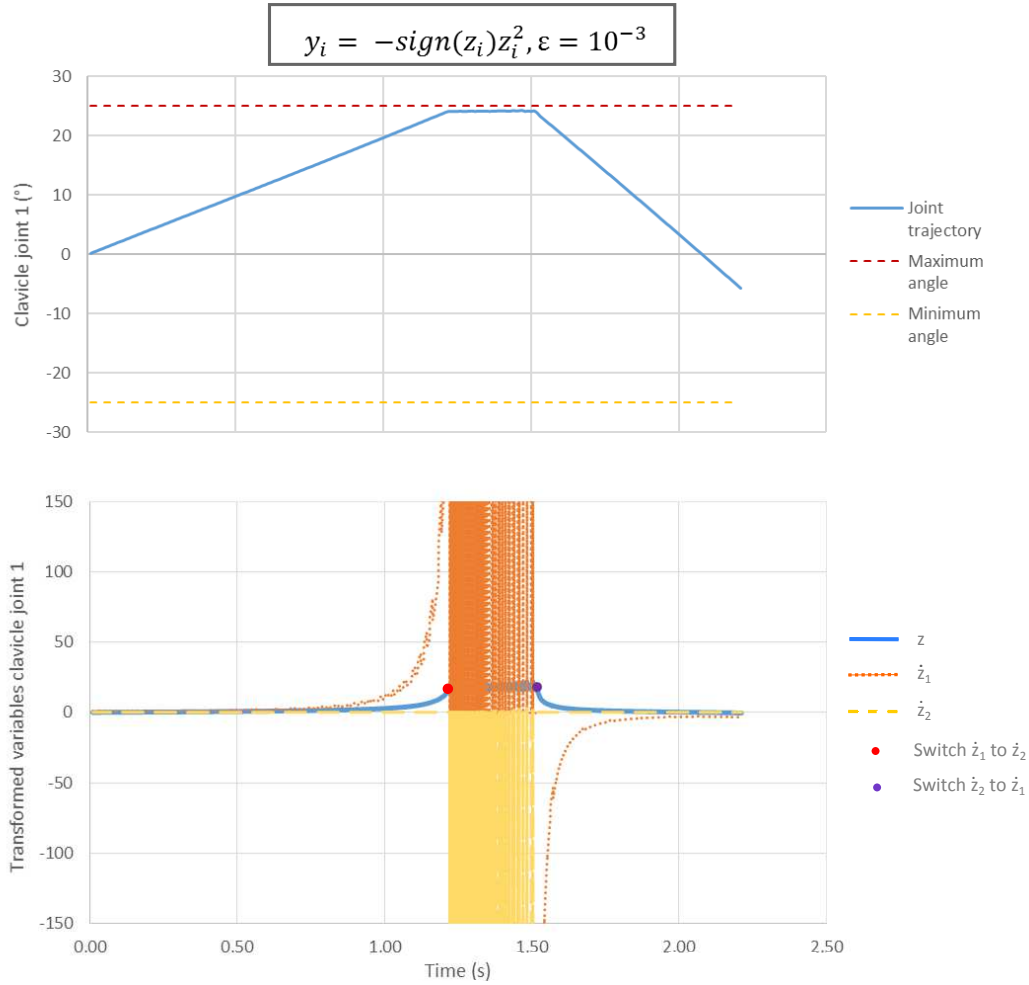
Figure 5. Calculated trajectory for the first clavicle joint of the configuration visualized in figure 2 for the execution of the emotional expression for sadness, followed by that for happiness. $y_i = -sign(z_i)z_i^2$ and $\epsilon = 10^{-3}$. (a) Trajectory for the real joint angle θ. (b) Trajectory for the transformed joint variable z, accompanied by the contributions of $\dot{z}_1$ and $\dot{z}_2$.

the total duration of the gesture is dependent of the current affective state, and also specified in the program. 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:

$$x_d(t_k) = x_{d,t_{end}} \qquad \text{if} \begin{cases} t_k > t_{end} \\ Abs(x_{e,t_k} - x_{e,t_{k-1}}) > error \end{cases}$$
(17)

To illustrate the effect of the joint speed limitation in the algorithm, a right-handed reaching gesture for

the robot Romeo was calculated. Table 3 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(a) shows the calculated speeds $\dot{\theta}$ for that joint, together with its boundaries, while Figure 6(b) 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 2.3) and imposed on the inverse kinematics algorithm (equation 2). Except for the $x-$ coordinate at the very

start of the gesture, the calculated end-effector position follows the desired trajectory perfectly.

Figure 7 visualizes the same quantities for the same reaching gesture, but now calculated with the joint speed limitation enabled. Figure 7(a) 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 7(b). For this gesture example, the algorithm succeeds in calculating a solution for the desired gesture in the imposed time span, so no time extension was necessary.

3.3 Self-collision avoidance

The possibility of self-collision is another important issue when working with physical robots. We did not implement a generic self-collision avoidance module in our method, but used the available facilities of the robots itself.
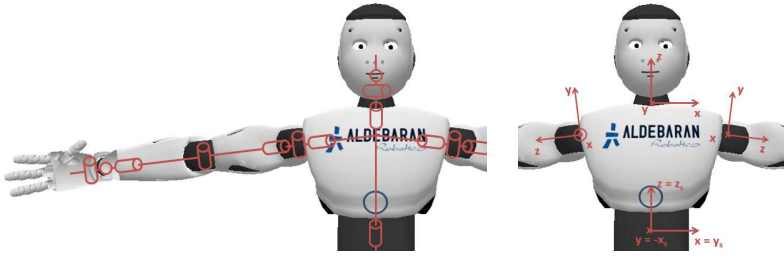
4 Experimental results on physical robots

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 (Hirose and Takenaka, 2001) and Justin (Ott et al., 2006), both having 6 DOF arms, but however with considerably different morphology (Van de Perre et al., 2015, 2016b,a). For this validation on physical robots we worked with the robots Romeo (Aldebaran Robotics, 2014b), Pepper (Aldebaran Robotics, 2014a) and Nao (Gouaillier et al., 2009). All three robots have a different morphology. The specifications for Romeo are grouped in table 3. 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, a DH-frame was assigned to each joint. The frames corresponding to the first joint of each chain, the chain base frames, are visualized in the middle top of table 3, together with the standard reference frame $x_s y_s z_s$ 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 3 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 necessary 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 4 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 5. 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 his mother back. A number of calculated postures for all three robots are listed in figure 8. 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 into a youtube page (Van de Perre, 2016). 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 3), a number of calculated gestures violated these limits. The calculated joint

Table 3. Specifications for the robot Romeo

Joint configuration	DH-base frames	$^{base}R_{stand}$

Body

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

Head

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -51 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Right

$$\begin{pmatrix} 0.91 & 0.42 & 0 & 8 \\ 0.07 & -0.16 & 0.99 & -43 \\ 0.42 & -0.89 & -0.17 & -9.7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Left

$$\begin{pmatrix} 0.91 & -0.42 & 0 & 8 \\ -0.07 & -0.16 & -0.99 & 43 \\ 0.42 & 0.89 & -0.17 & -9.7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Joint	DH-parameters				Joint limits		$\dot{\theta}_{max}$ (*rad/s*)
	α (°)	a (*cm*)	d (*cm*)	θ (°)	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
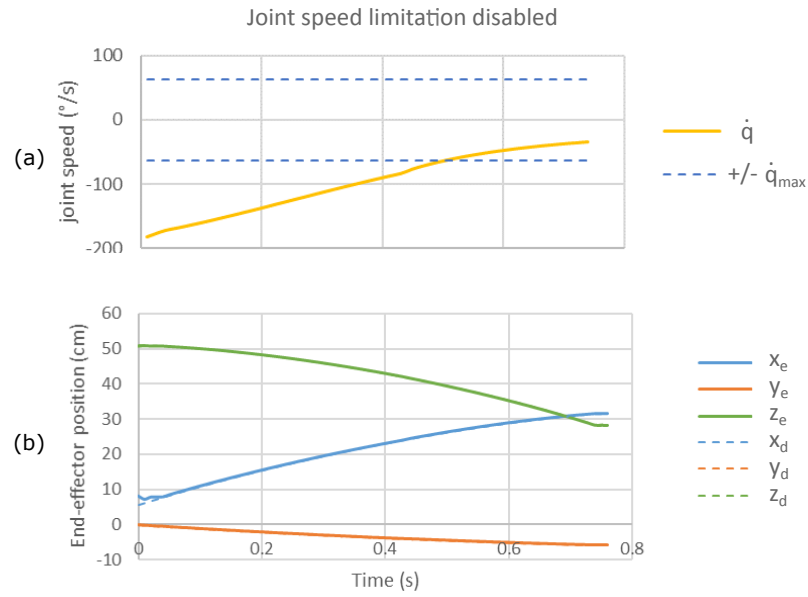
Figure 6. Output for a right-handed reaching gesture calculated for the robot Romeo without using the joint speed limitation algorithm: (a) Calculated speed for the first wrist joint, together with its boundaries, (b) End-effector trajectory of the wrist block. The full line denotes the calculated end-effector position $x_{i,e}$, while the dotted line shows the desired end-effector position $x_{i,d}$.
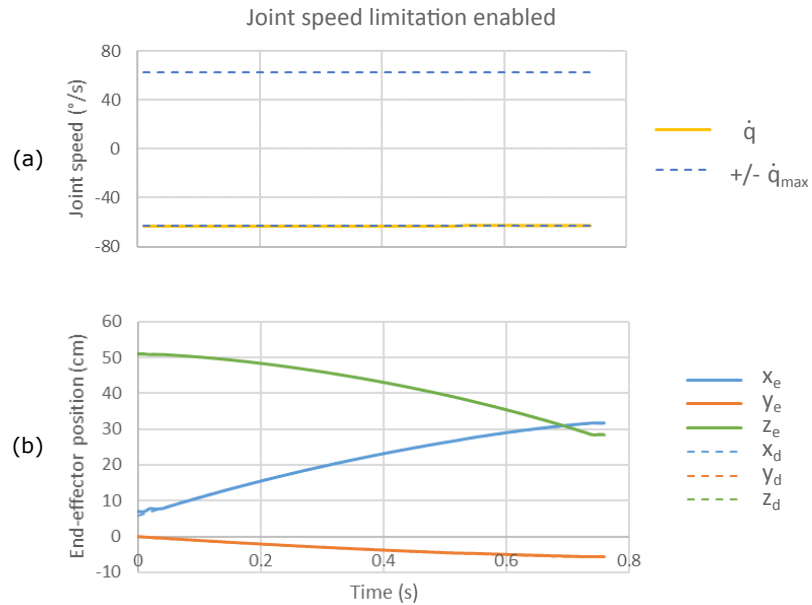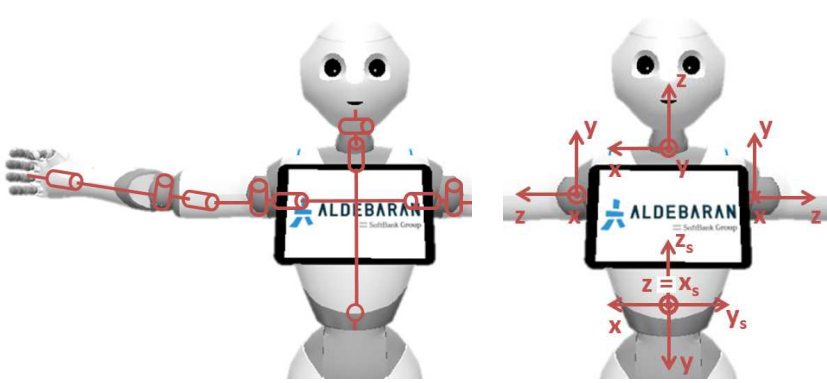
Figure 7. Output for a right-handed reaching gesture calculated for the robot Romeo using the joint speed limitation algorithm: (a) Calculated speed for the first wrist joint, together with its boundaries, (b) End-effector trajectory of the wrist block. The full line denotes the calculated end-effector position $x_{i,e}$, while the dotted line shows the desired end-effector position $x_{i,d}$.

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 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 calculate the joint trajectories for Romeo. Since the physical robot was not any more to

Table 4. Specifications for the robot Pepper
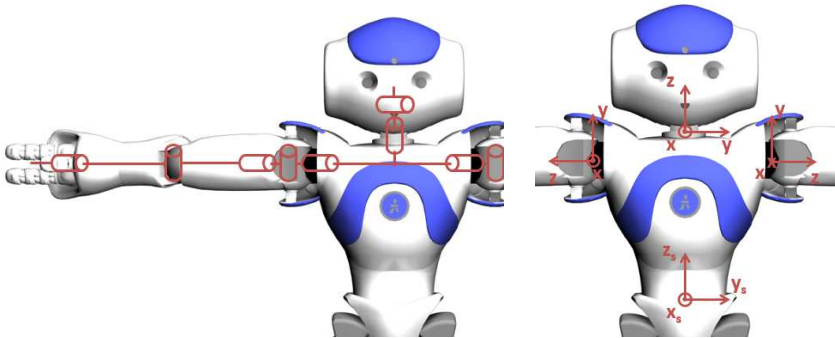
Joint configuration	DH-base frames	$^{base}R_{stand}$	
		Body	$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
		Head	$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -38.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
		Right	$\begin{pmatrix} 1 & 0 & 0 & 7.2 \\ 0 & 0 & 1 & -24 \\ 0 & -1 & 0 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
		Left	$\begin{pmatrix} -1 & 0 & 0 & -7.2 \\ 0 & 0 & 1 & -24.4 \\ 0 & 1 & 0 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Joint	DH-parameters				Joint limits		$\dot{\theta}_{max}$ (rad/s)
	α (°)	a (cm)	d (cm)	θ (°)	min (°)	max (°)	
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

our availability, 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

Table 5. Specifications for the robot NAO

Joint configuration	DH-base frames	$^{base}R_{stand}$	

Head $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -21.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Right $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & -1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Left $\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & 1 & 0 & -9.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Joint	DH-parameters				Joint limits		$\dot{\theta}_{max}$ (rad/s)
	α (°)	a (cm)	d (cm)	θ (°)	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

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 with $160ms$. Figure 9 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.
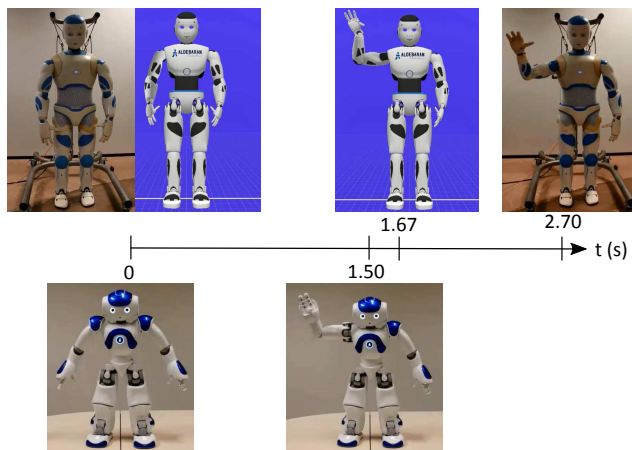
Figure 9. Timing of the first gesture of the test scenario for Romeo, how it was implemented on the real robot without using the joint speed limitation and rescaling the resulting joint trajectories, how it is optimally calculated using the joint speed limitation algorithm, visualized on the virtual model, and the result obtained for the robot NAO, having the desired gesture duration.

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.

5   Conclusions

To ease the sharing of gestures between different robot morphologies, we proposed the use of a generic gesture method. The framework of the method was designed very flexible, using a human base model as reference. By calculating a mapping based on a random configuration chosen by the user, different types of gestures can be generated. The different modalities of the method were validated on the virtual model of several robots in previous publications (Van de Perre et al., 2015, 2016b,a). 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 guaranty 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 (2012). Through an illustrative example, the optimal parameters for this algorithm were 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.

References

Aldebaran Robotics (2014a) Pepper documentation. URL `http://doc.aldebaran.com/2-4/home_pepper.html`. [accessed February-2016].

Aldebaran Robotics (2014b) Romeo documentation. URL `http://doc.aldebaran.com/2-4/home_romeo.html`. [accessed February-2016].

Amaya K, Bruderlin A and Calvert T (1996) Emotion from motion. In: *Graphics interface*, volume 96. Toronto, Canada, pp. 222–229.

Ascher UM and Petzold LR (1998) *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam.

Azad P, Asfour T and Dillmann R (2007) Toward an unified representation for imitation of human motion on humanoids. In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, pp. 2558–2563.

Balit E, Vaufreydaz D and Reignier P (2016) Integrating animation artists into the animation design of social robots. In: *ACM/IEEE Human-Robot Interaction 2016*.

Drexler DA and Harmati I (2012) Joint constrained differential inverse kinematics algorithm for serial
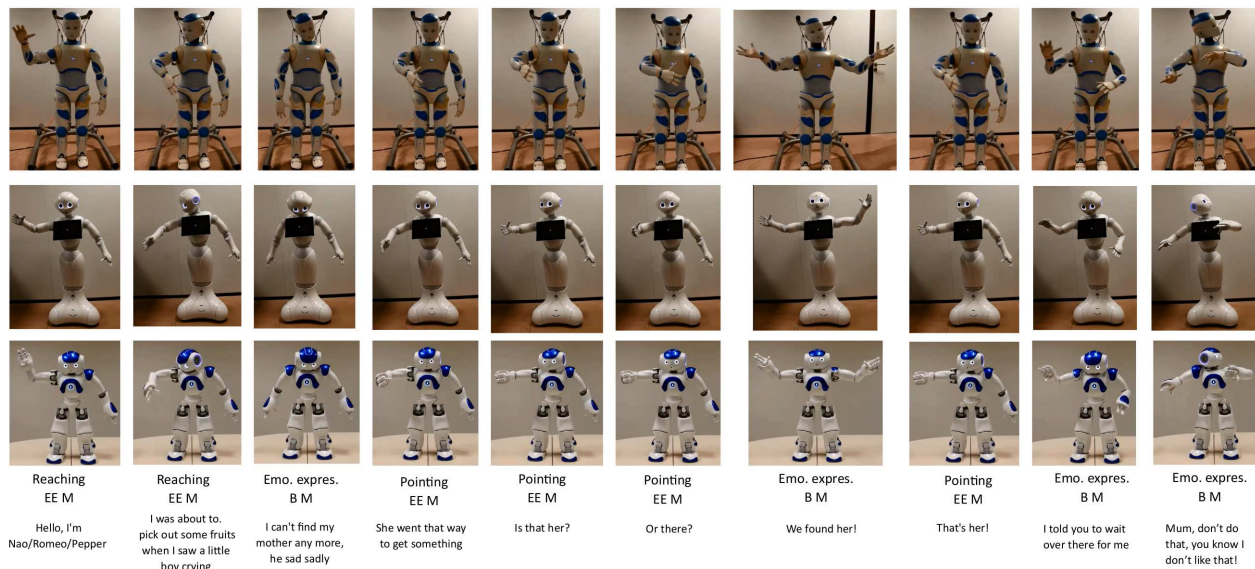
Figure 8. Postures captured from the gestures calculated by the method for the robots Romeo, Pepper and NAO. Below every posture, the type of gesture, the used calculation mode (B M: Block mode or EE M: End-effector mode) and the context in the story are added.

manipulators. *Periodica Polytechnica. Electrical Engineering and Computer Science* 56(4): 95.

Gienger M, Janssen H and Goerick C (2005) Task-oriented whole body motion for humanoid robots. In: *Humanoid Robots, 2005 5th IEEE-RAS International Conference on.* IEEE, pp. 238–244.

Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J and Maisonnier B (2009) Mechatronic design of nao humanoid. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, pp. 769–774.

Hirose R and Takenaka T (2001) Development of the humanoid robot asimo. *Honda R&D Technical Review* 13(1): 1–6.

Kadaba MP, Ramakrishnan H and Wootten M (1990) Measurement of lower extremity kinematics during level walking. *Journal of Orthopaedic Research* 8(3): 383–392.

Klein CA and Blaho BE (1987) Dexterity measures for the design and control of kinematically redundant manipulators. *The International Journal of Robotics Research* 6(2): 72–83.

Koga Y, Kondo K, Kuffner J and Latombe JC (1994) Planning motions with intentions. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* ACM, pp. 395–408.

Le QA, Hanoune S and Pelachaud C (2011) Design and implementation of an expressive gesture model for a humanoid robot. In: *11th IEEE-RAS International*

Conference on Humanoid Robots. IEEE, pp. 134–140.

Liegeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics* 7(12): 868–871.

Lin YH, Liu CY, Lee HW, Huang SL and Li TY (2009) Evaluating emotive character animations created with procedural animation. In: *Intelligent Virtual Agents.* Springer, pp. 308–315.

Mühlig M, Gienger M and Steil JJ (2012) Interactive imitation learning of object movement skills. *Autonomous Robots* 32(2): 97–114.

Ott C, Eiberger O, Friedl W, Bauml B, Hillenbrand U, Borst C, Albu-Schaffer A, Brunner B, Hirschmuller H, Kielhofer S, Konietschke R, Suppa M, Wimbrock T, Zacharias F and Hirzinger G (2006) A humanoid two-arm system for dexterous manipulation. In: *Humanoid Robots, 2006 6th IEEE-RAS International Conference on.* IEEE, pp. 276–283.

Pelachaud C (2009) Studies on gesture expressivity for a virtual agent. *Speech Communication* 51(7): 630–639.

Salem M, Kopp S, Wachsmuth I and Joublin F (2010) 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).*

Sciavicco L (2009) *Robotics: modelling, planning and control.* Springer.

Soechting JF and Flanders M (1989) Errors in pointing are due to approximations in sensorimotor transformations. *Journal of Neurophysiology* 62(2): 595–608.

Stanton C, Bogdanovych A and Ratanasena E (2012) Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In: *Proc. Australasian Conference on Robotics and Automation*.

Van de Perre G (2016) Experimental results : video's of the test scenario for romeo, pepper and nao. URL `https://www.youtube.com/playlist?list=PLjQ6geXigjulOapQdhE67ZuKkJEVOvUmk`.

Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D and Vanderborght B (2016a) Generic method for generating blended gestures and mood expressions for social robots. *Autonomous Robots (in review)* .

Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D and Vanderborght B (2016b) Reaching and pointing gestures calculated by a generic gesture system for social robots. *Robotics and Autonomous Systems (accepted)* .

Van de Perre G, Van Damme M, Lefeber D and Vanderborght B (2015) Development of a generic method to generate upper-body emotional expressions for different social robots. *Advanced Robotics* 29(9): 59–609.

Xu J, Broekens J, Hindriks K and Neerincx MA (2013a) Bodily mood expression: Recognize moods from functional behaviors of humanoid robots. In: *Proceedings of International Conference on Social Robotics (ICSR)*. pp. 511–520.

Xu J, Broekens J, Hindriks K and Neerincx MA (2013b) The relative importance and interrelations between behavior parameters for robots' mood expression. In: *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. IEEE, pp. 558–563.

Yamaguchi A, Yano Y, Doki S and Okuma S (2006) A study of emotional motion description by motion modification and adjectival expressions. In: *IEEE Conference on Cybernetics and Intelligent Systems 2006*. pp. 1–6.

Period 4

A personalized and platform-independent behavior control system for social robots in therapy: development and applications

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

Abstract—**Social robots have been proven beneficial in different types of healthcare interventions. An ongoing trend is to develop (semi-)autonomous socially assistive robotic systems in healthcare context to improve the level of autonomy and reduce human workload. This paper presents a behavior control system for social robots in therapies with a focus on personalization and platform-independence. This system architecture provides the robot an ability to behave as a personable character, which behaviors are adapted to user profiles and responses during the human-robot interaction. Robot behaviors are designed at abstract levels and can be transferred to different social robot platforms. We adopt the component-based software engineering approach to implement our proposed architecture to allow for the replaceability and reusability of the developed components. We introduce three different experimental scenarios to validate the usability of our system. Results show that the system is potentially applicable to different therapies and social robots. With the component-based approach, the system can serve as a basic framework for researchers to customize and expand the system for their targeted healthcare applications.**

Index Terms—**robot-assisted therapy, supervised autonomy, social robots, personalized behaviors, platform-independent.**

I. INTRODUCTION

SOCIAL robots have been developed to provide therapeutic assistance in healthcare for broad populations of users from children to adults e.g. children with autism, diabetes, physical exercises, elderly care [1]–[4]. Robots used in these contexts are required to understand the environment, human intention and performance, and to follow the therapeutic goals to perform meaningful and personalized interaction [5]–[7]. To lessen these challenges, most studies using socially assistive robots in healthcare are restricted to the Wizard of Oz (WoZ) approach [8], [9], in which robots are pre-programmed or remote-controlled by a human operator [3], [10]–[12]. This approach requires a significant amount of manual work to control the robot which deviates the attention of the operator away from the behaviors of the patient and might lead to missing important therapeutical reactions. Therefore, there is an ongoing trend in social robotics toward developing (semi)-autonomous behavior control systems to increase the level

H.-L. Cao, G. Van de Perre, P. Gómez Esteban, A. De Beir, D. Lefeber, and B. Vanderborght are with Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel and Flanders Make, Brussels, Belgium.

