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: **1/4/2016**
Submission Date: **25/3/2016**

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: **2.3**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

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 for the first two years of the DREAM project. During these years the cognitive architecture and the Expression and Actuation subsystem have been designed. They are described within this report.

Principal Contributors

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

Paul Baxter, Plymouth University
Tony Belpaeme, Plymouth University
Hoang-Long Cao, Vrije Universiteit Brussel
Albert de Beir, Vrije Universiteit Brussel
Pablo Gómez, Vrije Universiteit Brussel
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.

1   Overview of WP6 architecture

In DREAM we will move away from Wizard of Oz-controlled behaviour for the robot, which too often is the de facto mode of interaction in Robot Assisted Therapy (RAT) [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 RAT in a supervised autonomous way. 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 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.
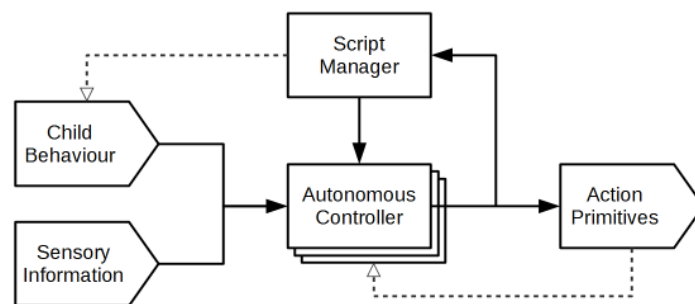

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied.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

The autonomous controller is composed of a number of 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 will be 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). In the cognitive controller architecture it is defined the control that the supervising therapist can exert over the behaviour of the robot (effectively

a limited 'remote control' functionality). This naturally has a number of operational consequences
for other subsystems of the cognitive controller, which will be handled through the oversight of the
Self-Monitoring subsystem (to prevent conflicting commands for example). As a result, we have for-
mulated the following architecture describing how cognitive control informed by the therapy scripts
is to be achieved (Figure 2), see Annex 3.5 for further details. 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. The next
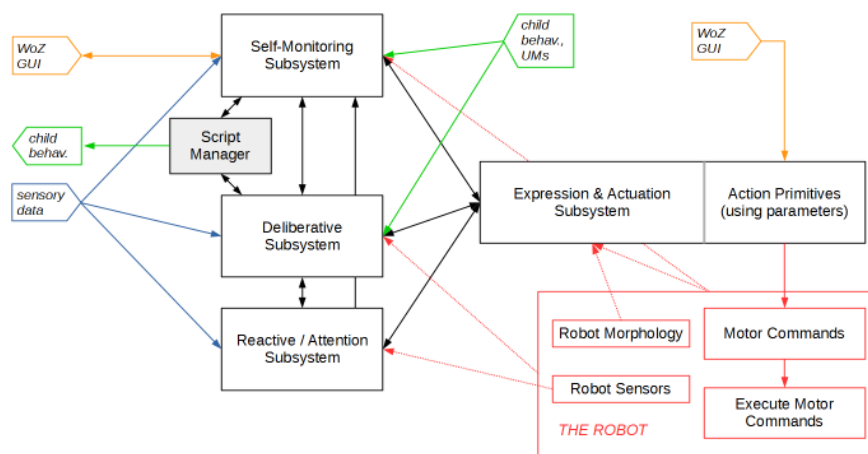version of this document will be ready for month 36.

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.

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, espe-
cially 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.