R. Simut is with the Department of Clinical and Life Span Psychology, Vrije Universiteit Brussel, Brussels, Belgium.

J. Kennedy, E. Senft, and T. Belpaeme are with Centre for Robotics and Neural Systems, Plymouth University, Plymouth, U.K.

Manuscript received ***; revised ***.

of robot autonomy and lighten direct human control while fulfilling the desired requirements [7], [11], [12].

The role of personalization in a robot behavior control system is considered important to sustain user's motivation and engagement [5], [6]. Personalization should be considered from both therapeutic perspective (e.g. interaction scenario) and robot behavior perspective to reflect the user's needs, requirements and preferences [13]–[15]. Behaviors can be personalized at different levels from physical level to cognitive and social levels taking into account static (e.g. user name, gender, personality) and dynamic (e.g. emotion, current response) parameters [16], [17]. This points out that personalization should be implemented throughout different parts/modules of the behavior control system.

The role of the therapists in robot assisted therapy systems should not be neglected. Results from surveys investigating the general public as well as the therapists recommend that therapist should not be replaced by robots (fully autonomous) for clinical and ethical concern, as well as to compensate for technical limitations [7], [18]–[23]. This indicates that the robot behavior control system should be organized for the ease of shared-control.

Another issue is that the current behavior control systems used are typically developed for particular therapeutic scenarios and robot platforms [6]. The developed systems and importantly the robot behaviors are not easily applicable and transferable to other applications, which is known as the correspondence problem [24], [25]. Some studies propose that the robot behaviors should be programmed at abstract levels and can be translated to low-level commands for different robot morphologies [6], [7], [12], [26]–[29].

Taking into account the above-addressed issues, we present in this paper a robot behavior control system to generate personalized behaviors under therapist supervision and can be applied to different robot platforms. The system architecture enables the robot to behave as a personable character, which behaviors are adapted to user profiles and responses during the human-robot interaction. Personalized behaviors are generated using a generic method allowing the behaviors to be performed in different robot platforms [28], [29]. The system is implemented following the component-based software engineering (CBSE) approach allowing the developed components to be replaceable or reusable [30]–[33]. We introduce three experimental scenarios in autism therapy, elderly care, and physical exercise to validate the usability of our system.

Results show that the system is potentially applicable in different therapeutic scenarios. Moreover, with the component-based approach, the system architecture can serve as a basic framework for researchers to customize and expand the system for their applications.

The paper is organized as follows. Section II overviews the related work on autonomous behavior control systems, personalized behaviors, and generic methods to generate behaviors. Section III presents the approach and design principles. Section IV and V describe the design and implementation of our proposed system. Section VI introduces some case studies to validate the system. Finally, the conclusion is given.

II. RELATED WORK

A. Robot Behavior Control Architectures in Therapies

Robot behavior control architecture is the central part of a socially assistive robotic system allowing the robot control to go beyond the WoZ approach. The architecture enables the robot to select proper behaviors according to external events from the environment and changes of the robot's internal variables. The organization of the architecture is important to the success of a robot-assisted intervention because the success of such intervention hinges on the behavior of the robot [34].

Robot behavior control architectures in therapies have been designed following various approaches to *achieve therapeutic goals* (see [6] for a review). These approaches include event-driven [35], [36], homeostasis [23], [37] [38], finite-state machine [5] [39], neural networks [40] [41], etc. These architectures enable the robots to generate two main types of behaviors: tasked-based behaviors and social behaviors. *Task-based behaviors* are typically generated based on a sequence of robot actions following a certain therapeutic scenario, which can be understandable to the therapists and constrained within specific clinical and ethical boundaries [1], [20]. *Social behaviors* are generated to enhance the robot's social properties e.g. reaction, attention, emotion, and create the illusion of the robot being alive [42], [43]. Social behaviors are either hard-coded in the task-based actions; or generated, combined/arbitrated by separate modules [6]. To better achieve the therapeutic goal, robot behaviors should sustain user's motivation and engagement during all phases of the therapeutic process by providing personalized behaviors (see Sect. II-B) and interesting play scenarios during and between the interventions [5], [6], [21], [44]–[46].

Previous works (e.g. [6], [7], [12], [18]–[22]) also suggest other important requirements that a robot control architecture in healthcare should acquire:

- *Shared-control*: Human therapists should not be replaced by robots but supervise the robot operation due to clinical and ethical concerns as well as compensating for technical limitations [6], [7], [18]–[23].
- *Scenario-independent and platform-independent*: Robot behavior modules developed for one scenario should be reusable for another without significant modifications e.g. reorganizing, re-sequencing [20], [21]. Similarly, robot behaviors should be developed to be expressed in various robot platforms (see Sect.II-C) [6], [7], [12].

- *Providing data for analysis*: Robot behavior control architectures should record data (e.g. therapeutic performance, user's performance history, robot operation) in structured forms, and also document and analyze the data to reduce the workload of the therapists [6], [21].

B. Personalized Robot Behaviors

Personalization in social robotics is defined in different but non-conflicting ways. Dautenhahn *et al.* defined personalization as reflecting the needs and requirements of the social environment where the robot is operating in [14]. Lee *et al.* described personalization as the creation of a personable character to increase friendliness, to fit user preferences, and to adapt over repeated encounters [15]. Baxter *et al.* identified three particular facets of personalisation: adaptation of non-verbal behavior, personable language content, and alignment to task performance [13].

These definitions point out that behaviors can be personalized at different levels from physical level to cognitive and social levels taking into account static (e.g. user name, gender, personality) and dynamic (e.g. emotion, current response) parameters [16], [17]. In socially assistive robotic systems, personalization should be considered from therapeutic aspects (e.g. selecting proper interaction scenarios) to robot behavior aspects in both task-based and social behaviors.

C. Methods for Generating Behaviors for Different Robots

Robot behaviors especially gestures are often preprogrammed or generated by mapping motion capture data to the robot, which are dependent on the robot's configuration and not easily transferable to other robots [29]. Some studies have attempted to create generic methods allowing the developed robot behaviors to be applied to other robots' morphologies. Stanton *et al.* used neural networks, which requires training, to flexibly generate gestures for different robots [47]. Others applied techniques used in virtual characters to humanoid robots. Salem *et al.* abstractly described gestures of MAX virtual agent in three features: the location of the wrist, the shape of the hand and the orientation of the wrist; and translated these features into ASIMO's motor control commands [27]. Similarly, Le *et al.* used three features: the hand shape, wrist position and palm orientation to generate gestures for NAO robot, which relied on a predetermined table listing all possible wrist positions and the corresponding joint values [26]. Van de Perre *et al.* used a Body Action Coding System (BACS) to describe abstract human gestures and generate gestures for different robot platforms e.g. NAO, ASIMO, Justin [29]. In this method, a set of joints in a certain robot configuration is considered as a subset of the human body configuration, where virtual joints were added if necessary [29]. Ultimately, these methods reduce hard-coding effort and time to develop and reuse developed robot behaviors.

III. APPROACH AND DESIGN PRINCIPLES

A. Approach

Our approach to design a system architecture was motivated by two fundamental concepts. First, the system should provide

the robot an ability to behave as a personable character. Second, the developed system can be applied to different therapies and robot platforms. These two concepts are consistent with the requirements derived from related works (Sect. II-A). These requirements function as *guidelines* for us to form our design principles for the system architecture development. We also use these guidelines as criteria to evaluate the system.

B. Design Principles

Following the guidelines, we stated three design principles as the basis for the architecture design process. Fig. 1 overviews how the design principles fulfill different elements of the guidelines.

1) Multi-layer behavior: The behavior generation is organized in layers to produce different types of behaviors. We followed the three-layer behavior concept from behavioral psychology, which is widely used in behavioral sciences and robotics [48]–[51]. The behavior system includes the reactive layer, the deliberative layer, and the reflective layer. The reactive and deliberative layers are respectively responsible for generating social behaviors and tasked-based behaviors to follow the therapeutic goals. The reflective layer evaluates the correctness of behaviors generated by the lower layers, and correct them if necessary following the clinical and ethical standards. The reflective layer can be accessed by the therapists allowing shared control.

2) Personalization: The generated behaviors are personalized enabling the robot to behave as a personable character. The architecture should provide the robot an affect system to manage the robot's variables including personality (which can be matched or complementary to the user's personality depending on scenarios [4], [52]–[55]), and affect (mood and emotion). These variables are expressed through the robot behaviors generated by the behavior decision-making.

3) Modularity: The architecture is organized into modules/components for the ease of customization and logging structured data for different parties (e.g. engineers, therapists, patients). The behavior generation is separated with the scenario management to make the system scenario-independent. The behavior generation produces abstract behaviors and is independent to the behavior expression. This allows the generated behaviors to be expressed in different robot platforms. The modularity principle is also applied to the implementation process by using the component-based software engineering approach [30]–[33].

IV. System Architecture

Our proposed system architecture aims to provide a personalized and platform-independent system for robots in different therapies. The system architecture is depicted in Fig. 2. The perceptual system interprets raw data from sensors into abstract interaction events. The affect system allows the robot to behave as a personable character, which is expressed through behaviors generated by a three-layer behavior generation mechanism. These behaviors include tasked-based and social behaviors depending on the therapeutic scenario and user's profile (behavior generation). The behaviors are
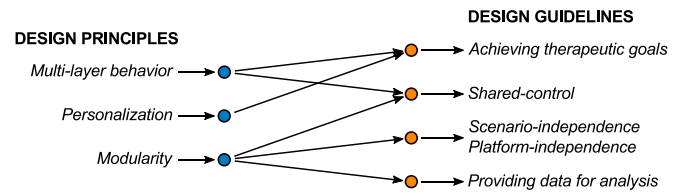
Fig. 1. Relationships between design principles and guidelines derived from related works.
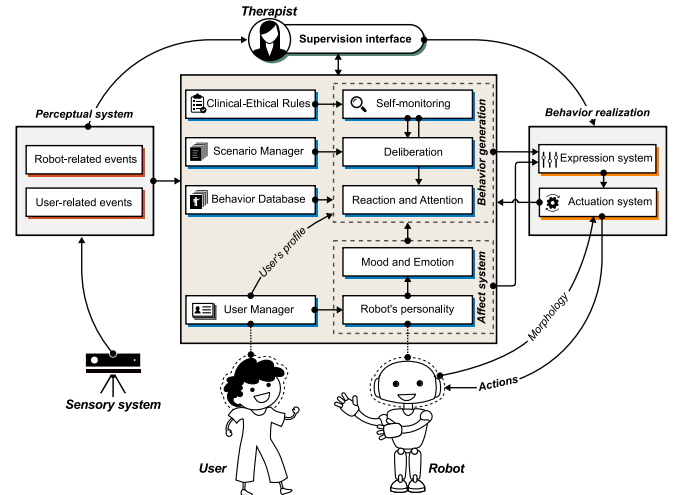
Fig. 2. System architecture. Personalized behaviors are generated taking into account interaction events, user's profile, therapeutic scenario, and clinal-ethical rules. The abstract behaviors are automatically mapped to different robot platforms. A human therapist supervises the system operation.

accordingly verified automatically by clinical-ethical rules and the therapist. Robot behaviors are designed at abstract levels and translated into robot-special action commands (behavior realization). The following describes general functions of the system elements and models used in each system.

A. Perceptual System

This system receives raw sensory data from the robot's built-in sensors and external sensors (e.g. cameras) and interprets into abstract interaction events. These events are passed to the behavior generation system and the robot's affect system.

B. User Manager

This system manages user profiles e.g. name, age, gender, personality, preference, performance history. This data is used to produce personalized behaviors and can be extracted to structured formats for analysis.

C. Affect System

This system enables the robot to behave as a personal character by having its own personality and affect (mood and emotion). The robot's personality can be adapted to the user's personality depending on the scenario [4], [52]–[55]. The robot's mood and emotion are changed according to the interaction events. Under the same event, the numerical changes are adapted to the personality. The robot's affect is expressed through the robot's behaviors.
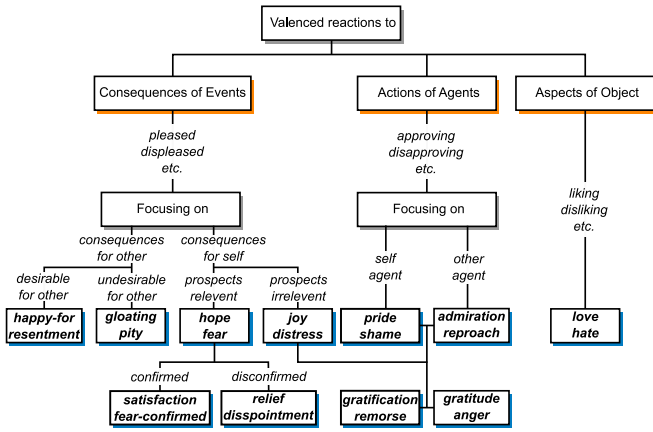
Fig. 3. The OCC model used in the proposed architecture allowing the robot to adapt its emotional state according to events [59].

Fig. 4. Positions of OCC basic emotion types in a three-dimensional emotion space [61], [62].

We applied widely used models in psychology and robotics for the affect system. Personality is modeled by five factors: Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism [56], [57]. And affect is modeled by three factors: Valence, Arousal, and Dominance [58].

$$\mathbf{P}^T = \begin{bmatrix} O & C & E & A & N \end{bmatrix} \quad \mathbf{E}^T = \begin{bmatrix} V & A & D \end{bmatrix}$$

These affective values are changed according to the type and intensity of the occurred event, and the robot's personality [60]. These changes are managed by the Orthony Claire Collins model (OCC model) [59] as illustrated in Fig. 3. The model includes 22 basic emotion types (Fig. 4). The mapping of emotion types into the emotion space is based on the work of Gebhard [61], which was adapted from Mehrabian [62]. When there is no event, the robot affect decays to its default value $\mathbf{E_0}$ [61].

$$\mathbf{E_0}(\mathbf{P}) = \begin{bmatrix} V_0 \\ A_0 \\ D_0 \end{bmatrix} = \begin{bmatrix} 0.21E + 0.59A + 0.19N \\ 0.15O + 0.30A - 0.57N \\ 0.25O + 0.17C + 0.60E - 0.32N \end{bmatrix}$$

When an event $\mathbf{e_t}$ occurs, the OCC model evaluates the event by seven properties i.e. desirability, praiseworthy, appealing, likelihood, liking, realization, and agency [61].

Based on this evaluation, a desired emotion $\mathbf{E_t^d}$ is selected. The robot affect moves from the previous affect position $\mathbf{E_{t-1}}$ to the new affect position $\mathbf{E_t}$ taking into account the event intensity calculated from the individual intensities of seven properties $i_e(\mathbf{e_t})$, and the influence of personality on the desired emotion $i_P(\mathbf{P}, \mathbf{E_t^d})$.

$$\mathbf{E_t} = \mathbf{E_{t-1}} + i_e(\mathbf{e_t}).i_P(\mathbf{P}, \mathbf{E_t^d}).(\mathbf{E_t^d} - \mathbf{E_{t-1}})$$

In this work, the event intensity is the greatest value among those of the individual properties [61]. The influences of personality factors on the basic emotion types are summarized in Table I, which is inspired by the review of Doce *et al.* [63].

D. Behavior Generation

This system generates robot abstract behaviors based on user's profiles, interaction events and databases. Following
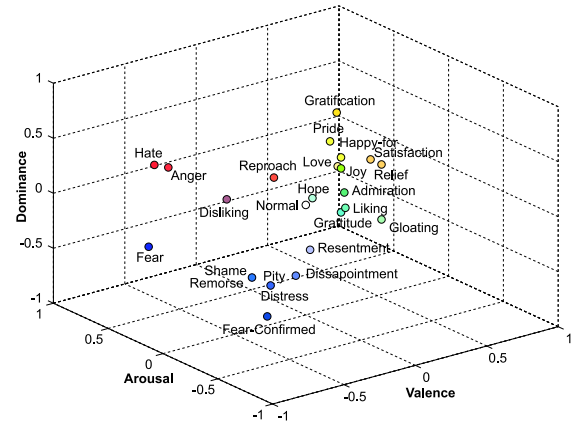
TABLE I
INFLUENCE OF PERSONALITY ON EMOTION TYPES

Factors	Positive influence	Negative influence
Openness	–	Pride, Shame, Admiration, Reproach
Conscientiousness	Pride	Gratification
Extroversion	–	All emotions
Agreeableness	Love, Happy-For, Pity, Admiration, Gratitude	Hate, Resentment, Gloating, Reproach, Anger
Neuroticism	Distress, Joy, Disappointment, Relief, Gratification, Fear	Satisfaction, Fears-Confirmed, Remorse, Hope

the three-layer behavior organization approach consisting of the reactive layer, the deliberative layer, and the reflective layer [48]–[51], the behavior generation system is composed of three subsystems, namely the reaction and attention, the deliberation, and the self-monitoring subsystems.

The Deliberation subsystem generates task-based behaviors following the current therapeutic scenario, managed by the Scenario Manager. Tasked-based behaviors are personalized by using the user's name and giving feedback based on the user's performance during the interaction.

The Reaction and Attention subsystem generates social behaviors, which creates the illusion of the robot being alive and acts as a catalyst for acceptance [4], [42], [43]. The reaction part includes: falling reaction to avoid physical damage; social reaction to touches and user's emotion; like-life reaction e.g. eye blinking, micro motions. The attention part includes gaze adaptation to the user and sound direction [64].

The Self-monitoring subsystem partially verifies the selected behaviors of the Deliberation and the Reaction and Attention subsystems by referring to Clinical-Ethical Rules. These rules include prohibitions and obligations set by therapists [65]. However, most of the robot behaviors are verified by a human therapist before execution to strictly guarantee the ethical and clinical standards (see IV-E).

E. Therapist Supervision Interface

This interface provides the therapist an intuitive tool to supervise the robot operation, which is important to increase

their involvement in the robot's control [66]–[70]. Through this interface, the therapist can verify a generated behavior before execution or select a more proper one if necessary. The technical limitation of the system can be compensated by having the therapist to correct the operation of the perceptual system and the affect system. The interface also provides an ability to switch on/off individual subsystems and manage the databases.

F. Behavior Realization

This system is responsible for expressing personalized behaviors on the robot platform. It converts abstract generated behaviors and influences of personality and affect into robot-specific behaviors. This process is based on the two following bases.

1) Mapping behavior from human base model to robot configuration: Robot behaviors are designed on a human base model using a Body Action Coding System (BACS). The behavior is converted into robot-specific behavior by adopting a method developed by Van de Perre *et al.* [28], in which the robot configuration is considered as a subset of a human body configuration, and virtual joints are added if necessary. The joint angles are then calculated using the Runge-Kutte algorithm [71] by the following equation [72]:

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

where x_d is the desired end effector pose, $J_A^\dagger(q)$ is the Moorse-Penrose pseudo inverse of the analytical jacobian $J_A(q)$, x_e the current end effector pose, and K a positive definite gain matrix. Fig. 5 shows an example of happy behavior expressed on different robot platforms.
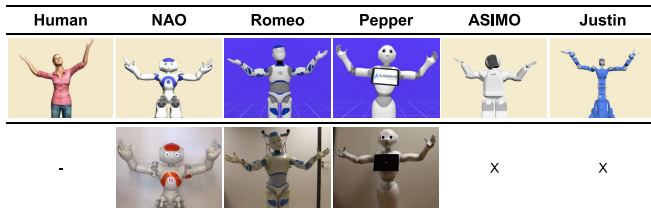
Fig. 5. Expression of a happy behavior on different robots platforms: virtual models (top) and physical robots (bottom).

2) Influences of personality and affect on behavior expression: Previous works show that valence, arousal, and extroversion have significantly higher influences on behavior expression than other factors [73]–[76]. These factors influence pose's amplitude, palm direction, finger rigidness, head's position, motion speed, holding time, and repetition as suggested by different studies e.g. [73], [74], [77]–[81]. These influences are considered when designing behaviors on the human base model.

The behavior realization process used in our work is illustrated in Fig. 6. In this work, we simplify the influences of valence, arousal, and extroversion into a single affect level formed by the weighted sum of individual factors. Unlike previous works in which affective behaviors are generated based on expression at the neutral affect, our method is based
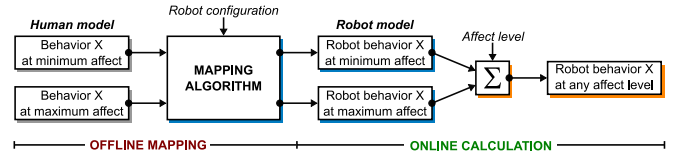
Fig. 6. The process of converting behavior from human base model to robot-specific behavior with affect influence.
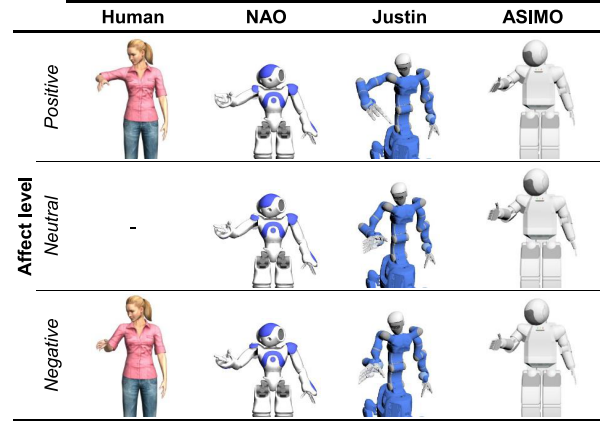
Fig. 7. The end pose of pointing behavior expressed on NAO, Justin, and ASIMO with different affect levels: positive, neutral, negative.

on expression at two extreme affect levels i.e. minimum and maximum.

The process starts from designing behaviors on a human-based model. Each behavior is coded in two extreme affective levels. This behavior is converted into robot behavior. Behavior expression at a certain affect level is calculated by linear interpolation. The joint angles of each frame of a robot behavior timeline at a certain affect level is calculated by the following equation:

$$q_i = q_{i,min} + \underbrace{(\alpha_V V + \alpha_A A + \alpha_E E)}_{\text{Affect level}}(q_{i,max} - q_{i,min})$$

This process does not require inverse kinematics calculation during the robot operation. Hence, computation workload is reduced. An example of this process is illustrated in Fig. 7 showing the end pose of pointing behavior at different affect levels expressed on NAO, Justin, and ASIMO. After this process, the robot-specific robot behavior with calculated joint angles is used to control the physical robot platform.

V. SYSTEM IMPLEMENTATION

The system architecture is implemented following the component-based approach [30]–[33] with the aim to increase the reusability and replaceability of developed components (i.e. systems, subsystems). The system is developed using YARP robot developing framework which highly supports component development and system integration [82]–[84].

The implemented system consists of a number of components representing systems and subsystems described in the system architecture (Sect. IV). The communication among components is handled by YARP using ports and connectors. The system is connected to the sensors and robot platform by
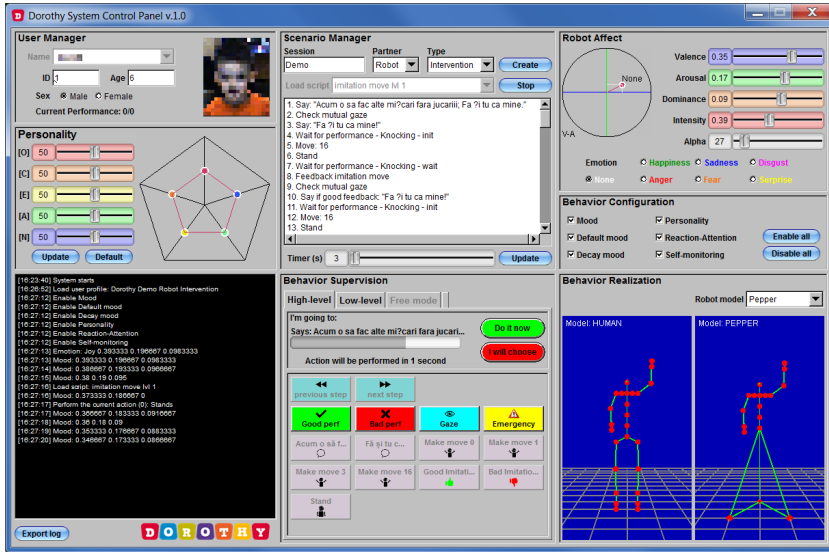
Fig. 8. The system graphical user interface is used to visualize and supervise the system operation. It includes eight group-boxes: User Manager, Scenario Manager, Personality, Robot Affect, Behavior Configuration, Behavior Supervision, Behavior Realization and Log Window.

different means of communication depending on the hardware. The system operation is visualized and supervised by a therapist via a graphical-user interface (GUI). The GUI is divided into eight group-boxes as shown in Fig. 8.

User Manager & Scenario Manager: These group-boxes are used to manage users' profiles, therapeutic scenarios, and to select the interacting user and scenario.

Personality & Affect: This group-box is used to visualize the robot's personality (which is adapted to user's personality), mood, and emotion. These variables can be manually adjusted by the therapist.

Behavior Supervision: This group-box is the main element for supervision purpose. It allows the therapist to approve or correct autonomously decided behaviors from the behavior generation process (i.e. Approve, Correct, Timeout). Besides that, the therapist can also control the scenario's flow.

Behavior Configuration This group-box is used to switch on/off individual layers of behavior generation and personalization depending on the therapeutic requirements.

Behavior Realization This group-box is used to visualize the personalized behavior on the selected robot platform.

Log window This control box records the system operation and can export data to a structured format for analysis.

VI. SYSTEM VALIDATION

In this section, we introduce three experimental scenarios for system validation purpose: autism therapy, elderly care, and physical exercise. In these scenarios, we demonstrate different features of the system as summarized in Table II. The robot platforms used in these scenarios are NAO and Pepper, which have been proven beneficial in different types of therapies due to their appearances, compactness, and stability e.g. [85]–[87]. At the end of this section, we present a usability evaluation by investigating the therapists' opinions toward the system.

A. Autism therapy: Joint attention

The first scenario follows a developmental approach of joint attention used in autism therapy intervention studies e.g.

TABLE II
VALIDATED FEATURES IN THREE SCENARIOS

Scenario	Validated features
Autism therapy	The behavior generation system performance compared with the Wizard-of-Oz
Elderly care	The affect system and platform-independent behaviors
Physical exercise	Personalization and affective behaviors
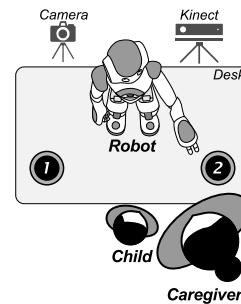
Fig. 9. The experimental setup of the joint attention intervention in autism therapy. A child follows NAO's gaze or pointing to two objects on a desk.
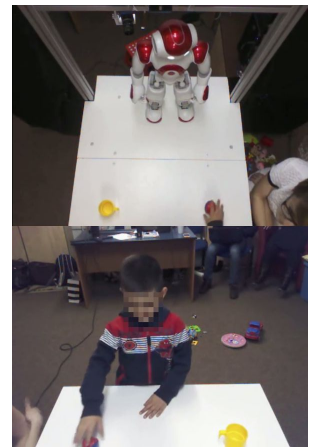
[88], [89]. The idea of this intervention is to increase step-by-step the complexity of the joint attention task from easy (pointing and gaze) to more difficult (gaze only) ones. In this scenario, we compare the performance of the behavior generation system with the WoZ. The affect system was not used to keep the children from being overwhelmed by many social situations.

1) Experimental setup: The setup includes a NAO robot standing on a desk in front of a child who is sitting next to a caregiver as seen in Figure 9. First, the robot introduces the activity to the child. During the interaction, the robot switches its attention to two objects on the desk with different levels of prompts. The robot performs feedback actions according to the child's performance. Finally, the robot says goodbye.

Wizard of Oz

Time (s)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Robot action					Say: Introduction #1							Say: Introduction #2									Look-Point: Left						Good feeback														
Therapist						M						M									M Check gaze						M Check performance														

Time(s)	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
Robot action	Look-Point: Right				Good feeback											Look-Point: Left					Good feeback													Look: Left							
Therapist	M Check gaze				M Check performance											M Check gaze					M Check performance													M Check gaze							

Our proposed system

Time(s)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Robot action		Stand				Say: Introduction #1									Say: Introduction #2																Say: Focus				Look-Point: Left					
Therapist	A					A									A																A										
Suggested action	Stand				Say: Introduction #1							Say: Introduction #2																	G		Say: Focus				Look-Point: Left					
Detected event																														G											

Time(s)	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81
Robot action					Stand			Good feeback					Say: Focus														Look-Point: Right							Stand		Good feeback				
Therapist					A			A						G						P				R			A						A								
Suggested action	P			Stand			Good feedback			G		Say: Focus					Look and Point: Left					Look-Point: Right							P		Stand			Good feedback						
Detected event				P																													P								

Legend: Robot action / Duration — Suggested action — A Approve — G Gaze correction — G Waiting for gaze — P Waiting for performance — M Mouse click — P Pause — R Resume — G Gaze detected — P Performance detected

Fig. 10. Subsets of two interactions using our proposed system and the Wizard of Oz approach. The autonomous behaviors of the robot are comparable with those manually selected by the human therapist. Manual work is therefore reduced.

2) The behavior generation system performance in comparison with the WoZ method: One of the objectives of the system is to reduce manual work for the therapist. In this scenario, we compare the robot autonomous robot behaviors and the behaviors selected by a human therapist in the WoZ in Fig. 10. In the WoZ, the therapist has to constantly observe the interaction events and manually selects proper behaviors by clicking the corresponding buttons. By contrast, our proposed system generates the behaviors autonomously and suggests them to the therapist who has the highest priority of the robot control. In two cases, the executed behaviors are comparable. However, using our proposed system reduces manual work of the therapist during the interaction e.g. looking for proper buttons, detecting user's gaze and performance, keeping track of the interaction flow. The therapist can also access the user's profile including performance history to decide more suitable interacting scenarios for a specific child in the next therapeutic sessions.

B. Elderly care: music therapy

The second scenario is music therapy for individuals with dementia and cognitive impairment. This activity aims to maintain or improve the user's cognitive attention and enhance the quality of life [90], [91]. This scenario demonstrates the affect system performance and the applicability of our proposed system in another robot platform.

1) Experimental setup: The setup includes a Pepper robot standing in front of a senior user as seen in Fig. 11. First, the robot sends a greeting to the user by referring to his/her name. During the interaction, the robot plays three songs randomly. After a song has been played, the user has to answer the song title using the robot's touchscreen. The robot performs feedback actions according to the user's answers. Finally, the robot says goodbye.

Fig. 11. The experimental setup in the elderly care scenario. A senior user plays a music game with Pepper robot.

2) Platform-independent behavior: Fig. 12 demonstrates a frame-by-frame comparison of a happy behavior expressed on Pepper and NAO. This behavior was used when the user answers the song title correctly. The behavior was coded using a human-based model and the method presented in Subsection IV-F. Although the robots are different in morphologies, the generated behaviors still convey a happy message to the user.
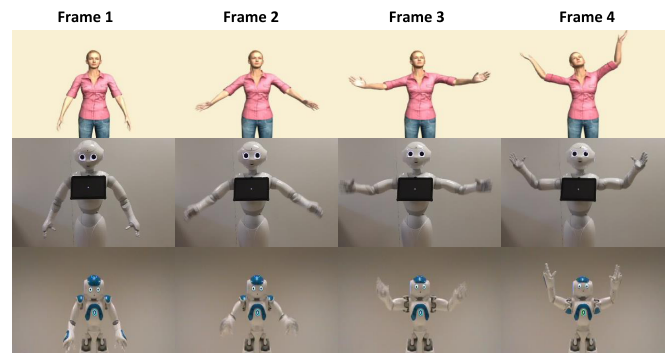
Fig. 12. A happy behavior from a human-based model is expressed on different robot platforms: Pepper and NAO.

3) Affect system demonstration: The affect system performance can be seen in Fig. 13 in which the affect values evolve differently in two cases of personalities (i.e. extrovert and introvert) under similar stimuli. The affect values in two cases started from the same affect point of joy at the beginning of the interaction. When there was no event, the affect values decayed to their corresponding default values i.e. extrovert case $(0.21, 0.0, 0.6)$ and introvert case $(-0.21, 0.0, -0.6)$ (see Subsection IV-C). When good events (e.g. face detected, correct answer) occurred, the affect values moved to the joy direction. By contrast, they moved to the distress direction if bad events occurred (e.g. incorrect answer). The magnitudes of these moves were influenced by the personalities. Specifically, the introvert character felt emotion more intensely than the extrovert one. The magnitudes of moves in the introvert case were greater than those of the extrovert cases. As a result, the robot behaviors were expressed differently.
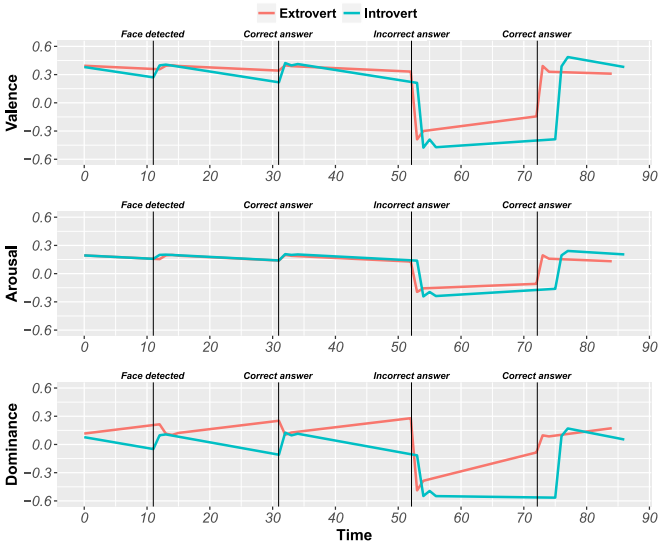
Fig. 13. Evolutions of affect values in two cases of personality: extrovert and introvert, under similar stimuli.

C. Physical exercise: Shoulder restoration routine

The third scenario is a shoulder restoration routine physical exercise in which a user performs shoulder stretches and shoulder strengthening exercises. The activity aims to improve shoulder mobility and posture, and pain relief [92]. In this activity, we demonstrate the ability to express personalized affective behaviors.

1) Experimental setup: The setup includes a NAO robot standing on a desk in front of an interacting user as seen in Fig. 14. Light dumbbells are used at some steps of the activity. First, the robot says hello to the user by referring to his/her name, and describes the purpose of the exercise. After that, the robot guides the user through different steps of the exercise. The user receives feedback depending on his/her performance, which also influences the robot's affect system. Finally, the robot says goodbye.
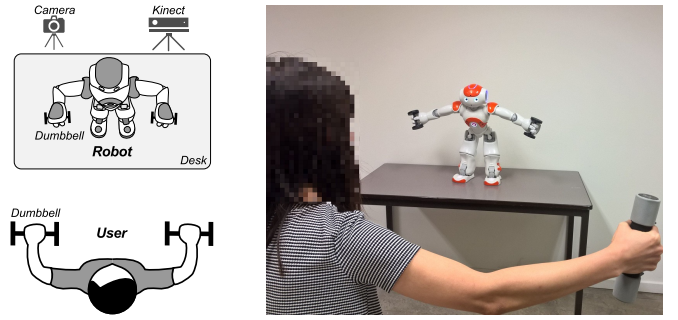
Fig. 14. The experimental setup for the shoulder restoration routine scenario. A user follows NAO's instruction through different exercises.

2) Personalization and affective behaviors validation: We recorded videos from the interactions to perform an online survey investigating the perception of personalization, personality, and affect level. For the personalization perception, we recorded two cases. In the first case, the robot was personalized verbally and nonverbally in which the user manager, reaction and attention subsystem, and the affect system were turned on. The robot, therefore, referred to the user's name and performed affective behaviors. In the second case, the robot was not personalized. We also recorded separate videos to investigate the perception of personality (extrovert and introvert) and affect level (negative, neutral, positive). Manipulation of eye colors was not used to avoid any confounds. Screenshots of videos showing affective behaviors (waving and pointing) can be seen in Fig. 15. All videos are available online[1].
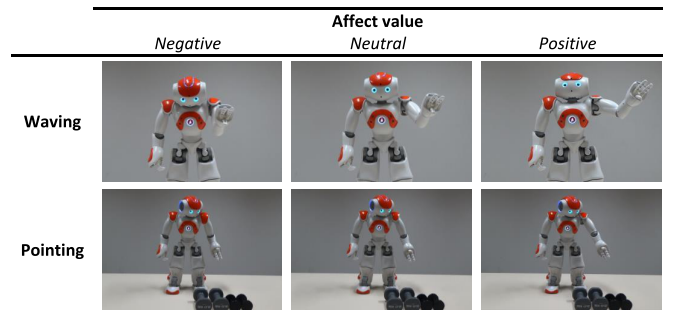
Fig. 15. Screenshots of videos expressing affective behaviors: Waving and Pointing at three affect levels (negative, neutral, positive).

Fourty people (22 men, 18 women) from different backgrounds participated in the survey on Qualtrics. They were recruited through an announcement over the university networks. The age range was from 18 to 37 with the mean age of 26.5 (SD=4.07). The videos were presented in a random sequence to avoid order bias. After watching each video, participants were asked to fill in a questionnaire describing the robot's manner. The questionnaire to measure personalization perception was composed of the personalization-related items of the Robot Immediacy Questionnaire (RIQ) [93] and Robot Nonverbal Immediacy Questionnaire (RNIQ) [94]. Extroversion-related items of the Big Five Inventory test were used to measure personality perception [95]. And a short

[1]hoanglongcao.github.io/research/dorothy#Videos

version of the PANAS test [96] was used to evaluate the perception of the affect level [97]. The questionnaires use five-point Likert scale and are available online[2].

The survey results are summarized in Fig. 16. In the personalization test, a paired two-tailed t-test reveals a significant difference between the average personalization score of the personalized condition (M=3.45, 95%CI[3.37,3.53]) and the non-personalized condition (M=3.13, 95%CI[3.05,3.21]); t(39)=5.77, p<.001. In the personality expression test, a paired t-test shows that the average extroversion score of the extrovert condition (M=2.85, 95% CI[2.75,2.95]) is significantly higher than the introvert condition (M=2.66, 95%CI[2.55,2.76]); t(39)=2.61, p<.05.

A repeated measures analysis using multilevel modelling was conducted to investigate the recognition of affect level. Results show that the total affect level has an influence in the robot behaviors in both waving and pointing. In the waving behavior, there is a significant impact of affective behavior on the affect level score ($X^2(2)$=71.32, p<.001). Posthoc comparisons (Tukey HSD) show that the average score of the positive affect behavior (M=3.75, 95%CI[3.57,3.92]) is significantly higher than the neutral affect behavior (M=3.08, 95%CI [2.94,3.21], p<.001) and the negative affect behavior (M=2.53, 95%CI[2.34,2.71], p<.001). The average score of the neutral affect behavior is significantly higher than the negative affect behavior with p<.001.

In the pointing behavior, there is also a significant impact of the pointing behavior on the affect level score ($X^2(2)$=23.19, p<.001). Posthoc comparisons (Tukey HSD) show that the average score of the positive affect behavior (M=3.22, 95%CI[3.07,3.36]) is significantly higher than the neutral affect behavior (M=2.90, 95%CI[2.79,3.00], p<.001) and the negative affect behavior (M=2.80, 95%CI[2.69,2.91], p<.001). The average score of the neutral affect behavior is not significantly different from the negative affect behavior with p value of 0.4616. In this pointing behavior, it is more difficult to recognize affect level since the arm and the head (gaze) directions were constrained.

D. Usability evaluation

We conducted a survey on Qualtrics to measure the usability of the system, especially the GUI. We recruited four therapists who have been using our system from three to nine months (M=6). The age range was from 25 to 31 (M=26.75). The therapists were asked to answer the USE questionnaire consists of 30 five-point Likert scale items divided into four dimensions: Usefulness, Satisfaction, Ease of use, and Ease of learning[3] [98]. Screenshots of the GUI were used in the questionnaire items to elicit the therapists' memories of their experiences. We also asked open-ended questions to better understand the therapists' opinions. Results showed that the system is useful (M=4.06, SD=0.06), satisfying (M=3.60, SD=0.68), easy to use (M=4.23, SD=0.08), and easy to learn (M=3.75, SD=1.09). Regarding the therapists' opinions, it is worth noting that although the GUI is composed of many

[2]hoanglongcao.github.io/research/dorothy#Questionnaires
[3]hoanglongcao.github.io/research/dorothy#GUIEvaluation
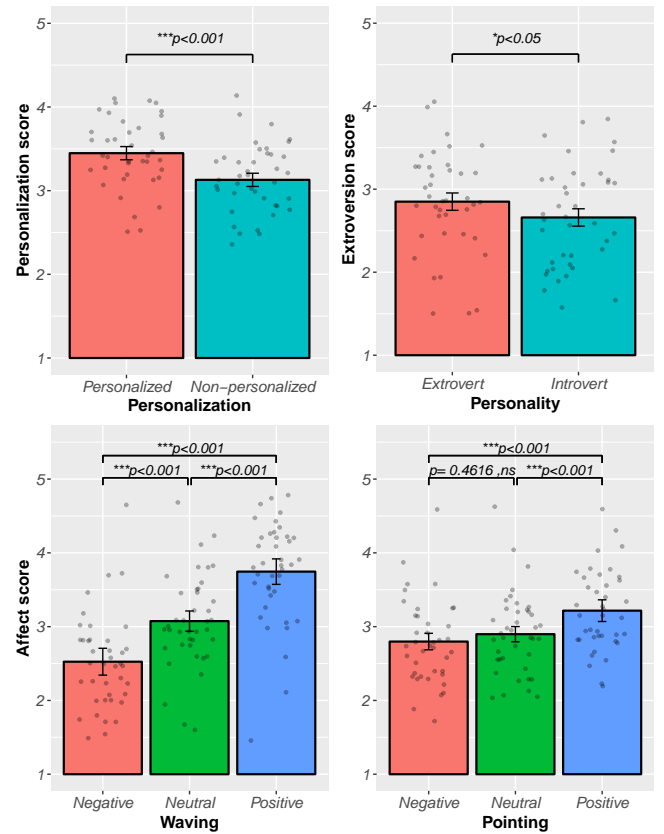
Fig. 16. The survey results. Personalization and affective behaviors were recognized by the participants. Significance is indicated by *p<0.05, **p<0.01, and ***p<0.001. Error bars show the 95% Confidence Interval.

group-boxes, the therapists mainly focused on the Scenario Manager and Behavior Supervision during the intervention. However, they suggested having customized interfaces for different applications.

VII. CONCLUSION

In this paper, we present the development and applications of a platform-independent and personalized behavior control system for social robots in therapies. The system was designed following guidelines drawn from the previous works i.e. multi-layer behavior organization, personalization, and modularity. The implementation process follows a component-based software engineering approach using YARP robot developing framework. This allows the developed components to be reusable and replaceable. Robot behaviors were coded for a human-based model and can be transferred to different robot platforms without reprogramming. A human therapist supervises the system operation through a GUI.

The system validation showed that our proposed system is potentially applicable to therapies using different robot platforms. In these scenarios, the system was able to perform tasked-based and social behaviors, which is similar to other behavior control systems in healthcare, e.g., [5], [23], [34], [86], [99]. Personalization, personality, and affective behaviors were recognized as expected. Although previous works also

implemented these features (e.g., [13], [46], [73], [74], [100]), our system is platform-independent and able to generate the robot behaviors for different robot platforms. To our knowledge, there is a very limited number of studies considering platform-independence, especially in healthcare domain [6], [7], [12]. The usability evaluation investigating the therapists' opinions showed that the system is useful, satisfying, easy to use, and easy to learn.

With the component-based and platform-independent approaches, our system architecture is expected to serve as a basic framework for researchers to customize and expand the system for their targeted healthcare applications. We are currently working on larger scale experiments using the system e.g. investigating the children's performance between child-therapist interaction and child-robot interaction, the effect of personalization on the user's performance, comparing the effect of user-robot personality matching in different tasks.

ACKNOWLEDGMENT

REFERENCES

[1] A. Tapus, M. J. Mataric, and B. Scasselati, "Socially assistive robotics [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 35–42, 2007.

[2] C. Breazeal, "Social robots for health applications," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 5368–5371.

[3] M. J. Matarić and B. Scassellati, "Socially assistive robotics," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1973–1994.

[4] T. Belpaeme, P. E. Baxter, R. Read, R. Wood, H. Cuayáhuitl, B. Kiefer, S. Racioppa, I. Kruijff-Korbayová, G. Athanasopoulos, V. Enescu, *et al.*, "Multimodal child-robot interaction: Building social bonds," *Journal of Human-Robot Interaction*, vol. 1, no. 2, pp. 33–53, 2012.

[5] J. Fasola and M. Mataric, "A socially assistive robot exercise coach for the elderly," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 3–32, 2013.

[6] H.-L. Cao, P. G. Esteban, A. De Beir, R. Simut, G. Van de Perre, D. Lefeber, and B. Vanderborght, "A survey on behavior control architectures for social robots in healthcare interventions," *International Journal of Humanoid Robotics*, pp. 1–21, 2017.

[7] P. G. Esteban, P. Baxter, T. Belpaeme, E. Billing, H. Cai, H.-L. Cao, M. Coeckelbergh, C. Costescu, D. David, A. 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*, pp. 1–22, 2017.

[8] T. K. Landauer, "Psychology as a mother of invention," *ACM SIGCHI Bulletin*, vol. 18, no. 4, pp. 333–335, 1987.

[9] J. Wilson and D. Rosenberg, "Rapid prototyping for user interface design," 1988.

[10] L. D. Riek, "Robotics technology in mental health care," *Artificial Intelligence in Behavioral and Mental Health Care*, p. 185, 2015.

[11] S. M. Rabbitt, A. E. Kazdin, and B. Scassellati, "Integrating socially assistive robotics into mental healthcare interventions: Applications and recommendations for expanded use," *Clinical psychology review*, vol. 35, pp. 35–46, 2015.

[12] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, and B. Vanderborght, "Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook," *Paladyn, Journal of Behavioral Robotics*, vol. 3, no. 4, pp. 209–217, 2012.

[13] P. Baxter, E. Ashurst, R. Read, J. Kennedy, and T. Belpaeme, "Robot education peers in a situated primary school study: Personalisation promotes child learning," *PLoS One*, vol. 12, no. 5, p. e0178126, 2017.

[14] K. Dautenhahn, "Robots we like to live with?!-a developmental perspective on a personalized, life-long robot companion," in *Robot and human interactive communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*. IEEE, 2004, pp. 17–22.

[15] M. K. Lee, J. Forlizzi, S. Kiesler, P. Rybski, J. Antanitis, and S. Savetsila, "Personalization in hri: A longitudinal field experiment," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. IEEE, 2012, pp. 319–326.

[16] I. Nahum-Shani, E. B. Hekler, and D. Spruijt-Metz, "Building health behavior models to guide the development of just-in-time adaptive interventions: A pragmatic framework." *Health Psychology*, vol. 34, no. S, p. 1209, 2015.

[17] S. Rossi, F. Ferland, and A. Tapus, "User profiling and behavioral adaptation for hri: A survey," *Pattern Recognition Letters*, 2017.

[18] M. Coeckelbergh, C. Pop, R. Simut, A. Peca, S. Pintea, D. David, and B. Vanderborght, "A survey of expectations about the role of robots in robot-assisted therapy for children with asd: Ethical acceptability, trust, sociability, appearance, and attachment," *Science and engineering ethics*, pp. 1–19, 2015.

[19] D. Feil-Seifer and M. J. Mataric, "Dry your eyes: examining the roles of robots for childcare applications," *Interaction Studies*, vol. 11, no. 2, p. 208, 2010.

[20] N. Giullian, D. Ricks, A. Atherton, M. Colton, M. Goodrich, and B. Brinton, "Detailed requirements for robots in autism therapy," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2595–2602.

[21] I. Zubrycki and G. Granosik, "Understanding therapists needs and attitudes towards robotic support. the roboterapia project," *International Journal of Social Robotics*, vol. 8, no. 4, pp. 553–563, 2016.

[22] M. Martin-Ortiz, M.-G. Kim, and E. I. Barakova, "Mobile application for executing therapies with robots," in *International Work-Conference on Artificial Neural Networks*. Springer, 2017, pp. 82–92.

[23] H.-L. Cao, P. G. Esteban, R. Simut, G. Van de Perre, D. Lefeber, B. Vanderborght, *et al.*, "A collaborative homeostatic-based behavior controller for social robots in human–robot interaction experiments," *International Journal of Social Robotics*, pp. 1–16.

[24] K. Dautenhahn and C. L. Nehaniv, "The correspondence problem," in *Imitation in Animals and Artifacts, MIT Press*. MIT Press, 2002.

[25] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 4, pp. 482–496, 2002.

[26] Q. A. Le, S. Hanoune, and C. Pelachaud, "Design and implementation of an expressive gesture model for a humanoid robot," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 134–140.

[27] M. Salem, S. Kopp, I. Wachsmuth, and F. Joublin, "Generating multimodal robot behavior based on a virtual agent framework," in *Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR)*, 2010.

[28] G. Van de Perre, A. De Beir, H.-L. Cao, P. G. Esteban, D. Lefeber, and B. Vanderborght, "Reaching and pointing gestures calculated by a generic gesture system for social robots," *Robotics and Autonomous Systems*, vol. 83, pp. 32–43, 2016.

[29] G. Van de Perre, M. Van Damme, D. Lefeber, and B. Vanderborght, "Development of a generic method to generate upper-body emotional expressions for different social robots," *Advanced Robotics*, vol. 29, no. 9, pp. 597–609, 2015.

[30] D. Brugali and A. Shakhimardanov, "Component-based robotic engineering (part ii)," *IEEE Robotics & Automation Magazine*, vol. 17, no. 1, pp. 100–112, 2010.

[31] D. Brugali and P. Scandurra, "Component-based robotic engineering (part i)[tutorial]," *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 84–96, 2009.

[32] D. Vernon, E. Billing, P. Hemeren, S. Thill, and T. Ziemke, "An architecture-oriented approach to system integration in collaborative robotics research projects: An experience report," *Journal of Software Engineering for Robotics*, vol. 6, no. 1, pp. 15–32, 2015.

[33] M. Y. Jung, M. Balicki, A. Deguet, R. H. Taylor, and P. Kazandzides, "Lessons learned from the development of component-based medical

robot systems," *Journal of Software Engineering for Robotics*, vol. 5, no. 2, pp. 25–41, 2014.

[34] D. Feil-Seifer and M. J. Mataric, "B3ia: A control architecture for autonomous robot-assisted behavior intervention for children with autism spectrum disorders," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*. IEEE, 2008, pp. 328–333.

[35] H.-L. Cao, C. Pop, R. Simut, R. Furnémont, A. De Beir, G. Van de Perre, P. G. Esteban, D. Lefeber, and B. Vanderborght, "Probolino: A portable low-cost social device for home-based autism therapy," in *Social Robotics*. Springer, 2015, pp. 93–102.

[36] J. Wainer, K. Dautenhahn, B. Robins, and F. Amirabdollahian, "A pilot study with a novel setup for collaborative play of the humanoid robot kaspar with children with autism," *International Journal of Social Robotics*, vol. 6, no. 1, pp. 45–65, 2014.

[37] R. Simut, J. Vanderfaeillie, A. Peca, G. V. de Perre, and B. Vanderborght, "Exploring joint attention: 9-18 months old typically developing infants in responding and initiating joint attention with a human partner and with the social robot Keepon," *Journal of Autism and Developmental Disorders, submitted for the Special Issue: Robotics & Technology and Computer Assistance*, 2014.

[38] R. C. Arkin, M. Fujita, T. Takagi, and R. Hasegawa, "An ethological and emotional basis for human–robot interaction," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 191–201, 2003.

[39] D. McColl, W.-Y. G. Louie, and G. Nejat, "Brian 2.1: A socially assistive robot for the elderly and cognitively impaired," *Robotics & Automation Magazine, IEEE*, vol. 20, no. 1, pp. 74–83, 2013.

[40] S. Calinon and A. Billard, "Pda interface for humanoid robots," in *Proceedings of the IEEE International Conference on Humanoid Robots (Humanoids)*, no. LSA3-CONF-2003-002, 2003.

[41] G. Pioggia, M. Sica, M. Ferro, R. Igliozzi, F. Muratori, A. Ahluwalia, and D. D. Rossi, "Human-robot interaction in autism: Face, an android-based social therapy," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*. IEEE, 2007, pp. 605–612.

[42] N. Lazzeri, D. Mazzei, A. Zaraki, and D. De Rossi, "Towards a believable social robot," in *Biomimetic and Biohybrid Systems*. Springer, 2013, pp. 393–395.

[43] J. Saldien, B. Vanderborght, K. Goris, M. Van Damme, and D. Lefeber, "A motion system for social and animated robots," *International Journal of Advanced Robotic Systems*, vol. 11, 2014.

[44] D. David, S.-A. Matu, and O. A. David, "Robot-based psychotherapy: Concepts development, state of the art, and new directions," *International Journal of Cognitive Therapy*, vol. 7, no. 2, pp. 192–210, 2014.

[45] C. A. Pop, S. Pintea, B. Vanderborght, and D. O. David, "Enhancing play skills, engagement and social skills in a play task in asd children by using robot-based interventions. a pilot study," *Interaction Studies*, vol. 15, no. 2, pp. 292–320, 2014.

[46] A. Tapus, C. Ţăpuş, and M. J. Matarić, "User–robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy," *Intelligent Service Robotics*, vol. 1, no. 2, pp. 169–183, 2008.

[47] C. M. Stanton, P. H. Kahn Jr, R. L. Severson, J. H. Ruckert, and B. T. Gill, "Robotic animals might aid in the social development of children with autism," in *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*. IEEE, 2008, pp. 271–278.

[48] D. A. Norman, A. Ortony, and D. M. Russell, "Affect and machine design: Lessons for the development of autonomous machines," *IBM Systems Journal*, vol. 42, no. 1, pp. 38–44, 2003.

[49] A. Ortony, D. Norman, and W. Revelle, "Affect and proto-affect in effective functioning. who needs emotions," *The Brain Meets the Robot*, 2005.

[50] A. Sloman and B. Logan, "Evolvable architectures for human-like minds," in *Affective Minds*, 2000.

[51] A. Sloman *et al.*, "Beyond shallow models of emotion," *Cognitive Processing: International Quarterly of Cognitive Science*, vol. 2, no. 1, pp. 177–198, 2001.

[52] A. B. Karami, K. Sehaba, and B. Encelle, "Adaptive artificial companions learning from users feedback," *Adaptive Behavior*, vol. 24, no. 2, pp. 69–86, 2016.

[53] A. Aly and A. Tapus, "Towards an intelligent system for generating an adapted verbal and nonverbal combined behavior in human–robot interaction," *Autonomous Robots*, vol. 39, no. 1, 2015.

[54] M. Joosse, M. Lohse, J. G. Pérez, and V. Evers, "What you do is who you are: The role of task context in perceived social robot personality," in *Robotics and automation (ICRA), 2013 IEEE international conference on*. IEEE, 2013, pp. 2134–2139.

[55] S. Woods, K. Dautenhahn, C. Kaouri, R. te Boekhorst, K. L. Koay, and M. L. Walters, "Are robots like people?: Relationships between participant and robot personality traits in human–robot interaction studies," *Interaction Studies*, vol. 8, no. 2, pp. 281–305, 2007.

[56] L. R. Goldberg, "An alternative" description of personality": the big-five factor structure." *Journal of personality and social psychology*, vol. 59, no. 6, p. 1216, 1990.

[57] R. R. McCrae and O. P. John, "An introduction to the five-factor model and its applications," *Journal of personality*, vol. 60, no. 2, pp. 175–215, 1992.

[58] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, no. 4, pp. 261–292, 1996.

[59] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.

[60] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann, "A model for personality and emotion simulation," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2003, pp. 453–461.

[61] P. Gebhard, "Alma: a layered model of affect," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 2005, pp. 29–36.

[62] A. Mehrabian, "Analysis of the big-five personality factors in terms of the pad temperament model," *Australian Journal of Psychology*, vol. 48, no. 2, pp. 86–92, 1996.

[63] T. Doce, J. Dias, R. Prada, and A. Paiva, "Creating individual agents through personality traits." in *IVA*. Springer, 2010, pp. 257–264.

[64] P. Gómez Esteban, H.-L. Cao, A. De Beir, G. Van de Perre, D. Lefeber, and B. Vanderborght, "A multilayer reactive system for robots interacting with children with autism," 2016.

[65] J. Shim and R. C. Arkin, "An intervening ethical governor for a robot mediator in patient-caregiver relationships," in *A World with Robots*. Springer, 2017, pp. 77–91.

[66] E. I. Barakova, J. C. Gillesen, B. E. Huskens, and T. Lourens, "End-user programming architecture facilitates the uptake of robots in social therapies," *Robotics and Autonomous Systems*, vol. 61, no. 7, pp. 704–713, 2013.

[67] H.-L. Cao, A. De Beir, P. G. Esteban, R. Simut, G. Van de Perre, D. Lefeber, and B. Vanderborght, "An end-user interface to generate homeostatic behavior for nao robot in robot-assisted social therapies," in *International Work-Conference on Artificial Neural Networks*. Springer, 2017, pp. 609–619.

[68] I. Zubrycki, M. Kolesiński, and G. Granosik, "Graphical programming interface for enabling non-technical professionals to program robots and internet-of-things devices," in *International Work-Conference on Artificial Neural Networks*. Springer, 2017, pp. 620–631.

[69] E. Senft, P. Baxter, J. Kennedy, and T. Belpaeme, "SPARC: Supervised progressively autonomous robot competencies," in *International Conference on Social Robotics*. Springer, 2015, pp. 603–612.

[70] E. Senft, P. Baxter, J. Kennedy, S. Lemaignan, and T. Belpaeme, "Supervised autonomy for online learning in human-robot interaction," *Pattern Recognition Letters*, 2017.

[71] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[72] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. Siam, 1998, vol. 61.

[73] J. Xu, J. Broekens, K. Hindriks, and M. A. Neerincx, "Mood expression through parameterized functional behavior of robots," in *2013 IEEE RO-MAN*, 2013, pp. 533–540.

[74] ——, "Mood contagion of robot body language in human robot interaction," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 6, pp. 1216–1248, 2015.

[75] R. A. Lippa and J. K. Dietz, "The relation of gender, personality, and intelligence to judges' accuracy in judging strangers' personality from brief video segments," *Journal of Nonverbal Behavior*, vol. 24, no. 1, pp. 25–43, 2000.

[76] K. M. Lee, W. Peng, S.-A. Jin, and C. Yan, "Can robots manifest personality?: An empirical test of personality recognition, social responses, and social presence in human–robot interaction," *Journal of communication*, vol. 56, no. 4, pp. 754–772, 2006.

[77] K. Amaya, A. Bruderlin, and T. Calvert, "Emotion from motion," in *Graphics interface*, vol. 96. Toronto, Canada, 1996, pp. 222–229.

[78] C. Pelachaud, "Studies on gesture expressivity for a virtual agent," *Speech Communication*, vol. 51, no. 7, pp. 630–639, 2009.

[79] A. Yamaguchi, Y. Yano, S. Doki, and S. Okuma, "A study of emotional motion description by motion modification rules using adjectival expressions," in *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, vol. 4. IEEE, 2006, pp. 2837–2842.

[80] C. Hu, M. A. Walker, M. Neff, and J. E. F. Tree, "Storytelling agents with personality and adaptivity," in *International Conference on Intelligent Virtual Agents*. Springer, 2015, pp. 181–193.

[81] H. Kim, S. S. Kwak, and M. Kim, "Personality design of sociable robots by control of gesture design factors," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*. IEEE, 2008, pp. 494–499.

[82] D. Vernon, E. Billing, P. Hemeren, S. Thill, and T. Ziemke, "An architecture-oriented approach to system integration in collaborative robotics research projects: An experience report," *Journal of Software Engineering for Robotics*, vol. 6, no. 1, pp. 15–32, 2015.

[83] P. Fitzpatrick, E. Ceseracciu, D. Domenichelli, A. Paikan, G. Metta, and L. Natale, "A middle way for robotics middleware," 2014.

[84] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: yet another robot platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, p. 8, 2006.

[85] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 769–774.

[86] B. Görer, A. A. Salah, and H. L. Akın, "An autonomous robotic exercise tutor for elderly people," *Autonomous Robots*, vol. 41, no. 3, pp. 657–678, 2017.

[87] A. Tapus, A. Peca, A. Aly, C. Pop, L. Jisa, S. Pintea, A. S. Rusu, and D. O. David, "Children with autism social engagement in interaction with nao, an imitative robot–a series of single case experiments," *Interaction studies*, vol. 13, no. 3, pp. 315–347, 2012.

[88] I. Schietecatte, H. Roeyers, and P. Warreyn, "Exploring the nature of joint attention impairments in young children with autism spectrum disorder: Associated social and cognitive skills," *Journal of autism and developmental disorders*, vol. 42, no. 1, pp. 1–12, 2012.

[89] C. Whalen and L. Schreibman, "Joint attention training for children with autism using behavior modification procedures," *Journal of Child Psychology and Psychiatry*, vol. 44, no. 3, pp. 456–468, 2003.

[90] A. A. Clair and A. G. Ebberts, "The effects of music therapy on interactions between family caregivers and their care receivers with late stage dementia," *Journal of Music Therapy*, vol. 34, no. 3, pp. 148–164, 1997.

[91] A. Tapus, "Improving the quality of life of people with dementia through the use of socially assistive robots," in *Advanced Technologies for Enhanced Quality of Life, 2009. AT-EQUAL'09*. IEEE, 2009, pp. 81–86.

[92] Hasfit, "20 min shoulder stretches & strengthening for pain relief," 01 2017. [Online]. Available: http://hasfit.com/workouts/home/senior/shoulder-stretches-strengthening-for-pain-relief/

[93] J. Kennedy, P. Baxter, E. Senft, and T. Belpaeme, "Higher nonverbal immediacy leads to greater learning gains in child-robot tutoring interactions," in *International conference on social robotics*. Springer, 2015, pp. 327–336.

[94] ——, "Social robot tutoring for child second language learning," in *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*. IEEE, 2016, pp. 231–238.

[95] G. Saucier, "Mini-markers: A brief version of goldberg's unipolar big-five markers," *Journal of personality assessment*, vol. 63, no. 3, pp. 506–516, 1994.

[96] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: the panas scales." *Journal of personality and social psychology*, vol. 54, no. 6, p. 1063, 1988.

[97] S. Park, L. Moshkina, and R. C. Arkin, "Recognizing nonverbal affective behavior in humanoid robots," in *Intelligent Autonomous Systems*, vol. 11, 2010, p. 12.

[98] A. M. Lund, "Measuring usability with the use questionnaire12," *Usability interface*, vol. 8, no. 2, pp. 3–6, 2001.

[99] D. Feil-Seifer and M. J. Matarić, "Toward socially assistive robotics for augmenting interventions for children with autism spectrum disorders," in *Experimental robotics*. Springer, 2009, pp. 201–210.

[100] L. Moshkina, S. Park, R. C. Arkin, J. K. Lee, and H. Jung, "Tame: Time-varying affective response for humanoid robots," *International Journal of Social Robotics*, vol. 3, no. 3, pp. 207–221, 2011.

Hoang-Long Cao received his B.Eng. in Mechatronics from Can Tho University in 2010, and M.Sc. in Electromechanical engineering from the Vrije Universiteit Brussel and Université Libre de Bruxelles in 2013. From September 2013, he has been working as a Ph.D. researcher at the Multibody Mechanics and Robotics research group, Vrije Universiteit Brussel, Belgium. His research interest is social robotics. Currently he is involved in DREAM project for the development of robot-enhanced therapy for children with autism spectrum disorders.

Greet Van de Perre studied Mechanical Engineering at the Vrije Universiteit Brussel. After graduating in 2011, she started her Ph.D. at the Robotics and Multibody Mechanics Research Group, funded by an FWO fellowship. Her research focuses on the development of a generic gesture system for social robots and the design of an actuated upper-torso for the social robot Probo.

James Kennedy received his Ph.D. from Plymouth University, U.K. in 2017 for his work using social robots to tutor children. During his PhD, he worked as a Research Assistant on the EU-funded DREAM (Development of Robot-Enhanced therapy for children with AutisM spectrum disorders) project, and collaborated with the ALIZ-E (Adaptive Strategies for Sustainable Long-Term Social Interaction), and L2TOR (Second Language Tutoring using Social Robots) projects. In 2016, James was a Lab Associate with Disney Research, Pittsburgh. His research interests lie in Human-Robot Interaction and Socially Intelligent Agents.

Emmanuel Senft received his Master of Science from EPFL in 2013 in Robotic and Autonomous Systems. He is a PhD student in Human Robot Interaction at Plymouth University since 2014. His interests are mainly interactive machine learning and social robotic and especially how to use humans to teach robots how to interact with other humans. He is involved in the EU FP7 DREAM project aiming at developing new robot enhanced therapies for children with autism.

Pablo Gómez Esteban received, in March 2014 from the Rey Juan Carlos University of Madrid, a Ph.D. degree for his thesis entitled "Cooperation and Competition in Emotional Robot Societies" with highest distinction. He is currently a post-doctoral researcher working at Vrije Universiteit Brussel in Belgium. His research was addressed towards decision making processes within competitive and cooperative environments; the impact that affective factors have in such processes; and designing and implementing those models to be used in non-expensive platforms within domestic environments. But his research interests also lie on cognitive architectures for social robots and human-robot interaction. He is the author or co-author of several chapters in books and published articles in international journals and conferences.

Albert De Beir received in 2014 his Master degree in electromechanical engineering from the Vrije Universiteit Brussel and in 2016 his Master degree in Psychology from the Université Libre de Bruxelles. His research interest focus on the development of artificial intelligence for social robots.

Ramona Simut received her M.S. degrees from Babes-Bolyai University in 2011. From 2008 to 2011 she worked as a psychologist at the Research Department of Autism Transylvania Association on the project of Using the social robot Probo in therapy for children with ASD. From September 2011 to present she is a Ph.D. student with the research focus on the clinical use of social robots for ASD children at the Clinical and Life Span Psychology Department of Vrije Universiteit Brussel. Her research interests include developing robot assisted therapies for children with autism, perception of the child-robot interaction, responses to robots of children with ASD.

Tony Belpaeme received his Ph.D. in Computer Science from the Vrije Universiteit Brussel in 2002 and is currently Professor in Robotics and Cognitive Systems at Plymouth University (UK) where he leads a research lab in the Centre for Robotics and Neural Systems. Starting from the premise that cognition is rooted in social interaction, Belpaeme and team try to further the science and technology behind artificial intelligence and social robots. This results in a spectrum of fundings, from theoretical insights to practical applications. He coordinated the FP7 ALIZ-E project, and collaborated on the ROBOT-ERA, DREAM and ITALK projects. He is currently the coordinator of the H2020 L2TOR project.

Dirk Lefeber received a degree in Civil Engineering in 1979 and a Ph.D. in Applied Sciences in 1986 from the Vrije Universiteit Brussel. Currently, he is full professor, former head of the department of Mechanical Engineering and head of the Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, which he founded in 1990. The research interests of the group are new actuators with adaptable compliance, dynamically balanced robots, robot assistants, rehabilitation robotics and multibody dynamics, all in the context of physical and cognitive human robot interaction with emphasis on improving the quality of life of human.

Bram Vanderborght focused his Ph.D. research on the use of adaptable compliance of pneumatic artificial muscles in the dynamically balanced biped Lucy. In May-June 2006, he performed research on the humanoids robot HRP-2 at the Joint Japanese/French Robotics Laboratory (JRL) in AIST, Tsukuba (Japan) in the research 'Dynamically stepping over large obstacles by the humanoid robot HRP-2'. From October 2007-April 2010 he worked as post-doc researcher at the Italian Institute of Technology in Genova (Italy) on the humanoid robot iCub and compliant actuation. Since October 2009, he is appointed as professor at the Vrije Universiteit Brussel. He has an ERR starting grant on SPEA actuation concept. He is member of the Young Academy of the Royal Flemish Academy of Belgium for Science and the Arts. His research interests include cognitive and physical human robot interaction, social robots, humanoids, robot assisted therapy and rehabilitation/assistive robotics. He is current EiC of IEEE Robotics & Automation Magazine.

Period 5

Designing the social robot Elvis: how to select an optimal joint configuration for effective gesturing

Greet Van de Perre, Hoang-Long Cao, Albert De Beir, Pablo Gómez Esteban, Dirk Lefeber and Bram Vanderborght

Abstract

Depending on their exact application, social robots have been designed with different arm morphologies, ranging from under-actuated designs to arms featuring 9 degrees of freedom (DOF). It is however difficult to investigate if the chosen arm morphology is the best possible solution for the intended application. Existing robots differ from each other in a large range of aspects an therefore, 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 proposed a methodology based on the calculation of gestures for different morphologies and their visualization on one single virtual model. In this paper, we present the design process followed to select and optimal morphology for our new social robot Elvis. A gesture study was performed to select an optimal morphology, resulting in three interesting joint configurations that were studied in more detail and later physically developed. To realize the three different morphologies, the arm system was designed semi-modular, allowing different modules to be switched. Different gestures were generated for all three Elvis-variants, resulting in the selection of one final morphology to be used for future work.

Keywords

social robot, arm design, arm morphology, gestures

1 Introduction

Social robots are aimed to be operated by untrained users, including children, elderly people, therapists and teachers. Therefore, 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 that are more or less based on a model of the human body. ASIMO (Hirai, Hirose, Haikawa and Takenaka, 1998), ARMAR-III (Albers, Brudniok, Ottnad, Sauter and Sedchaicharn, 2006), WABIAN-2 (Ogura, Aikawa, Shimomura, Kondo, Morishima, ok Lim and Takanishi, 2006) and iCub (Tsagarakis, Metta, Sandini, Vernon, Beira, Becchi, Righetti, Santos-Victor, Ijspeert, Carrozza and Caldwell, 2007) all feature 7 DOF arms, consisting of a 3 DOF shoulder,

1 DOF elbow and 3 DOF wrist. When designed for a certain application, some robots may have a different morphology to optimally fulfil the desired tasks. Pyrène of the TALOS series, for example, is developed 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. In addition, this placement benefits the compactness of the robot when the arms are folded to the front, allowing it to pass narrow sections in its working space (Stasse, Flayols, Budhiraja, Giraud-Esclasse, Carpentier, Del Prete, Soueres, Mansard, Lamiraux, Laumond, Marchionni, Tome and Ferro,

Robotics and Multibody Mechanics Research Group
Vrije Universiteit Brussel, Belgium
Flexible Assembly Flanders Make
www.brubotics.eu,
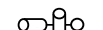
Corresponding author:
Greet Van de Perre
Email: Greet.Van.de.Perre@vub.ac.be

2017). The robot WE-4RII (Itoh, Miwa, Matsumoto, Zecca, Takanobu, Roccella, Carrozza, Dario and Takanishi, 2004), on the other hand, was developed to study human-like emotion. Next to a 3 DOF shoulder part, 1 DOF elbow part and 3 DOF wrist part, the robot was designed with an additional a 2 DOF base shoulder part, simulating the human scapula joint. These additional DOF allow the robot to shrug or square the shoulders, and as, such to create more human-like emotional expressions. Robovie's (Ishiguro, Ono, Imai, Maeda, Kanda and Nakatsu, 2001) design is more minimalistic. Since it's applications were mostly focussed on object indication and route direction-giving, mostly deictic gestures were aimed to be used. The robot features arms with only 4 DOF, consisting of a 3 DOF shoulder part and a 1 DOF elbow. Other robots with less articulated arms are KHR-3 (Park, Kim, Lee and Oh, 2005) and NAO (Gouaillier, Hugel, Blazevic, Kilner, Monceaux, Lafourcade, Marnier, Serre and Maisonnier, 2009). Both robots feature a 3 DOF shoulder consisting of a yaw, pitch and roll joint and 1 DOF elbow. But while NAO's wrist only consists a roll-joint, KHR-3's wrist consists of a yaw and pitch joint. For the robot R1, a parallel mechanism was used for the wrist actuation. The advantages of a high payload and structural stiffness, combined with the lightness of the platform fulfilled the important requirement of manipulating and carrying objects. 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 wel 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 (Parmiggiani, Luca Fiorio and, Sureshbabu, Randazzo, Maggiali, Pattacini, Lehmann, Tikhanoff, Domenichelli, Cardellino, Congiu, Pagnin, Cingolani, Natale and Metta, 2017). ARMAR-IV is another example of a robot featuring 8 DOF arms. To achieve more dexterity in bimanual manipulation, the typical 7 DOF arm was extended with the sternoclavicular joint (Asfour, Schill, Peters, Klas, Bücker, Sander, Schulz, Kargov, Werner and Bartenbach, 2013). Table 1 visualises the joint configuration of the robots discussed above, when placed in T-pose, the hand palm facing out.

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

Table 1. Different robots have been designed with different arm morphologies. This table lists a number of joint configurations, placed in T-pose with the hand palm facing out. The different joints are grouped into a the clavicle, shoulder, elbow and wrist module.

Robot	Clav	Shldr	Elbow	Wrist
ASIMO*				
ARMAR-III*				
Wabian-2*				
iCub*				
Pyrène				
WE-4RII*				
Robovie				
KHR-3*				
NAO*				
R1				
ARMAR-IV				
Elvis-Ca*				
Elvis-Cb*				
Elvis-Cc*				

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 proposed a methodology based on the calculation of gestures for different morphologies and their visualization on one single virtual model (Van de Perre, Cao, De Beir, Esteban, Lefeber and Vanderborght, 2018).

The methodology is based on the use of our developed gesture method, which, thanks to its generic framework, allows the calculation of gestures for different morphologies. 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 method uses a minimal amount of morphological data, inputted by the user, to evaluate the generic framework. To calculate different types of gestures, our method was designed to work in two modes. The block mode, presented in Van de Perre, Van Damme, Lefeber and Vanderborght (2015) is used to calculate gestures whereby the overall pose of the arms is important. The end-effector mode, on the other hand, was developed to generate gestures whereby the placement of the end-effector is crucial. In addition, blended gestures can be calculated by mixing the two implemented modes, and neutral behaviors can be modulated into affective gestures by modulating the amplitude and the motion speed (Van de Perre, De Beir, Cao, Esteban, Lefeber and Vanderborght, 2017).

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.

In this paper, we present how this methodology was followed to select an optimal design for our new developed robot, Elvis (figure 1). The robot is developed within the frame of the Probo-project, which aims to study cognitive human-robot interaction and the possibilities of robot assisted therapy, with a special focus on children. Section 2 discusses how three joint configurations were selected and studied in more detail. The followed design principles are briefly highlighted in section 3. A set of gestures were generated by all three Elvis-variants, for both the covered and the uncovered model and are discussed in section 4.

2 Joint configurations

2.1 Design methodology

To select a suitable joint configuration for the arms to be designed, a gesture study was performed for 7 different morphologies. 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 (Van de Perre, Cao, De Beir, Esteban, Lefeber and Vanderborght, 2018). The joint collocations of table 1 that were included in the study are denoted with an asterisk. For every configuration, suitable Denavit-Hartenbergh frames were assigned to the joints, and the corresponding parameters were calculated and served as input to the gesture software. Subsequently, a set of emotional expressions, including one for *anger*, *disgust*, *happiness* and *sadness* could be
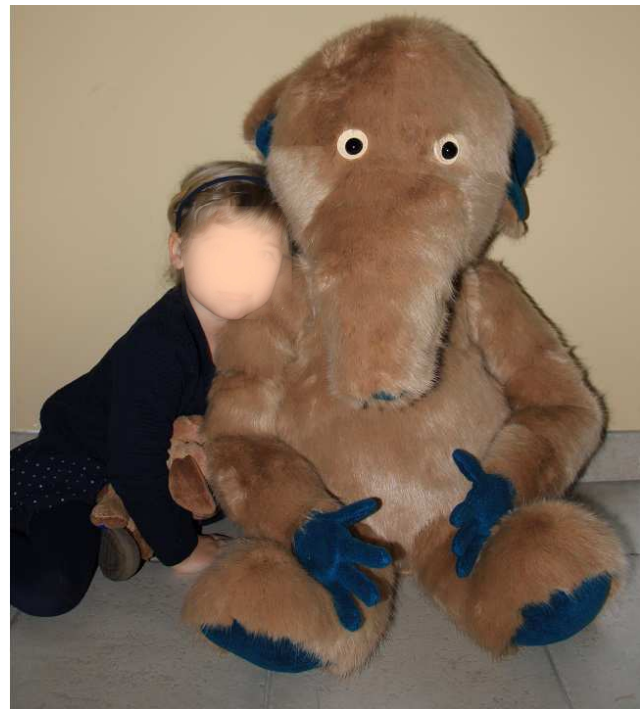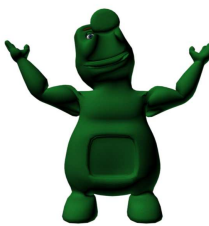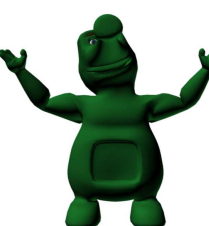
Figure 1. The social robot Elvis.

calculated. The methods output is a series of data files containing the calculated joint trajectories. To visualize the calculated gestures, the joint trajectories were loaded into Autodesk's 3DS Max and applied on the virtual model of the original Probo robot. For all different configurations, the gesture's end postures, and therefore the most expressive ones, were studied and compared. A second important constraint, next to the ability of generating recognizable emotional expressions, is to keep the design low-cost and low-complex. Therefore, a trade-off between expressibility and the number of actuated joints was made. This process resulted in the selection of three joint configurations, that appeared to be interesting to consider in more detail.

2.2 Elvis-Ca

The top row of table 2 visualizes the results of the gesture study for configuration (a). The first row shows the used joint configuration 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. Different emotional expressions, including one for *anger*, *disgust*, *happiness*, *sadness* were calculated using our developed gesture

Table 2. Results of the gesture study for the three joint configurations, selected to study in more detail. First row; configuration (a), second: configuration (b), third: configuration (c). The first column visualizes the joint configuration.

Joint configuration	Anger	Disgust	Happiness	Sadness
(a)				
(b)				
(c)				

method (Van de Perre, Van Damme, Lefeber and Vanderborght, 2015). The end postures are listed in the remaining columns. Observing the resulting postures learns that this configuration has a good performance of the calculated gestures, for a relatively low complex arm chain of 5 DOF.

2.3 Elvis-Cb

Configuration b, visualised in the second row of table 2, only differs from the previous configuration by the replacement of the joint responsible for pronation/supination of the forearm by one generating a flexion/extension motion. This configuration appears to be less interesting to generate the emotional expressions of *anger*, *disgust* and *sadness*. The resulting end postures suggest that the missing joint responsible for the pronation and supination of the forearm, is important for a good performance of the gestures. However, for other types of gestures, such as

a number of emblems, the flexion and extension of the wrist is a necessary motion to naturally execute the arm movements. Examples are the hand movement for *stop*, or clapping the hands. Therefore, it is worthwhile to take a deeper look into this joint configuration. By choosing the initial placement of the wrist differently, it can become an interesting morphology for generating emotional expressions too.

For the results visualized in table 2, 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 3 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
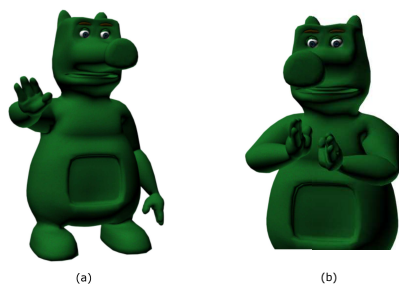
Figure 2. Positioning the hand according to a pronation of 60° with respect to the original T-pose after coupling the configuration (b) to the virtual model, allows a good performance of a number of emblem gestures, such as (a) the *stop*-gesture, and (b) clapping the hands.

the morphology used table 2; the model is placed in T-pose with the palm facing out. In the second column, a pronation of 30° 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 an angle of, respectively, 45°, 60° and 90°. The calculated end postures show that a pronation of 60° 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 2).

2.4 Elvis-Cc

The last morphology, configuration (c), features an arm configuration consisting of only 4 DOF (last row of table 2), composed of a 2 DOF shoulder, 1 DOF elbow and 1 DOF wrist. 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 an interesting test case. The gesture study resulted in a reasonable performance of the postures, and therefore, this morphology can be interesting to realize in physical form.

3 Design principles

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. 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. 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. To achieve a huggable Probo-look, the mechanical system was covered with a soft fur-jacket.

4 Results

4.1 Uncovered model

Table 4 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*. The results are in line with the expectations raised by the gesture study. Because Elvis-Cb misses the joint responsible for the internal rotation of the forearm, and a constant pronation of 60° was applied on the forearm, a difference in arm placement can be noted for this configuration. As predicted by the tests with the virtual model, the results in table 4 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 5, 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. Elvis-Cb 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

*http://probo.vub.ac.be/GestureMethod/Elvis.htm

Table 3. By altering the hand orientation of the robot's model after coupling it to the configuration (b), a different robot appearance, and therefore, different end postures can be reached.

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

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.

4.2 Covered model

The same gestures were created for the covered robot. The emotional expressions are listed in table 6, while the other gestures can be found in table 7. For the internal rotations, the fur was created as two separate pieces to guarantee 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° as for the uncovered model, only an angle of 90° 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 6), as well as in the *thumbs up* gesture for Elvis-Cb (table 7). 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 4, 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.
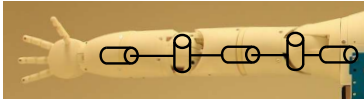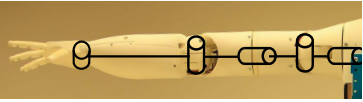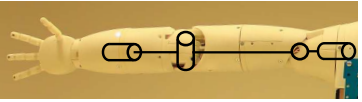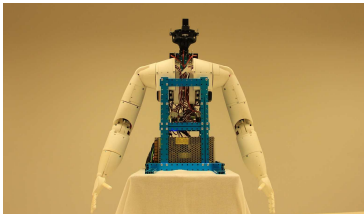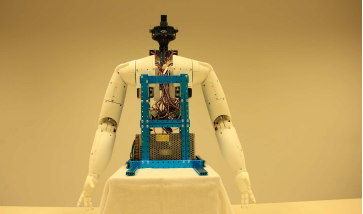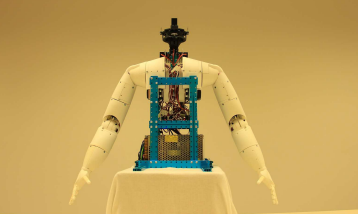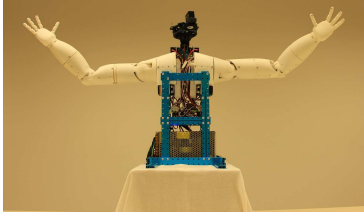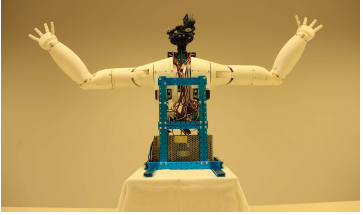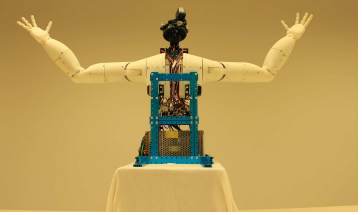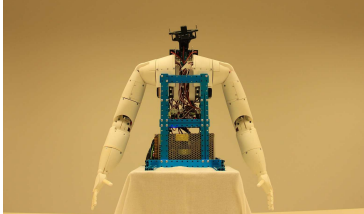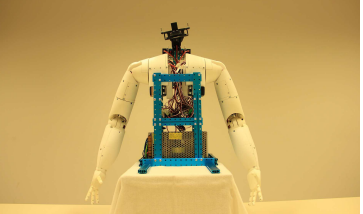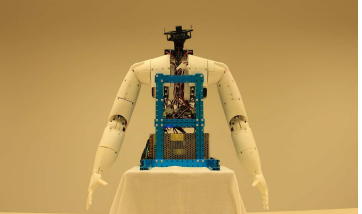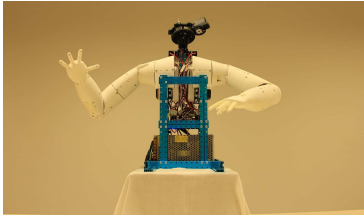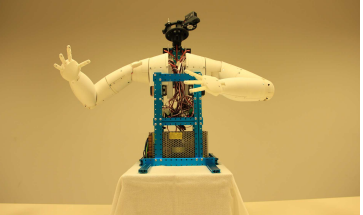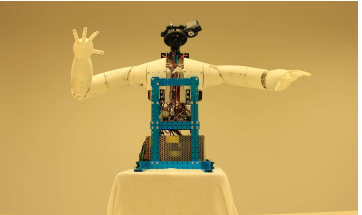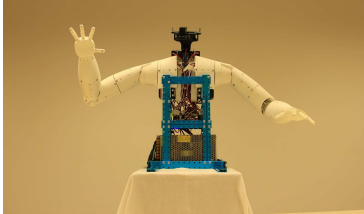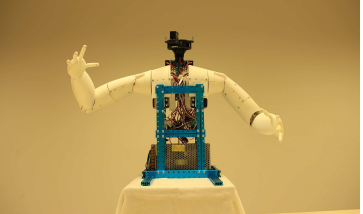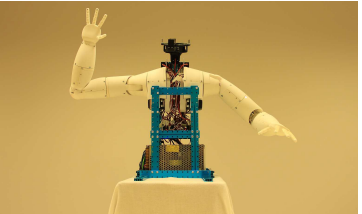
5 Conclusions

In this paper, we presented the design process followed for the social robot Elvis. The robot was developed in the frame of the Probo-project, which aims to study cognitive human-robot interaction and the possibilities of socially assistive robots, with a special focus on children. To select a morphology, optimally suited for its desired application, a gesture study was performed to investigate the effect of different design aspects

on the expressibility. Since the robot is aimed to interact with children on an emotional level, in a first step, a set of emotional expressions was used. The gestures were calculated for a number of different joint configurations, using our previously developed gesture method. Given the highly generic aspect of the method, gestures can be calculated for a different morphology with a minimal effort of the programmer. Three configurations were selected to study in more detail, and were later physically developed. The arm system was designed semi-modular, to allow different joint modules to be switched, in order to realize the three different Elvis-variants. To achieve a characteristic huggable Probo-look, the complete mechanical system was covered with a customized fur-jacket. Different gestures were generated for both the uncovered as the covered prototype. As a first set, the same set of emotional expressions used in the gesture study were generated. 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. 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 the joint allowing the flexion/extension of the wrist, realised in Elvis-Cb, contributed to the performance of the gesture. The internal rotation of the forearm, realised in Elvis-Ca, 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*. To work properly and guarantee the provided joint range, the elbow cover should be redesigned to reduce the friction.

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 it's performance of the tested emotional expressions,

Table 4. 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.

	Elvis-Ca	Elvis-Cb	Elvis-Cc
Config			
Neutral			
Happiness			
Sadness			
Disgust			
Anger			

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. Therefore, this will be the configuration to work with in future work.

In this stage, we focused on designing an optimal arm system to generate gestures. To guarantee a safe

Table 5. 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.
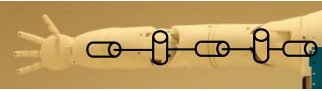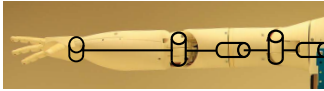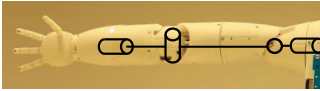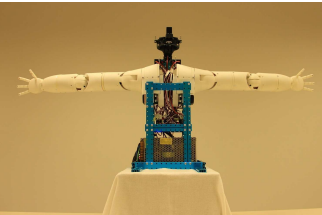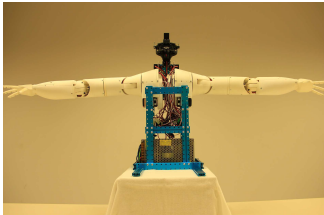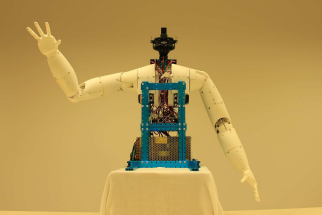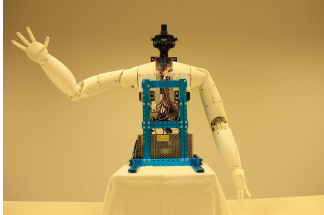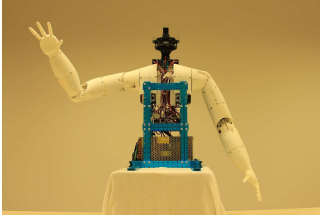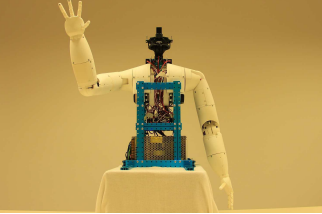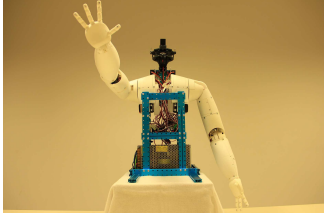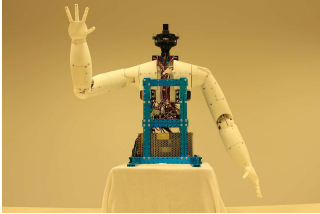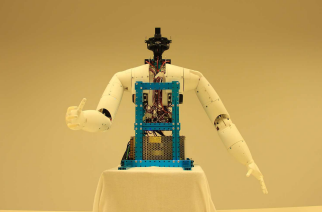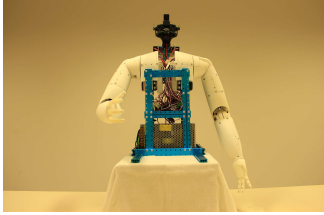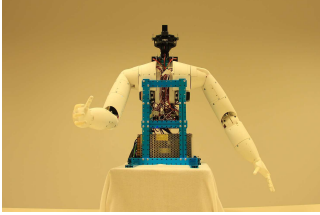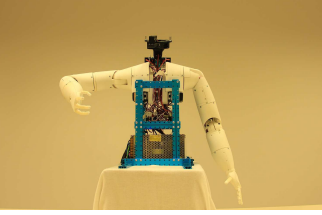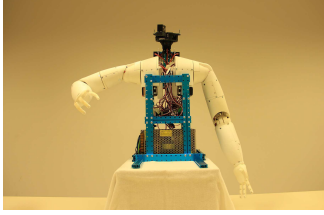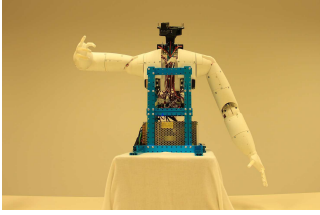
	Elvis-Ca	Elvis-Cb	Elvis-Cc
Config			
T-pose			
Hello			
Stop			
Thumbs up			
Point			
Clapping			

Table 6. 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 7. 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.

interaction with children, the actuation mechanism should be improved. Most importantly, a feedback concerning the servos' current position should be provided to the global control software. Since at this stage hobby servos are used, the positional feedback is restricted to the servo's internal control circuit. Other possible improvements include the use of gravity compensation techniques to reduce the necessary motor torques and providing non-back drivable mechanisms to protect the motors.

Funding

References

Albers A, Brudniok S, Ottnad J, Sauter C and Sedchaicharn K (2006) Upper body of a new humanoid robot-the design of armar iii. In: *Humanoid Robots, 2006 6th IEEE-RAS International Conference on.* IEEE, pp. 308–313.

Asfour T, Schill J, Peters H, Klas C, Bücker J, Sander C, Schulz S, Kargov A, Werner T and Bartenbach V (2013) Armar-4: A 63 dof torque controlled humanoid robot. In: *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on.* IEEE, pp. 390–396.

Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J and Maisonnier B (2009) Mechatronic design of nao humanoid. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, pp. 769–774.

Hirai K, Hirose M, Haikawa Y and Takenaka T (1998) The development of honda humanoid robot. In: *IEEE International Conference on Robotics and Automation (ICRA 1998)*, volume 2. pp. 1321–1326.

Ishiguro H, Ono T, Imai M, Maeda T, Kanda T and Nakatsu R (2001) Robovie: an interactive humanoid robot. *Industrial robot: An international journal* 28(6): 498–504.

Itoh K, Miwa H, Matsumoto M, Zecca M, Takanobu H, Roccella S, Carrozza M, Dario P and Takanishi A (2004) Various emotional expressions with emotion

expression humanoid robot WE-4RII. In: *IEEE Technical Exhibition Based Conference on Robotics and Automation.* pp. 35–36.

Ogura Y, Aikawa H, Shimomura K, Kondo H, Morishima A, ok Lim H and Takanishi A (2006) Development of a new humanoid robot WABIAN-2. In: *IEEE International Conference on Robotics and Automation (ICRA 2006).* pp. 76– 81.

Park IW, Kim JY, Lee J and Oh JH (2005) Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In: *Humanoid Robots, 2005 5th IEEE-RAS International Conference on.* IEEE, pp. 321–326.

Parmiggiani A, Luca Fiorio and AS, Sureshbabu AV, Randazzo M, Maggiali M, Pattacini U, Lehmann H, Tikhanoff V, Domenichelli D, Cardellino A, Congiu P, Pagnin A, Cingolani R, Natale L and Metta G (2017) The design and validation of the r1 personal humanoid. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

Stasse O, Flayols T, Budhiraja R, Giraud-Esclasse K, Carpentier J, Del Prete A, Soueres P, Mansard N, Lamiraux F, Laumond JP, Marchionni L, Tome H and Ferro F (2017) Talos: A new humanoid research platform targeted for industrial applications .

Tsagarakis NG, Metta G, Sandini G, Vernon D, Beira R, Becchi F, Righetti L, Santos-Victor J, Ijspeert AJ, Carrozza MC and Caldwell D (2007) iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics* 21(10): 1151–1175.

Van de Perre G, Cao HL, De Beir A, Esteban PG, Lefeber D and Vanderborght B (2018) Studying design aspects for social robots using a generic gesture method. *International Journal on Social Robotics, submitted* .

Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D and Vanderborght B (2017) Generic method for generating blended gestures and mood expressions for social robots. *Autonomous Robots* .

Van de Perre G, Van Damme M, Lefeber D and Vanderborght B (2015) Development of a generic method to generate upper-body emotional expressions for different social robots. *Advanced Robotics* 29(9): 59–609.

Studying Design Aspects for Social Robots Using a Generic Gesture Method

Greet Van de Perre[1,2] · Albert De Beir[1,2] · Hoang-Long Cao[1,2] · Pablo Gómez Esteban[1,2] · Dirk Lefeber[1,2] ·
Bram Vanderborght[1,2]

Abstract

Since social robots are aimed to interact and communicate with humans in a natural way and operate 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 their specific application. In this paper, we propose a novel methodology to study the influence of different design aspects, based on a generic gesture method. The gesture method was developed to overcome the difficulties in transferring gestures to different robots, providing a solution for the correspondence problem. A small set of morphological information, inputted by the user, is used to evaluate the generic framework of the software at runtime. Therefore, gestures can be calculated fast and easy for a desired robot configuration. By generating a set of gestures for different morphologies, the importance of specific joints and their influence on a series of postures and gestures 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 specified task, and can give interesting insights in the required joint angle range. In this paper, this design methodology is illustrated by using the virtual model of the robot Probo.

Keywords Upper body design · Generic gesture method · Affective motions · Gestures

1 Introduction

The design of social robots is a challenging task. Since social robots are aimed to interact with people in an interpersonal manner, it is important to ensure a natural, intuitive interaction. Therefore, social robots need to be able to communicate using both verbal and nonverbal signs [1]. 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 (Fig. 1a) [2], ARMAR-III (Fig. 1b) [3], WABIAN-2 (Fig. 1c) [4] and iCub (Fig. 1d) [5]. 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 (Fig. 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 [6]. The robot WE-4RII [7], on the other hand, was developed to study human-like emotion. Humans feature a scapula joint, allowing us 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 a 2 DOF base shoulder part, enabling the robot to create more human-like emotional expressions (Fig. 2b). Robovie's [8] design is more minimalistic. It's applications were mostly

✉ Greet Van de Perre
Greet.Van.de.Perre@vub.be

1 Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Brussels, Belgium

2 Flexible Assembly Flanders Make, Louvain, Belgium
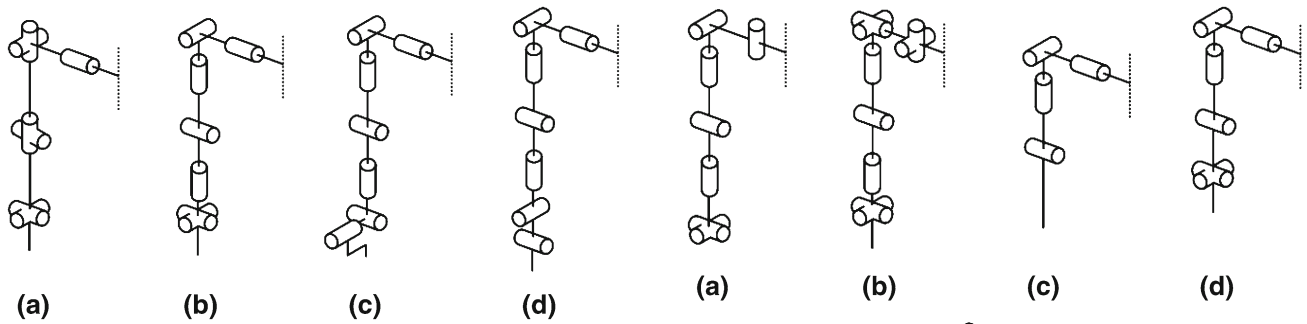
🙋 Springer

Fig. 1 Visualization of the right arm joint configuration of a number of robots featuring 7 DOF arms. The dotted line represents the midline of the body. **a** ASIMO, **b** ARMAR-III, **c** WABIAN-2, **d** iCub

Fig. 2 Visualization of the right arm joint configuration of a number of robots with different arm morphologies. The dotted line represents the midline of the body. **a** Pyrène, **b** WE-4RII, **c** Robovie, **d** KHR-3, **e** NAO, **f** R1, **g** ARMAR-IV

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 (Fig. 2c). Other robots with less articulated arms are KHR-3 [9] and NAO [10]. Both robots feature a 3 DOF shoulder consisting of a yaw, pitch and roll joint and 1 DOF elbow. But while NAO's wrist only consists a roll-joint (Fig. 2d), KHR-3's wrist consists of a yaw and pitch joint (Fig. 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 wel 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 (Fig. 2f) [11]. 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 (Fig. 2g) [12].

Because of the differences in arm morphology, sharing gestures between robots is not straightforward. When working with different robot platforms, new joint trajectories have to be calculated for every desired gesture, taking into account the specific morphology of the robot under consideration. This issue is known as the correspondence problem. Because of the differences in morphology, and also the general outer appearance defined by the robot's dimensions and proportions, it is not straightforward to investigate the effect of the placement of separate joints in existing robots and compare the performance of gestures with that of other robot configurations. To give insights in the influence of different design aspects on the performance of specified motions and help in making substantiated trade-off's in the design pro-
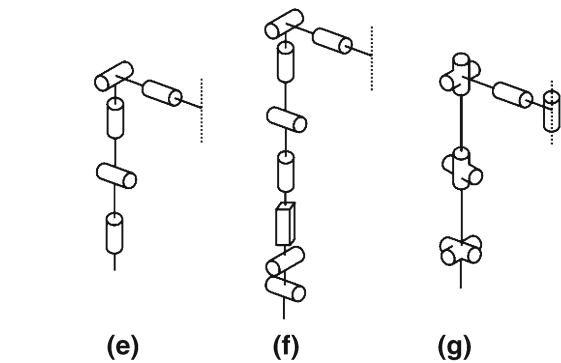
cess of new robots, we propose a methodology based on the calculation of gestures for different morphologies and their visualized on one single virtual model. In previous publications, we proposed a solution for the correspondence problem by designing a generic method to generate gestures for social robots [13–15]. The method only uses a small set of morphological information as input to calculate a variety of gestures for the desired robot. Given the flexibility of the method, generating gestures for different morphologies is very fast and straightforward. Using this method, we now propose a novel methodology to study the influence of different design aspects of social robots on a set of predefined behaviors. By generating a set of gestures for different morphologies, the importance of specific joints and their influence on the performance of the mapped gestures can be studied. Using this methodology in the design process can give interesting insights in the design requirements and help determining the optimal design/complexity trade-off for a certain application.

In the next section, the working principles of the developed gesture method are briefly discussed. Section 3 covers how the gesture method can be useful in the design process of new social robots. The study of the influence of the exact configuration and the joint angle range is illustrate by using the model of the robot Probo and is discussed in respectively Sects. 3.1 and 3.2.

2 Working Principles of the Method

To ensure a generic method usable for different kind of robots, a *human base model* was defined to serve as reference to construct the framework of the method. The model consists of different *chains* and *blocks* and models the rotational possibilities of a human. To quantify the motions, a reference frame was assigned to each joint block. The human base model was used as a reference to construct the general framework of the method. To generate gestures for a certain robot or model, the method uses the Denavit–Hartenberg (DH) parameters of the configuration as input, whereby the different joints of the robot are grouped into the chains and blocks of the human base model. At runtime, the generic framework of the method is evaluated using this information, and as such, adapted to the robot under consideration.

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, which are full body actions representing an internal emotional state [13]. 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 [14].

2.1 Block Mode

The block mode [13] is used for gestures whereby the overall placement of the arms is important, such as for emotional expressions. In this mode, the method uses a set of target gestures stored in a database and maps them to a selected configuration. To ensure a good overall posture, it is not sufficient to only impose the pose of the end effector, since inverse kinematics for robots with a different configuration and different relative arm lengths could result in unrecognisable global postures. Therefore, the orientation of every joint complex the robot has in common with a human needs to be considered. Hence, the target gestures are stored in the database by specifying the orientation of every joint block i of the base model using the orthopaedic angles [16] of frame $i+1$ (the base frame of block $i+1$) with respect to frame i (the base frame of block i). To make a robot or model perform a selected expression, a mapped rotation matrix for every present joint block is calculated by combining the information from the database and the morphological data specified by the user:

$$R_i = {}^{b,i}R_{st} \cdot R_{i,des} \cdot {}^{st}R_{e,i} \tag{1}$$

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 $i + 1$.

These mapped matrices serve as input for an inverse kinematics algorithm to calculate the necessary joint angles to make the specified robot configuration perform the desired expression. For every block, the joint angle derivatives $\dot{q}$ are calculated using the following closed loop inverse kinematics (CLIK) algorithm [17]:

$$\dot{q} = J_A^\dagger(q)(\dot{x_d} + K(x_d - x_e)) + \left(I - J_A^\dagger(q)J_A(q)\right)\dot{q_0} \tag{2}$$

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 $J_A(q)$. Since only rotational information is imposed, $J_A(q)$ is reduced to its rotational part only. x_e is the current end effector pose; i.e. the current $zyx-$Euler angles, and K a positive definite gain matrix. Since the different blocks are treated separately, no redundancy is present, causing the second term $\left(I - J_A^\dagger(q)J_A(q)\right)\dot{q_0}$ to be zero. For more detailed information and a validation of the block mode, we refer to our previous publication [13].

To guarantee the calculated joint angles to be in the possible physical range of the robot, the joint limitation algorithm proposed by Drexler and Harmati [18] was implemented. Their methodology prevents the violation of the joint limits by transforming the joint variables q_i to a set of fictive joint variables z_i:

$$q_i = \beta_i(z_i) \tag{3}$$

whereby the domain of β equals $[-\infty, \infty]$, whereas its range is $[q_{min,i}, q_{max,i}]$.

By expressing the kinematic equations in terms of the fictive variables z_i, and calculating the real joint values q_i by Eq. 3, 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 Eq. 2, whereafter they are transferred to the transformed joint space:

$$\dot{z} = d\beta^{-1}(z)\dot{q} \tag{4}$$

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 an implemented Runge–Kutta algorithm. Finally, the joint angles q can be acquired by using Eq. 3. Using this technique, the joint limits cannot be

violated. However, when reaching a boundary, the derivative of the corresponding function β_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:

$$d\beta_{ii}^\dagger = \begin{cases} \frac{1}{\frac{\partial \beta_i}{\partial z_i}} & \text{if } \frac{\partial \beta_i}{\partial z_i} \geq \epsilon \\ 0 & \text{else} \end{cases} \tag{5}$$

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 Eq. 4 to calculate the transformed joint derivatives, an extended formula is used:

$$\dot{z} = d\beta^\dagger(z)\dot{q} + \left(I - d\beta^\dagger(z)J_A^\dagger J_A d\beta(z)\right) y \tag{6}$$

A second important limitation factor to take into account are the joint speeds. To ensure the speeds to stay within their boundaries, a saturation on the joint speed calculated by Eq. 2, $\dot{q}_{calc}$, is included in the algorithm:

$$\dot{q} = \begin{cases} \dot{q}_{calc} & \text{if } -\dot{q}_{max} \leq \dot{q}_{calc} \leq \dot{q}_{max} \\ sign(\dot{q}_{calc})\dot{q}_{max} & \text{else} \end{cases} \tag{7}$$

2.2 End-Effector Mode

2.2.1 Place-At Condition

The end-effector mode [14] is used for gestures whereby the position of the end-effector is crucial, like for deictic gestures. In some situations, for example when reaching for an object, the position of the right and/or left hand is important and specified by the user. This situation is called the *place-at* condition. The specified position then serves as a basis to calculate the necessary end-effector position for the selected chain, which is used as input for the same inverse kinematics algorithm as used in the block mode (Eq. 2 in combination with Eq. 6). While in the block mode, a constraint is imposed on the end-effector of every block and the inverse kinematics algorithm is used to calculated the joint angles of every block separately, in the end-effector mode a constraint is imposed on the end-effector of the chain, and the algorithm is used to calculate the joint angles of the chain as a whole. Since in the end-effector mode the position is specified, the desired end effector pose x_d is limited to positional information only, reducing $J_A(q)$ to its translational part only. 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 second term of Eq. 2 will differ from zero, activating the

influence of $\dot{q}_0$ on the calculated joint speeds. $\dot{q}_0$ introduces the cost function $w(q)$:

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^T \tag{8}$$

with k_0 a positive weight factor. For the cost function w, we decided to work with a slightly adapted form of the joint range availability (JRA) criterion [19], whereby an optimal human like posture is calculated by keeping the joints close to a set of minimum posture angles q_{mi} (for more information, see our previous publication [14]):

$$w = \sum_{i=1}^{n} w_{0,i} \frac{(q_i - q_{mi})^2}{\left(q_{max,i} - q_{min,i}\right)^2} \tag{9}$$

Here q_i is the current value of joint i and q_{mi} the minimum posture angle for that joint. $q_{max,i}$ and $q_{min,i}$ are the maximum and minimum joint limits, and $w_{0,i}$ a weight factor for joint i.

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. For such cases, we provided the possibility of expressing an emotional state through an ongoing reaching or pointing gesture by modulating it, using a certain set of characteristic performance parameters. Based on the results of several expressivity models [20–24], we decided to focus on two modification parameters: the motion speed and the posture amplitude. The posture amplitude refers to the spatial extent; the amount of space occupied by the body. By making the minimum posture angles q_{mi} used in Eq. 9 dependent of the valence value, the openness of the posture will depend on the current state of affect. Therefore, for the joints mostly influencing the openness of the posture, a linear function of the valence is provided instead of the fixed minimum posture angle, which we, in this case, call the *affective posture angle* q_{ai} [15]:

$$q_{ai} = q_{ai,min} + val * (q_{ai,max} - q_{ai,min}) \tag{10}$$

2.2.2 Pointing Condition

The *pointing* condition uses the same calculation principles as the *place-at* condition. However, in contrast to the specified hand position in the place-at condition, no direct constraint is imposed on the end-effector by specifying a desired pointing position. The pointing constraint can be fulfilled by a series of configurations with a specific combination of end-effector position and orientation. To calculate the optimal end posture, the end-effector is gradually virtually extended. For every virtual length, the pointing position

is imposed on the current virtual end-effector and the corresponding posture is determined using Eq. 2. From the resulting collection of postures, the cost function (Eq. 9) finally selects the optimal result by comparing the total cost of every configuration.

2.3 Trajectory

After verifying if the desired end-effector position is located in the range of the robot, a suitable trajectory towards this point is is generated by calculating intermediate key frames. Also for the block mode, a trajectory between the starting pose and the desired end pose is determined by calculating the joint angles corresponding to a set of intermediate end-effector poses:

$$x_d(t_k) = f(x_{e,t_{start}}, x_{d,t_{end}}, t)$$

$$\text{with} \quad \begin{cases} x_d(t_{start}) = x_{e,t_{start}} \\ x_d(t_{end}) = x_{d,t_{end}} \end{cases} \qquad (11)$$

Here $x_d(t_k)$ is the desired pose at time t_k. Depending on the activated mode, this is a position or orientation and applied to the end-effector of a block or chain. $x_{e,t_{start}}$ is the actual end-effector starting pose of the block or chain under consideration, and $x_{d,t_{end}}$ the desired end pose. The exact nature of the trajectory function f is dependent on the activated mode and the location of the start and end pose in the workspace.

The total duration of the gesture t_{end} for an emotional expression is specified in the gesture database. For a gesture calculated by the end-effector mode, t_{end} is dependent on a speed factor, which on its turn, is dependent on the valence and arousal value of the current state of affect. As such, also the speed of the gesture serves a a modification parameter to influence the ongoing functional behavior according to the current affective state [15].

The calculated joint trajectories are stored by a series of data files, specifying the joint angles for every time interval.

2.4 Blended Gestures

Humans use and combine different types of gestures during natural communication. By combining the two working modes of our method, blended emotional expressions and deictic gestures can be generated. 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. This means that when starting a new gesture, previously started gestures will be aborted. When enabling the mode mixer on the other hand, the priority levels determine for every separate chain which calculation principle has to be used for the current iteration; that of the block mode or of the end-effector mode, and thus,

which constraints are loaded for the different chains: orientational information for every block composing the chain, or the desired end-effector position for the complete chain. Therefore, when gestures with different priority levels are selected with the mode mixer enabled, the imposed end-effector conditions originating from the different gestures result in a blended end posture. More information on this can be found in our previous publication [15].

3 Design 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. An additional degree of freedom indeed introduces additional parts and a motor. Furthermore, this extra weight needs to be carried by the higher positioned motors in the arm, which can result in the need of more powerful, heavier and more expensive motors. In addition, a wide joint range can increase the robot's range of motion, but can require a gearbox, which again introduces extra weight, and thus the use of more expensive motors. It is therefore worthwhile to investigate the benefits and drawbacks of introducing extra complexity in the design. With our proposed design methodology, we provide a tool to give insights in the influence of different design aspects, and to make substantiated trade-off's in the design process.

The methodology is based on the use of our developed gesture method, which, thanks to its generic framework, allows a fast and easy calculation of gestures for different morphologies. The effect of the placement of individual joints on a series of postures can be investigated by generating gestures for different configurations and visualizing them on a single virtual model. 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 in, for example, a design using servomotors, whereof the range is limited. Using the method, an optimal placement of the servo's neutral point can be determined, which can in some cases cancel the need of a 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 20 DOF actuated head [25,26]. Now, in a second iteration, we aim to build a new robot, extended with actuated arms. The virtual model of the robot's character was constructed in
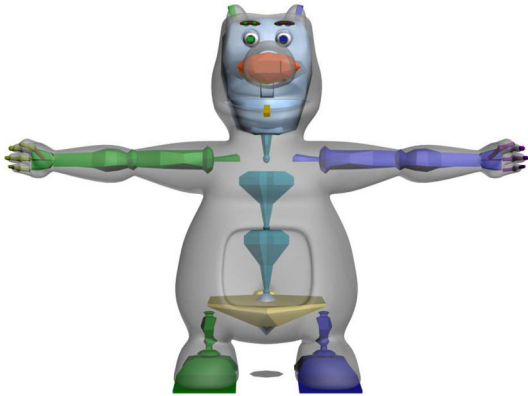
Fig. 3 Virtual model of the Probo character, constructed using the 3DS MAX Biped

Autodesk 3DS MAX and rigged as a 3DS MAX Biped (see Fig. 3). The Biped is a customizable skeleton consisting of linked bones, following the human anatomy.

Different joint configurations can be assigned to the virtual model, by associating 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 and offer it 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 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 [27]. Table 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. 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 method 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 and the Biped's specifications provides the necessary rotation axes. The mapping for the joint configuration discussed above is illustrated in Table 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.

3.1 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 morphology, the methodology discussed above was followed for a series of configurations. Table 3 shows a set of results for 7 different 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 3. From left to right, the end posture, and so 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 is seen as a *complete* configuration and therefore, the calculated end postures can serve as a reference for comparing 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
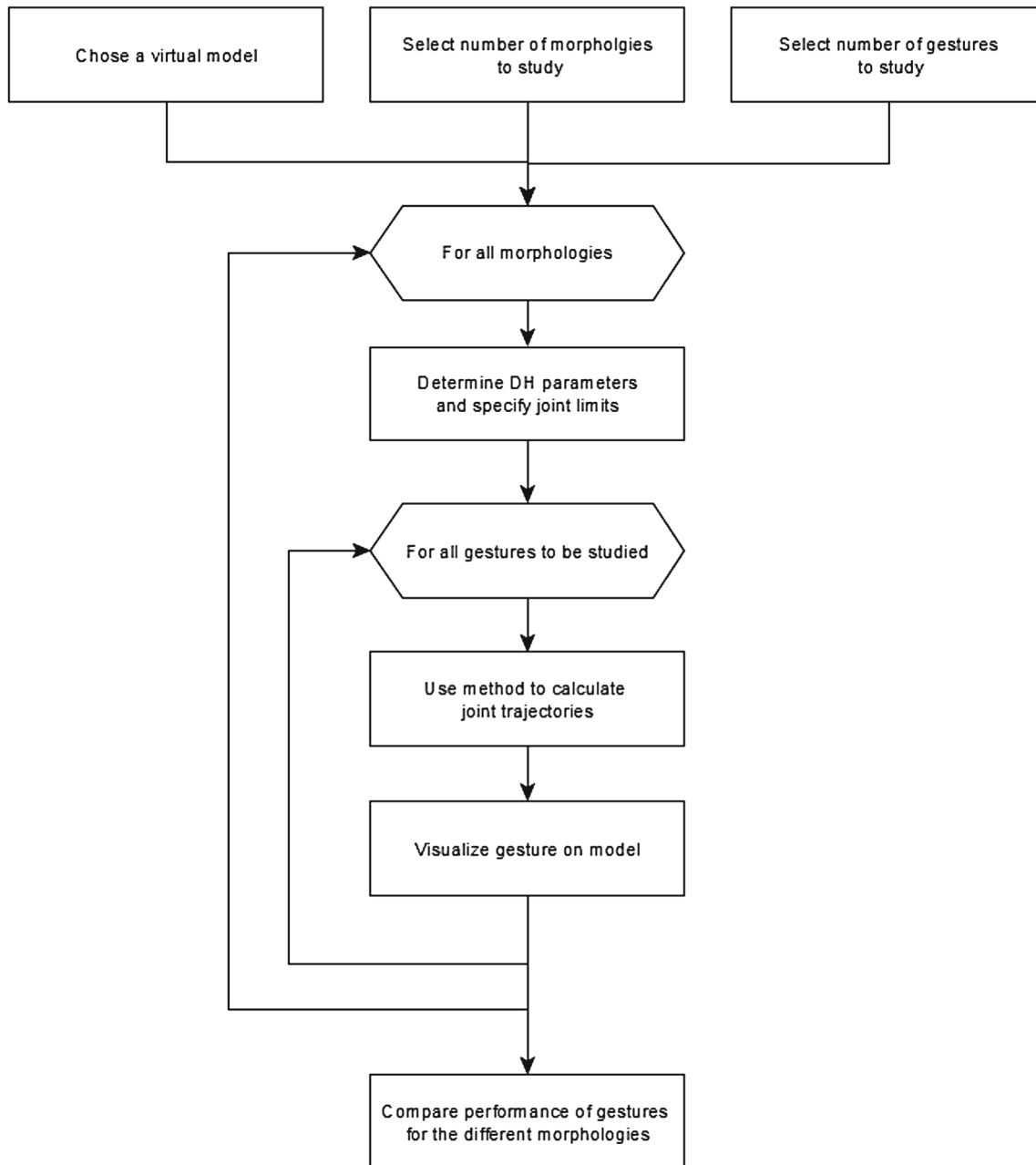
Fig. 4 Flowchart visualizing the process of the proposed design methodology. The process starts by selecting a virtual model to visualize the output, and a number of morphologies and gestures to study. For every morphology, the gestures are calculated using the gesture software and visualized on the virtual model, whereafter they can be compared to eventually select an optimal morphology regarding complexity and expressivity

6, where only 1 joint is present in the wrist. Configuration 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 which 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 adduction/abduction, 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, depending on the designer's goals and resources, this perfor-

Table 1 In a first step, the configuration's Denavit–Hartenberg (DH) parameters are calculated. This table 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 to top, arm joints ranked from right to left). The remaining columns list the corresponding Denavit–Hartenberg parameters
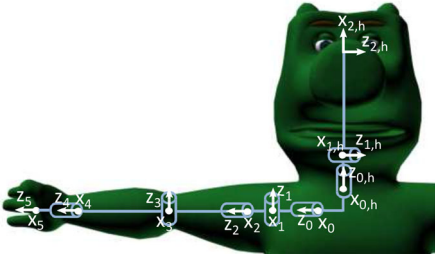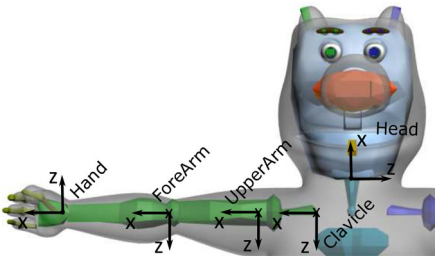
Configuration	Joint	DH-parameters			
		α (°)	a (cm)	d (cm)	θ (°)
	Head 1	-90	0	0	0
	Head 2	0	18.8	0	-90
	Shoulder 1	-90	0	0	0
	Shoulder 2	90	0	0	0
	Shoulder 3	-90	0	17.7	0
	Elbow	90	0	0	0
	Wrist 1	0	0	24.5	0

Table 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
	Shoulder 1	UpperArm	x
	Shoulder 2	UpperArm	$-z$
	Shoulder 3	UpperArm	x
	Elbow	ForeArm	$-z$
	Wrist 1	Hand	x

mance may be sufficient. This configuration can for example be interesting for expressing 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 limited workspace. 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 [28]), or a route direction-giving robot, a careful consideration of different morphologies can help finding an optimal design.

3.2 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 4 and Fig. 5 illustrate the use of the gesture method for this application, for configuration 6, shown in Table 3, considering the same gestures as above, namely the emotional expression for anger, disgust, happiness and sadness. The first row of Table 4, set (a), shows the end postures calculated using a wide joint limit range, namely $-180°$ to $180°$ for all joints, except for the elbow joint, which only goes to $0°$. These joint limits are represented by dots on the graphs shown in Fig. 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

Table 3 End postures for different emotional expressions, calculated for 7 different joint configurations
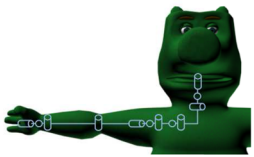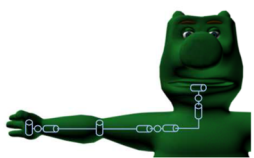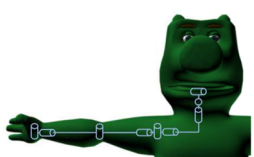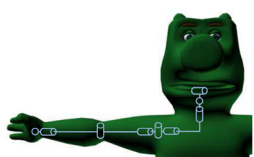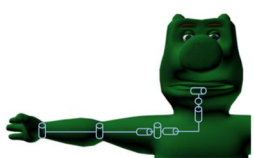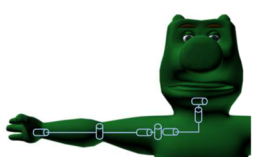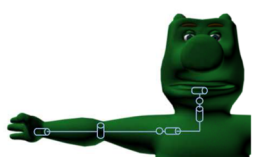
Joint configuration	Anger	Disgust	Happiness	Sadness

Table 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 3, using different sets of joint angle limits

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

Fig. 5 Calculated joint angles for the expression of anger, disgust, happiness and fear for configuration 6, visualized in Table 3, for different sets of joint angle limits

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 for $-100°$ to $20°$, constraining the initial value of $30°$ for the expression of anger and $-133°$ for that of disgust. The new limits, together with the corresponding resulting joint angles are visualized in the second row of Fig. 5. The effect of the new constraints can be noted from the difference in placement of the forearm in the second row of Table 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 however 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 $0°$ to $120°$, instead of to $150°$ 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 4, and the corresponding calculated angles are visualized in Fig. 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, however, 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 can be 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.

4 Conclusion

This paper proposes the use of a generic gesture method as a novel tool in the design of social robots. The framework of our developed gesture method is constructed very generally, and is only evaluated at runtime using a minimal set of morphological information specified by the user. Therefore, gestures can be created very fast for a desired configuration, which makes it interesting for design purposes as well. For robots designed to achieve predefined tasks, a number of essenti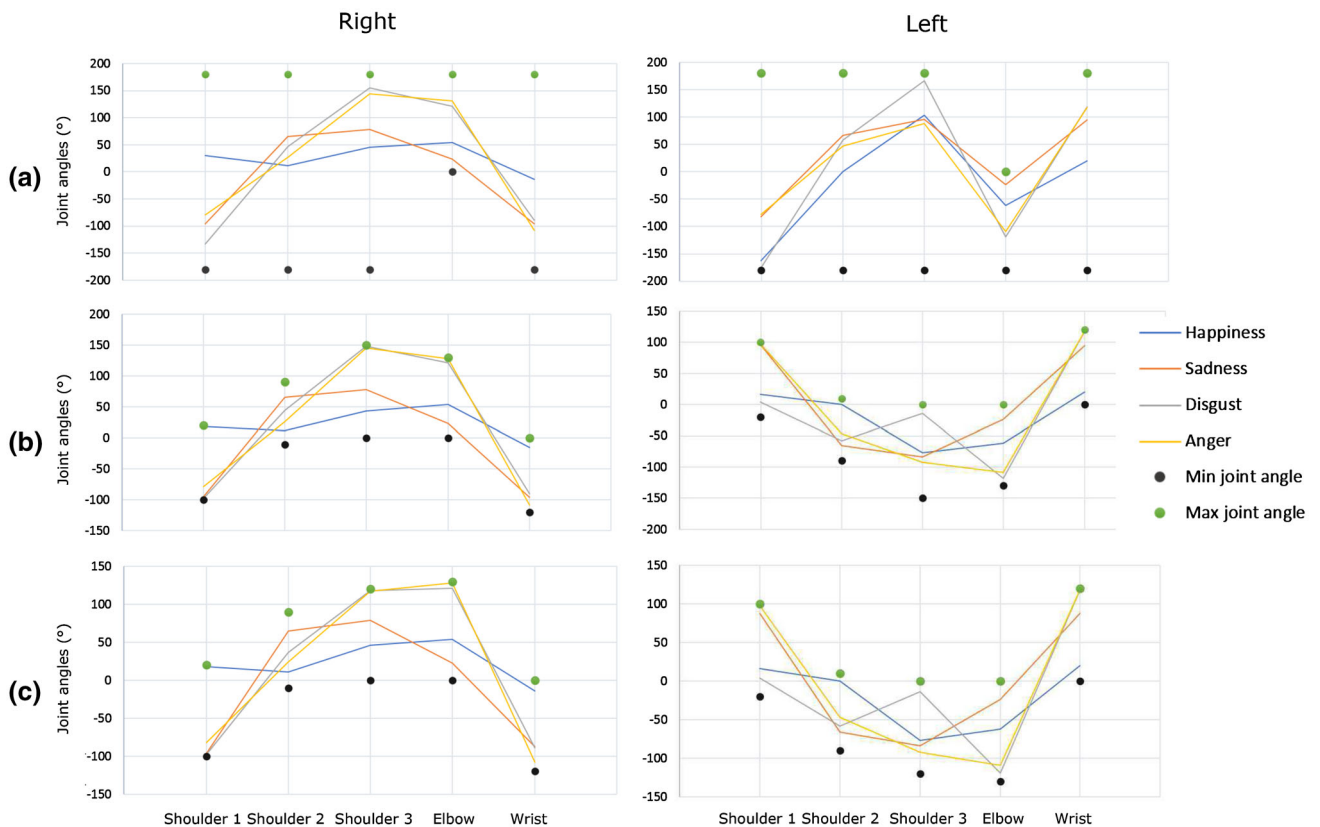al gestures can be nominated in advance. Generating these gestures for different morphologies can give interesting insights in the necessary design complexity for the desired task and can be useful to make a correct trade off in cost, simplicity and performance. An example of the study of both the influence of the exact configuration, i.e. the collocation of the different joints, and the joint angle range was illustrated in this paper 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.

Compliance with Ethical Standards

References

1. Breazeal C, Takanishi A, Kobayashi T (2008) Social robots that interact with people. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 1349–1369
2. Hirai K, Hirose M, Haikawa Y, Takenaka T (1998) The development of Honda humanoid robot. In: IEEE international conference on robotics and automation (ICRA 1998), vol 2, pp 1321–1326
3. Albers A, Brudniok S, Ottnad J, Sauter C, Sedchaicharn K (2006) Upper body of a new humanoid robot: the design of ARMAR III. In: 2006 6th IEEE-RAS international conference on humanoid robots. IEEE, pp 308–313
4. Ogura Y, Aikawa H, Shimomura K, Kondo H, Morishima A, Lim H, Takanishi A (2006) Development of a new humanoid robot WABIAN-2. In: IEEE international conference on robotics and automation (ICRA 2006), pp 76– 81
5. Tsagarakis NG, Metta G, Sandini G, Vernon D, Beira R, Becchi F, Righetti L, Santos-Victor J, Ijspeert AJ, Carrozza MC et al (2007) iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research. Adv Robot 21(10):1151–1175
6. Stasse O, Flayols T, Budhiraja R, Giraud-Esclasse K, Carpentier J, Del Prete A, Soueres P, Mansard N, Lamiraux F, Laumond JP et al (2017) TALOS: a new humanoid research platform targeted for industrial applications
7. Itoh K, Miwa H, Matsumoto M, Zecca M, Takanobu H, Roccella S, Carrozza M, Dario P, Takanishi, A (2004) Various emotional expressions with emotion expression humanoid robot WE-4RII. In: IEEE technical exhibition based conference on robotics and automation, pp 35–36
8. Ishiguro H, Ono T, Imai M, Maeda T, Kanda T, Nakatsu R (2001) Robovie: an interactive humanoid robot. Ind Robot Int J 28(6):498–504
9. Park IW, Kim JY, Lee J, Oh JH (2005) Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO). In: 2005 5th IEEE-RAS international conference on humanoid robots. IEEE, pp 321–326
10. Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J, Maisonnier B (2009) Mechatronic design of NAO humanoid. In: IEEE international conference on robotics and automation, 2009. ICRA'09. IEEE, pp 769–774
11. Parmiggiani A, Luca F, Scalzo A, Sureshbabu AV, Randazzo M, Maggiali M, Pattacini U, Lehmann H, Tikhanoff V, Domenichelli D, Cardellino A, Congiu P, Pagnin A, Cingolani R, Natale L, Metta G (2017) The design and validation of the R1 personal humanoid. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)

12. Asfour T, Schill J, Peters H, Klas C, Bücker J, Sander C, Schulz S, Kargov A, Werner T, Bartenbach V (2013) Armar-4: a 63 DOF torque controlled humanoid robot. In: 2013 13th IEEE-RAS international conference on humanoid robots (humanoids). IEEE, pp 390–396

13. Van de Perre G, Van Damme M, Lefeber D, Vanderborght B (2015) Development of a generic method to generate upper-body emotional expressions for different social robots. Adv Robot 29(9):59–609

14. Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D, Vanderborght B (2016) Reaching and pointing gestures calculated by a generic gesture system for social robots. Robot Auton Syst 83:32–43

15. Van de Perre G, De Beir A, Cao HL, Esteban PG, Lefeber D, Vanderborght B (2017) Generic method for generating blended gestures and mood expressions for social robots. Auton Robots 42:569–580

16. Kadaba MP, Ramakrishnan H, Wootten M (1990) Measurement of lower extremity kinematics during level walking. J Orthop Res 8(3):383–392

17. Sciavicco L (2009) Robotics: modelling, planning and control. Springer, Berlin

18. Drexler DA, Harmati I (2012) Joint constrained differential inverse kinematics algorithm for serial manipulators. Period Polytech Electr Eng Comput Sci 56(4):95

19. Klein CA, Blaho BE (1987) Dexterity measures for the design and control of kinematically redundant manipulators. Int J Robot Res 6(2):72–83

20. Xu J, Broekens J, Hindriks K, Neerincx MA (2013) The relative importance and interrelations between behavior parameters for robots' mood expression. In: 2013 humaine association conference on affective computing and intelligent interaction (ACII). IEEE, pp 558–563

21. Amaya K, Bruderlin A, Calvert T (1996) Emotion from motion. In: Graphics interface, vol 96. Toronto, pp 222–229

22. Pelachaud C (2009) Studies on gesture expressivity for a virtual agent. Speech Commun 51(7):630–639

23. Yamaguchi A, Yano Y, Doki S, Okuma S (2006) A study of emotional motion description by motion modification and adjectival expressions. In: IEEE conference on cybernetics and intelligent systems 2006, pp 1–6

24. Lin YH, Liu CY, Lee HW, Huang SL, Li TY (2009) Evaluating emotive character animations created with procedural animation. In: Ruttkay Z, Kipp M, Nijholt A, Vilhjálmsson HH (eds) Intelligent virtual agents. Springer, Berlin, pp 308–315

25. Goris K, Saldien J, Vanderborght B, Lefeber D (2011) Mechanical design of the huggable robot probo. Int J Humanoid Robot 8(3):481–511

26. Saldien J, Goris K, Vanderborght B, Vanderfaeilli J, Lefeber D (2010) Expressing emotions with the huggable robot probo. Int J Soci Robot Spec Issue Soc Accept HRI 2(4):377–389

27. Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer, New York

28. Jeong S, Santos KD, Graca S, O'Connell B, Anderson L, Stenquist N, Fitzpatrick K, Goodenough H, Logan D, Weinstock P, et al (2015) Designing a socially assistive robot for pediatric care. In: Proceedings of the 14th international conference on interaction design and children. ACM, pp 387–390

29. Kleinsmith A, De Silva PR, Bianchi-Berthouze N (2006) Cross-cultural differences in recognizing affect from body posture. Interact Comput 18(6):1371–1389

Greet Van de Perre received a Ph.D. degree in Engineering from the Vrije Universiteit Brussel (Belgium) in March 2018. Her Ph.D. research focused on the development of a generic method to generate gestures for social robots and the design of an actuated upper-torso for the social robot Elvis. Other research interests include collaborative and assistive robotics.

Albert De Beir received his Master degree in electromechanical engineering from the Vrije Universiteit Brussel and his Master degree in neuropsychology from the Universite Libre de Bruxelles. Currently, he is pursuing a Ph.D. at the Multibody Mechanics and Robotics research group (VUB). His research interest focus on the development of artificial intelligence for social robots.

Hoang-Long Cao received his B.Eng. in Mechatronics from Can Tho University (Vietnam) in 2010, and M.Sc. in Electromechanical engineering from the Vrije Universiteit Brussel (Belgium) and Universite Libre de Bruxelles (Belgium) in 2013. From September 2013, he has been working as a Ph.D. researcher at the Multibody Mechanics and Robotics research group (VUB). His research interest is social robotics. Currently he is involved in DREAM project for the development of robot-enhanced therapy for children with autism spectrum disorders.

Pablo Gómez Esteban received, in March 2014 from the Rey Juan Carlos University of Madrid, a Ph.D. degree for his thesis entitled Cooperation and Competition in Emotional Robot Societies with highest distinction. He is currently a post-doctoral researcher working at Vrije Universiteit Brussel in Belgium. His research was addressed towards decision making processes within competitive and cooperative environments; the impact that affective factors have in such processes; and designing and implementing those models to be used in non-expensive platforms within domestic environments. But his research interests also lie on cognitive architectures for social robots and human–robot interaction.

Dirk Lefeber was born in 1956. He received a degree in Civil Engineering in 1979 and a Ph.D. in Applied Sciences in 1986 from the Vrije Universiteit Brussel. Currently he is full professor, former head of the department of Mechanical Engineering and head of the Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, which he founded in 1990. The research interests of the group are new actuators with adaptable compliance, dynamically balanced robots, robot assistants, rehabilitation robotics and multibody dynamics, all in the context of physical and cognitive human robot interaction with emphasis on improving the quality of life of human.

Bram Vanderborght focused his Ph.D. research on the use of adaptable compliance of pneumatic artificial muscles in the dynamically balanced biped Lucy. In May–June 2006 he performed research on the humanoids robot HRP-2 at the Joint Japanese/French Robotics Laboratory (JRL) in AIST, Tsukuba (Japan) in the research Dynamically stepping over large obstacles by the humanoid robot HRP-2". From October 2007 to April 2010 he worked as post-doc researcher at the Italian Institute of Technology in Genova (Italy) on the humanoid robot iCub and compliant actuation. Since October 2009 he is appointed as professor at the VUB. He has an ERC starting grant on SPEA actuation concept. He is member of the Young Academy of the Royal Flemish Academy of Belgium for Science and the Arts. His research interests include cognitive and physical human robot interaction, social robots, humanoids, robot assisted therapy and rehabilitation/assistive robotics.

DualKeepon: A Human-Robot Interaction Testbed to Study Linguistic Features of Speech

Abstract In this paper we present a novel dual robot system, ideal for carrying out pair-wise comparisons of linguistic features of speech in human-robot interactions. Our solution, using a modified version of the My-Keepon robotic toy developed by Beatbox, is a portable open-source system for researchers to set up experiments quickly, and in an intuitive way. We provide an online tutorial with all required materials to replicate the system. We present human-robot interaction experiments to verify the usability of the system. In one study we investigate the perception of robots using filled pauses, also known as hesitation markers. In a second study we investigate how social roles, realized by different prosodic and lexical speaking profiles, affect trust. Results show that the proposed system is a helpful tool for linguistic studies. In addition to the basic setup, advanced users of the system have the ability to connect the system to different robot platforms such as the NAO, Pepper and others.

Keywords keepon · social robot · human-robot interaction · nao · low-cost robotics

1 Introduction

An often used method in Human-Robot Interaction (HRI) studies rely on a between-subject experiment design, meaning that each participant is subjected to only one robot in one experimental condition. Effects of manipulations are then measured between participants. However, in some cases it might be relevant to compare a manipulation directly. Thereby, analyses do not need to rely solely on post-experiment questionnaires, but

Address(es) of author(s) should be given

can draw on in-the-moment responses as analytical resources. This can be difficult to achieve if only one robot is available; participants can only experience one first impression. Furthermore, carrying out experiments on large robot platforms are often made difficult by complex architectures.

In an attempt to make an affordable solution available to the research community we present in this paper a portable open-source system using two MyKeepon robots, hence called *DualKeepon*. One of the advantages of DualKeepon is that it is cheap to construct, can be easily adapted to run different experiments and is easy to use. Thus, it can work as a testbed, through which researchers can test hypotheses on a small scale before implementing new behaviors on more complex systems. Although, the system can be used with several off-the-shelves robots (see Subsect. 3.3) we demonstrate the systems' capabilities using the MyKeepon robotic toy developed by Beatbox, and the NAO, developed by Softbank Robotics. From previous work, a typical My-Keepon robotic toy can be converted into a low-cost controllable robotic platform with a small modification on the control board[1]. This 'hack' has been carried out and used in many studies by different research groups [6, 49, 15, 40, 16]. Therefore, MyKeepon is a suitable candidate with which to design a robotic system to study various aspects of human-robot interaction. A configuration using just one robot is suitable for between-subject experiments, while the DualKeepon system can expand the usability of MyKeepon to work with within-subject research designs. The interaction scenario of the DualKeepon can be configured to use images, sound clips, as well as the robots' voices as stimuli. We provide

---

[1] The 'Hacking Keepon' workshop at the 2013 International Summer School on Social HRI for a multidisciplinary group of researchers.

an online tutorial on Github (link hidden - blink review) with required materials to replicate the system which consists of: (a) hardware setup with a PCB design, Arduino firmware, and how to connect system elements; (b) a software to setup and configure the interaction scenario and run the experiment. To verify the applicability of the system, we present two HRI experiments using the system. In brief, the contribution of this paper is to provide researchers, especially non-technical ones, a portable and easy-to-use tool with which to carry out various types of human-robot interaction experiments and studies.

Although, the system can be implemented to study multiple aspects of human-robot interaction, we will for this paper describe how the system can be used to investigate the effect of linguistic features of speech in interaction with robots. This paper is organized as follows. Sect. 2 describes the previous work. We present the proposed system in Sect. 3 including the hardware, software, and the procedure to set up an experiment. To verify the system, we present an HRI experiments using the setup in Sect. 4. Finally, we present our conclusion and future work in Sect. 5.

2 Previous Work

2.1 Current Methods in HRI

Methodically, HRI draws on inspiration from psychology, in which new insight are found by carrying out controlled experiments with clearly defined conditions. This methodology necessitates an iterative design process in which new features are implemented gradually based on careful experimentation. However, this process can for some complex robot architectures be difficult, costly and timely. A common and well-adopted solution is to conduct experiment using the Wizard-of-Oz (WoZ) methodology [18]. However, it is argued that the ecological validity of studies carried out under this paradigm is in question, as there is a risk that participants interact through rather than with the robot under investigation [43].

An alternative method is to use video recordings of robots. This method of investigation is increasingly taken into use with the rise of platforms such as Mechanical Turk, CrowdFlower, and Prolific Academic. The idea is that participants rate an interaction with a robot based on a video. That is, researchers make video recordings of interactions with robots and distribute these video with a questionnaire on on-line platforms (such as Mechanical Turk, etc.) or on the social media. For example, using this method Torrey et al. [53] investigate how robots can give advice to humans using hedges and discourse

markers, and in a related study Srinivasan and Takayama [47] study politeness principles in human-robot interactions. Aarestrup et al. [3] investigat how intonation contours in robot greetings influence how people perceive it on factors such as friendliness, assertive and engagement. However, there are potential problems using this method. Although some research have found only small differences from using this method over live experimentation [52], others have found that there are significant a differences between observing and participating in interaction [48].

A different method upon which our own system draws inspiration from is demonstrated by Andrist et al. [4], who set up an experiment in which they wanted to study the effects of rhetorical ability and expertise knowledge on a robot's persuasive ability in a tour guide scenario. They conducted this study as well as a follow-up study two years later [5] using a setup with two Lego Mindstorm robots and a visual display in which participants make selections based on the verbal input they get from the robots. However, the authors did not describe exactly how the setup works on the technical side, meaning that anyone wanting to validate their results, or run similar studies would have to build a new system from the ground up.

2.2 Current Methods in the Study of Linguistic Features of Speech

A significant challenge of studying linguistic and paralinguistic speech features is that the perception of an utterance is shaped by various co-occurring semantic and non-semantic characteristics of the speech, as well as nonverbal behaviors [21]. Thus a certain speech feature must be isolated from the others for its effects to be examined. In eliciting linguistic attitudes, researchers traditionally employ trained speakers in so called 'matched guise technique' scenarios [32,25], where subjects evaluate speakers based on multiple utterances read by a single speaker, without knowing that the same speaker is being evaluated [27]. While this method controls the voice qualities by using a single [14], or a few speakers [10] it depends on the availability and quality of the speakers, which might inadvertently bias participants in favor of the experimental hypothesis [31,46].

In place of the natural speech produced by a human confederate within the matched guise technique, synthesized speech has been used extensively as an alternative in perceptual experiments [42], with steadily improved quality over the past decade. Despite of the limited intelligibility and naturalness of a human speaker, the synthetic utterance significantly reduces

the cost and time associated with recruiting a speaker [7]. It also provides consistent conditions for easily generating linguistic output at any time and allows for endless repetition [26]. Nevertheless, as one of the linguistic features of speech, nonverbal modalities play an important role in interactions [56,17]. The lack of embodiment and interactivity using synthesized voice alone is therefore liable to affect subjects perceptions [19], especially in interactions of a more social nature e.g. getting acquainted or in negotiation [54]. In addition, levels of embodiment have been shown to have an impact on the degree to which people regard a non-human agent as a serious conversation partner [22,8].

Consequently, the need of an embodied agent that is not a human confederate make social robots an ideal choice. With robots' varied levels of embodiment, manipulated utterances can be played in interactive contexts in a repeatable and reliable manner. Robots are able to provide embodied interactions with social cues [50,38], while the complexity of interaction can be controlled [44]. Robots can be used in a controlled way so that only desired information is presented, and variations can be made in an informed manner [50]. In addition to its predictable and consistent behaviors, using robotic systems allows all the experimental data to be collected for further analysis, either directly during or after the interaction [21,41].

2.3 Low-cost Robotics

In recent years we have seen several low-cost robotic systems such as the RoboSapien [51], the EZ-Robot [1], the MyKeepon [30], and the Pleo [2] emerge. However, while they are potentially useful for both HRI and linguistic research, they are produced and sold with the private consumer in mind. Therefore, it takes quite an effort to get these products in a state where they can be useful for research purposes. In addition, the efforts required are usually not well documented. This means that these technologies remain largely unavailable to for example non-technical researchers, such as linguists.

Thus, what is currently missing is a low-cost system that is easy to setup, even by non-engineers. Such a system will enable also linguists to independently carry out research on speech characteristics on embodied autonomous agents. Among the existing simple platforms, MyKeepon is demonstrated to be a constructive open-source platform accommodating a wide variety of human-robot interaction studies [12,6,49,15,40,16]. In the following we will present the manipulations we have carried on retail versions of the MyKeepon, and document every step of the process.
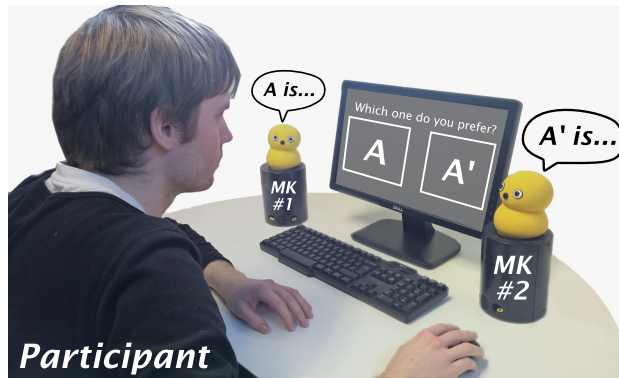

Fig. 1 The *DualKeepon* system includes two MyKeepon robots and a series of questions with images displayed on a computer screen.

3 The DualKeepon System

In this section, we present a portable and open-source setup to study linguistic features of speech. We provide a tutorial containing all required material to replicate the setup at includes a screen and two MyKeepon robotic toys with two different characteristics of a design object, as illustrated in Fig. 1. This object can be, but not limited to, speech pitch, range, intensity, and speed, socio-pragmatic communication strategies as well as various features of movement. A participant sits in a chair and faces two robots, which are placed on both sides of a computer screen. The setup is designed with a pairwise comparison between two pictures – a method that has been utilized in other human-robot interaction studies e.g. [13,4,5]. Specifically, the interaction consists of a number of questions, each with a pair of pictures displayed on a screen for the participant to select. Each robot describes the content of the picture on its side (e.g. a place, a building, an object) and asks the user which of these they would prefer. The users' selections are recorded for further analysis. Post-interaction questionnaires can be asked for self-evaluation purpose. Alternatively, the system can also be set up to work without visual stimuli. Since the two robots are identical, participants' selections are dependent on the behavioral stimuli under investigation. One of the major advantages of the system is it its ease of use, which is demonstrated in the validation section of the paper.

3.1 Hardware

The hardware of the *DualKeepon* system includes a computer, a control board with two Arduino boards and an audio amplifier, and two MyKeepon robotic
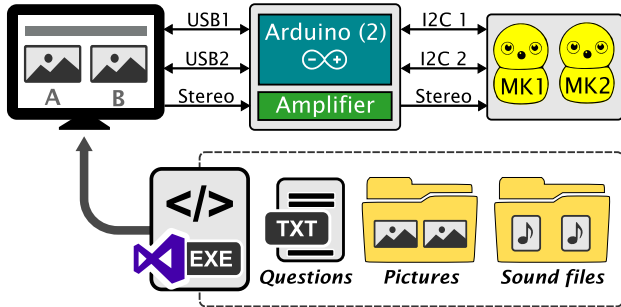
Fig. 2 The system components: a computer, a Visual C# software, a control board, and two MyKeepon robots.
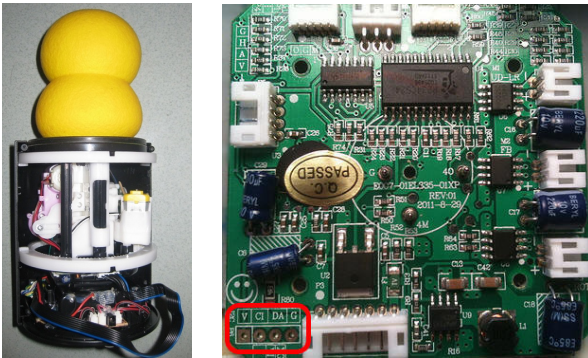

Fig. 3 The internal PCB of the robot with an I2C bus allows developers to enhance the robot's functions.

toys. Fig. 2 describes how these components are connected. Each MyKeepon is controlled by an Arduino board which is connected to the computer via a USB cable. Arduino boards receive movement commands from the computer. These commands are converted and sent to the robots via I2C buses to the internal PCB of the robots, which is accessible by opening the casting (Fig. 3).

The sound (speech) of the two robots is also sent from the computer via a stereo jack. The sound is amplified before being sent to the robots' internal speakers, but can alternatively also be sent to a set of external speakers. Audio is played back from a wave file, which can be generated using any text-to-speech engine or by self-recording speech or audio.

We provide the control board schematic which can be applicable to different Arduino models. We also provide a PCB layout designed in Eagle, which uses Arduino Nano boards, a PAM8403 amplifier board, and a 3.5 mm audio jack adapter as can be seen in Fig. 4.

3.2 Software

We developed an easy-to-use software to display the content of the interaction scenario of an experiment. First, the experimenter establishes serial connections
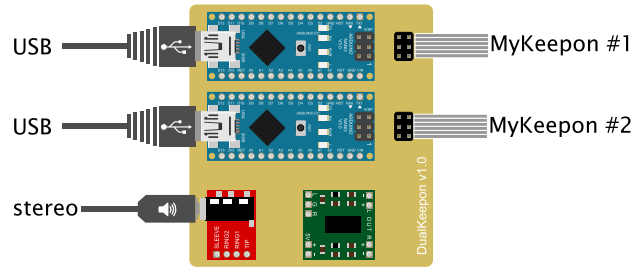
Fig. 4 The control board using two Arduino Nano boards, a PAM8403 amplifier board, and a 3.5 mm audio jack adapter. The two IDC connectors are connected to the internal PCBs of the robots.
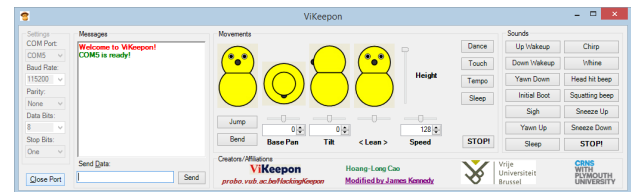
Fig. 5 Connection setup GUI.

(a) Question

(b) After making a selection
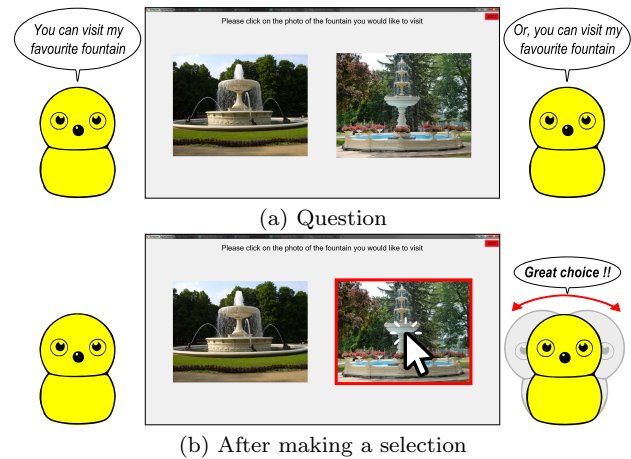
Fig. 7 An example of a question in which a participant has to select his/her preference between two pictures described by two robots with two different voices.

with two robots through a GUI (Fig. 5). This GUI also allows the experiments to control off-script actions if necessary e.g. jump, bend, pan, tilt, lean.

Second, the software loads questions, images, and sound files from a database as the resources of the interaction scenario (Fig. 2). The user information is then entered. The human-robot interaction begins when a welcome screen appears. The software then shows pairs of pictures one after another e.g. question about two fountains described by two robots with different voices (Fig. 7a). When the participant makes selections, the robots give feedback by sound and movement e.g. 'Great choice!' accompanied by a slight jump (Fig. 7b).
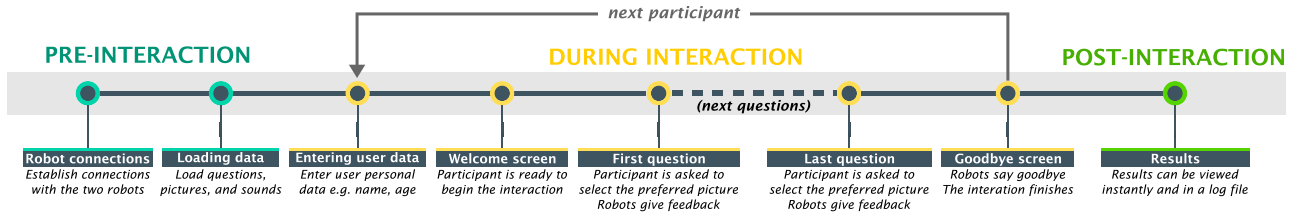
Fig. 6 The procedure to set up and run an experiment using the *DualKeepon* system.

Fig. 8 The advanced version of the software can be applied to other robot platforms e.g. NAO, Pepper.

Timing and design of movement and audio behavior can be customized. After the interaction finishes, the result is shown on the screen as well as recorded in a log file for further analysis. The whole procedure to set up and run the experiment is illustrated by Fig. 6.

The Visual C# software can work with an unlimited number of questions and various types and sizes of image and sound formats. The code is open-source and can be adaptable to different types of studies.
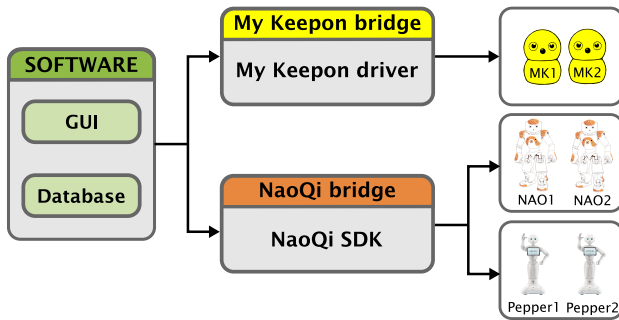
3.3 Developer Version for Other Robot Platforms

Although MyKeepon is the main robot platform used in the project, we also provide an advanced version of this software allowing developers to apply this setup to other robot platforms. Since different robots use various types of software development kits (SDKs) and means of communication (e.g. serial, wifi), the software controlling the GUI and database is made independent with the low-level robot control. The communication between between them is handled via a bridge.

At the moment, bridges for MyKeepon and NAO/Pepper are implemented (Fig. 8), which can be downloaded on our GitHub project[2]. The MyKeepon bridge is used to setup the serial communication (COM ports) with the Arduino boards controlling the MyKeepon robots. The NAO/Pepper bridge is used to setup the wifi communication with NAO/Pepper

robots by NAOqi SDK by SoftBank Robotics. We also provide a library containing various types of pre-programmed behaviors (e.g. body gestures while talking) for NAO/Pepper.

Similar bridges can be developed following the same principle. Developers can also possibly establish a mixed-robot interaction scenario. However, that is beyond the scope of this paper.

4 System Usability

We have conducted two HRI experiments to validate the system usability regarding the technical issues and the ability to be utilized to aspects of human-robot interaction. In this section, we present the two studies. One study investigates how hesitation markers in a robot's speech affect the robot's persuasive ability, and how charismatic they find the robot to be. In the second study, we investigate the relationship between robot voice with different social roles (tutor and peer) on trust and compliance.

4.1 Study 1: Effect of Hesitation Markers on Persuasive Ability in HRI

In this experiment, we studied the perception of robots using hesitation markers. The background of the study is that filled pauses, also referred to as hesitation markers, such as 'uhm' and 'uh' are generally seen as markers of disfluency. For example, Levinson argues that hesitation markers are features of dispreffered responses in face-to-face interaction [33], and Tree [24] argues that hesitation markers are associated with difficulty and dishonesty. However, recent work in pragmatics suggest that they may have a communicative and useful function as well, for example to manage turn-taking in interaction[20]. Recent work in HRI also suggest to use hesitation markers to when the robot needs more time to process a response [55], and that robots are rated as more engaging and as more polite when using hesitation markers [23]. To test whether hesitation markers can also have a persuasive effect we set up a study in which one robot used hesitation markers and one did

---

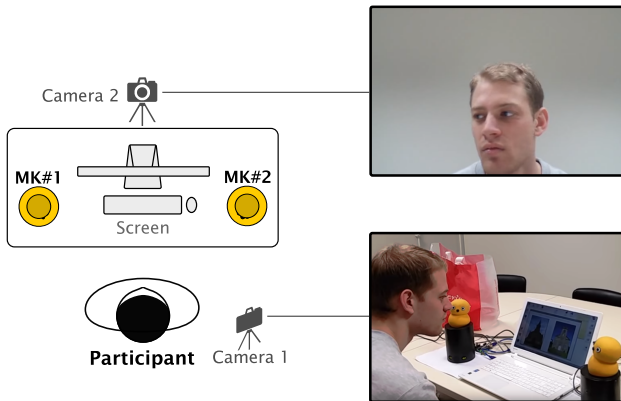[2] Bridge version: (link hidden - blink review)

Fig. 9 The experiment setup. A participant sits in front of a screen and two MyKeepon robots. Two cameras are used to record the participant's behavior.

not. To counter a potential ordering effect we varied the order in which the robots spoke. We recruited 32 people (11 men and 21 women), with an age range from 16 to 41 years ($M = 26.09, SD = 5.28$) from different backgrounds.

4.1.1 Method

Fig. 9 describes the experimental setup used in the experiment. A participant sits in front of a screen and two MyKeepon robots in a quiet and sufficiently enlightened room. The two robots make use of different speaking strategies. Two cameras are used to record the participant's behavior (front view and back view) during the interaction for further analysis.

Each participant interacted with two robots at the same in a scenario where they were asked to plan a tour of Paris based on suggestions made by the robots. Participants were shown two different but similarly-looking locations, and after each of the robots had given a short verbal description of a location, participants were asked to choose which of the locations they would like to visit. Participants were asked in total to choose 7 different locations. The interaction took from 7 to 10 minutes. At the end of the interaction, participants filled out post-experiement questionnaires.

During the experiment, the system worked without any significant technical errors. The communication between the robots and the database was stable. The system setup process was fast and easy (approx. 5 minutes), and participants had no difficulties interacting with the robots. However, we found that the sound quality of the internal speakers of MyKeepon is relatively poor. We therefore advise the use of external speakers in the vicinity of the robots or to replace the internal speakers with ones of better quality.

We recorded the answer for each question as an objective measure. At the end of the interaction, participants were asked to fill out a survey with seven questions about their impression of the robots (Enthusiastic, Charming, Persuasive, Boring, Passionate, Convincing, Engaging) using a 5-point Likert scale, where 1 is 'very' and 5 is 'not at all'. We also asked which robot he or she would prefer to interact again.

4.1.2 Results

We obtained the following results:

1. Participants say they would most prefer to speak to the non-hesitant robot (81% of the cases).
2. Participants follow the advice of the non-hesitant robot more often (90 % of the cases).
3. There is a positive Pearson correlation ($r = 0.37$, $p = .03$) between whose advice they follow and who they say they would want to talk to again.
4. We built a 'charisma' index of the 7 adjectives with an alpha value of 0.81. Participants rate the non-hesitant robot ($M = 2.76$, $SD = 0.63$) as significantly more charismatic compared to the hesitant robot ($M = 3.25$, $SD = 0.68$, $p = .004$).

The results we obtain show that hesitation markers are not perceived as more than markers of disfluency. This conforms with some perceptions of hesitations markers, but stand in contrast to work in linguistic pragmatics [20], and in HRI [55]. However, the frequency, synthesis or placement of the markers may have an influence on how they are perceived. Alternatively, although others have found positive effects of hesitations markers, these may not necessarily extent to a robot's persuasive capability.

4.2 Study 2: Trust and Robot Roles

For the second study, we adapted the system to be used with two NAO robots. This was achieved by using a NAOqi bridge between our software and the NAO hardware (see Subsect. 3.3).

The study reported on here is inspired by work carried out in second language teaching, in which students are found to distrust comments or recommendations from their peers in comparison to those from their teachers [37, 39, 35]. Language teaching, and learning contexts in general is one of the areas, in which robots are envisioned to make a contribution in the near future. For example robots are tested as tutors [34], teachers [11], teaching assistants [36], and peers [28]. Several studies report on the success of using robots as peer in
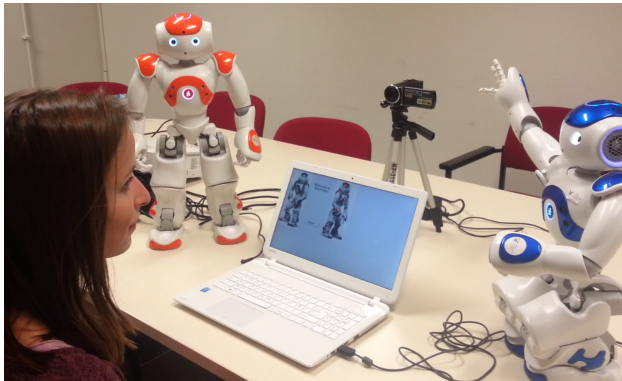
Fig. 10 The trust study experiment setup with two NAO robots in peer and tutor roles.

learning contexts (e.g. [29,9], however, only few compare the social roles directly (e.g. [57]). While many of these studies deal with important factors such as engagement, personalized behavior, and social roles, none of them directly deal with how social roles affect trust. In the work we present here, we investigate how the social roles 'peer' and 'tutor' affect trust and compliance.

4.2.1 Method

The physical setup of the experiment is identical with the setup in study 1, only the MyKeepon robots have been replaced with two NAO robots (Fig. 8). In this study, rather than planning a tour of Paris, participants are asked to answer a series of difficult multiple-choice questions, in which participant have either only very little or no prior knowledge at all. For each question each of the robots give their advice and recommendation about what to choose.

The characteristics of the two robots speech profiles (peer and tutor) draws on inspiration from Zaga et al. [57]. The robots differed in regard to both prosodic and lexical features of speech. Specifically, both robot voices were presynthesized using Acapela Box[3]. The 'peer' robot used the Elise voice while the 'tutor' robot used the Margaux voice. In addition, the 'tutor' robot used a more formal language, for example when addressing the participant. Finally, the tutor robot also had an academic vocabulary, whereas the 'peer' robot produced utterances in the vernacular. The study, including all stimuli, is carried out in French. After each interaction, participants answer a 14-item questionnaire to evaluate how much they trust each robot (11-point Likert scale from 0% to 100%) [45]. We also asked which robot he or she would trust more.

[3] https://acapela-box.com

4.2.2 Results

As with study 1, the system worked without any significant technical errors. The communication between the robots and the database was stable, the system setup process was fast and easy. We obtained the following results:

1. We built a trust index of the 14 questionnaire items. The result shows that the average trust score of the robot with a peer voice ($M = 59.21, SD = 8.24, \alpha = 0.84$) is significantly lower than the one with a tutor voice ($M = 64.02, SD = 7.09, \alpha = .82, p < .05$).
2. All participants subjectively say they would trust the tutor robot more.
3. Participants also follow the advice of the tutor robot more often (in 70 % of the questions).

The results we obtain show that the tutor robot is trusted more than the peer robot in the task-based scenario. This conforms to findings from research on trust in HHI, which finds that students trust their teachers more than they do their peers [37,39,35]. The results also show that the 'robot-as-peer' paradigm, which is widely used in robotic applications in learning contexts, is not the perfect approach for all scenarios. For example, robot designers can take advantage of the authority that comes with a social role such as 'tutor' or 'teacher' to influence users.

4.3 Summary of the Validation

Our two validation studies demonstrate the system's usability, reliability, and user-friendliness. The system is demonstrated to obtain the designed functionalities with MyKeepon robots, for which the system is originally constructed, as well as with the NAO robots.

5 Conclusion and Future Work

In this paper, we presented the *DualKeepon* system which is a portable system for studying human-robot interactions. The system is made open-source from hardware to software, which provides researchers possibilities for adapting the system to study different features of speech.

The system also works well as an entry-level platform for students interested in HRI. Currently, the system is made accessible to our own students in four different study programs, at both humanistic and technical faculties. The system has been used for data collections in several bachelor theses and term papers.

The system can be expanded by integrating more recording devices for more advanced analyses of human behavior, for example gaze. We evaluated the system by presenting two HRI experiments that focused on on linguistic factors, and how these factor affected users impressions and interactions with the system. In our further work, we will validate the system by conducting more HRI experiments on other robot platforms e.g. NAO, Pepper, Ono, EZ-Robot, etc.

References

1. EZ-Robot. http://www.ez-robot.com/. [Online; accessed 9-Feb-2017]
2. Pleo. http://www.pleoworld.com/. [Online; accessed 9-Feb-2017]
3. Aarestrup, M., Jensen, L.C., Fischer, K.: The sound makes the greeting: Interpersonal functions of intonation in human-robot interaction. In: 2015 AAAI Spring Symposium Series (2015)
4. Andrist, S., Spannan, E., Mutlu, B.: Rhetorical robots: making robots more effective speakers using linguistic cues of expertise. In: Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, pp. 341–348. IEEE Press (2013)
5. Andrist, S., Ziadee, M., Boukaram, H., Mutlu, B., Sakr, M.: Effects of culture on the credibility of robot speech: A comparison between english and arabic. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 157–164. ACM (2015)
6. Azmin, A.F., Shamsuddin, S., Yussof, H.: Hri observation with my keepon robot using kansei engineering approach. In: 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), pp. 1–6 (2016). DOI 10.1109/ROMA.2016.7847831
7. Azuma, J.: Creating" Micro" Exercises Utilizing TTS (Text-To-Speech) Technology: New Horizons in Foreign Language Teaching. na (2007)
8. Bainbridge, W.A., Hart, J.W., Kim, E.S., Scassellati, B.: The benefits of interactions with physically present robots over video-displayed agents. International Journal of Social Robotics **3**(1), 41–52 (2011)
9. Baxter, P., Ashurst, E., Read, R., Kennedy, J., Belpaeme, T.: Robot education peers in a situated primary school study: Personalisation promotes child learning. PloS one **12**(5), e0178,126 (2017)
10. Bresnahan, M.J., Ohashi, R., Nebashi, R., Liu, W.Y., Shearman, S.M.: Attitudinal and affective response toward accented english. Language & Communication **22**(2), 171–185 (2002)
11. Brown, L., Howard, A.M.: Engaging children in math education using a socially interactive humanoid robot. In: Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on, pp. 183–188. IEEE
12. Cao, H.L., Van de Perre, G., Simut, R., Pop, C., Peca, A., Lefeber, D., Vanderborght, B.: Enhancing my keepon robot: A simple and low-cost solution for robot platform in human-robot interaction studies. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, pp. 555–560. IEEE (2014)
13. Cao, H.L., Pop, C., Simut, R., Furnémont, R., De Beir, A., Van de Perre, G., Esteban, P.G., Lefeber, D., Vanderborght, B.: Probolino: A portable low-cost social device for home-based autism therapy. In: International Conference on Social Robotics, pp. 93–102. Springer (2015)
14. Clopper, C.G., Pisoni, D.B.: Perception of dialect variation: Some implications for current research and theory in speech perception. Research on Spoken Language Processing Progress Report (25), 2001–2002 (2002)
15. Costescu, C.A., Vanderborght, B., David, D.O.: Reversal learning task in children with autism spectrum disorder: A robot-based approach. J. Autism Dev. Disord. **45**(11), 3715–3725 (2015)
16. Costescu, C.A., Vanderborght, B., David, D.O.: Robot-enhanced cbt for dysfunctional emotions in social situations for children with asd. Journal of Evidence-Based Psychotherapies **17**(2), 119–132 (2017)
17. Docan-Morgan, T., Manusov, V., Harvey, J.: When a small thing means so much: Nonverbal cues as turning points in relationships. Interpersona **7**(1), 110 (2013)
18. Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J.D., Gandy, M.: Wizard of oz support throughout an iterative design process. IEEE Pervasive Computing **4**(4), 18–26 (2005)
19. Fasola, J., Mataric, M.: A socially assistive robot exercise coach for the elderly. Journal of Human-Robot Interaction **2**(2), 3–32 (2013)
20. Fischer, K.: From cognitive semantics to lexical pragmatics: The functional polysemy of discourse particles. Walter de Gruyter (2000)
21. Fischer, K.: Robots as confederates: How robots can and should support research in the humanities. In: Proceedings of the 2016 Robophilosophy Conference, Aarhus, Denmark (2016)
22. Fischer, K., Lohan, K., Foth, K.: Levels of embodiment: Linguistic analyses of factors influencing hri. In: Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on, pp. 463–470. IEEE (2012)
23. Fischer, K., Niebuhr, O., Novák-Tót, E., Jensen, L.C.: Strahlt die negative reputation von häsitationsmarkern auf ihre sprecher aus? In: Proc. 43rd Annual Meeting of the German Acoustical Society (DAGA), Kiel, Germany, pp. 1450–1453 (2017)
24. Fox Tree, J.E.: Interpreting pauses and ums at turn exchanges. Discourse Processes **34**(1), 37–55 (2002)
25. Gallois, C., Giles, H.: Communication accommodation theory. The international encyclopedia of language and social interaction (2015)
26. Henton, C.: Text-to-speech synthesis development. The Encyclopedia of Applied Linguistics (2012)
27. Holmes, J., Hazen, K.: Research methods in sociolinguistics: A practical guide. John Wiley & Sons (2013)
28. Hood, D., Lemaignan, S., Dillenbourg, P.: When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 83–90. ACM
29. Kanda, T., Hirano, T., Eaton, D., Ishiguro, H.: Interactive robots as social partners and peer tutors for children: A field trial. Human-Computer Interaction **19**(1/2), 61–84 (2004)
30. Kozima, H., Michalowski, M.P., Nakagawa, C.: Keepon. International Journal of Social Robotics **1**(1), 3–18 (2009)

31. Kuhlen, A.K., Brennan, S.E.: Language in dialogue: when confederates might be hazardous to your data. Psychonomic bulletin & review **20**(1), 54–72 (2013)
32. Lambert, W.E., Hodgson, R.C., Gardner, R.C., Fillenbaum, S.: Evaluational reactions to spoken languages. The Journal of Abnormal and Social Psychology **60**(1), 44 (1960)
33. Levinson, S.C.: Pragmatics. Cambridge University Press, New York, NY (1983)
34. Leyzberg, D., Spaulding, S., Scassellati, B.: Personalizing robot tutors to individuals' learning differences. In: Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, pp. 423–430. ACM (2014)
35. Mangelsdorf, K.: Peer reviews in the esl composition classroom: What do the students think? ELT journal **46**(3), 274–284 (1992)
36. Meghdari, A., Alemi, M., Ghazisaedy, M., Taheri, A., Karimian, A., Zandvakili, M.: Applying robots as teaching assistant in efl classes at iranian middle-schools. In: The 2013 International Conference on Education and Modern Educational Technologies
37. Mendonca, C.O., Johnson, K.E.: Peer review negotiations: Revision activities in esl writing instruction. TESOL quarterly **28**(4), 745–769 (1994)
38. Mohammad, Y., Ohya, T., Hiramatsu, T., Sumi, Y., Nishida, T.: Embodiment of knowledge into the interaction and physical domains using robots. In: Control, Automation and Systems, 2007. ICCAS'07. International Conference on, pp. 737–744. IEEE (2007)
39. Nelson, G.L., Murphy, J.M.: Peer response groups: Do l2 writers use peer comments in revising their drafts? TESOL quarterly **27**(1), 135–141 (1993)
40. Peca, A., Simut, R., Cao, H.L., Vanderborght, B.: Do infants perceive the social robot keepon as a communicative partner? Infant Behavior and Development **42**, 157–167 (2016)
41. Pfeifer, R., Gómez, G.: Morphological computation–connecting brain, body, and environment. In: Creating Brain-Like Intelligence, pp. 66–83. Springer (2009)
42. Pisoni, D., Remez, R.: The handbook of speech perception. John Wiley & Sons (2008)
43. Riek, L.D.: Wizard of oz studies in hri: a systematic review and new reporting guidelines. Journal of Human-Robot Interaction **1**(1) (2012)
44. Robins, B., Dautenhahn, K., Dubowski, J.: Robots as isolators or mediators for children with autism a cautionary tale. In: Procs of the AISB 05 Symposium on Robot Companions. AISB (2005)
45. Schaefer, K.: The perception and measurement of human-robot trust. Ph.D. thesis, University of Central Florida Orlando, Florida (2013)
46. Solís Obiols, M.: The matched guise technique: A critical approximation to a classic test for formal measurement of language attitudes. Noves SL. Revista de Sociolinguistica **1**, 1–6 (2002)
47. Srinivasan, V., Takayama, L.: Help me please: Robot politeness strategies for soliciting help from humans. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 4945–4955. ACM (2016)
48. Strait, M., Canning, C., Scheutz, M.: Let me tell you! investigating the effects of robot communication strategies in advice-giving situations based on robot appearance, interaction modality and distance. In: Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, pp. 479–486. ACM (2014)
49. Strupka, E., Niebuhr, O., Fischer, K.: Influence of robot gender and speaker gender on prosodic entrainment in hri. In: Interactive Session at the Ieee International Symposium on Robot and Human Interactive Communication (ro-man 2016), New York City (2016)
50. Thill, S., Pop, C.A., Belpaeme, T., Ziemke, T., Vanderborght, B.: Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. Paladyn **3**(4), 209–217 (2012)
51. Tilden, M.W.: Neuromorphic robot humanoid to step into the market. The Neuromorphic Engineer **1**(1), 12 (2004)
52. Toris, R., Kent, D., Chernova, S.: The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing. Journal of Human-Robot Interaction **3**(2), 25–49 (2014)
53. Torrey, C., Fussell, S., Kiesler, S.: How a robot should give advice. In: Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, pp. 275–282. IEEE Press (2013)
54. Whittaker, S., O'Conaill, B.: The role of vision in face-to-face and mediated communication. (1997)
55. Wigdor, N., de Greeff, J., Looije, R., Neerincx, M.A.: How to improve human-robot interaction with conversational fillers. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 219–224 (2016)
56. Wolfgang, A.: Nonverbal behavior: Applications and cultural implications. Academic Press (2013)
57. Zaga, C., Lohse, M., Truong, K.P., Evers, V.: The effect of a robot's social character on children's task engagement: Peer versus tutor. In: International Conference on Social Robotics, pp. 704–713. Springer (2015)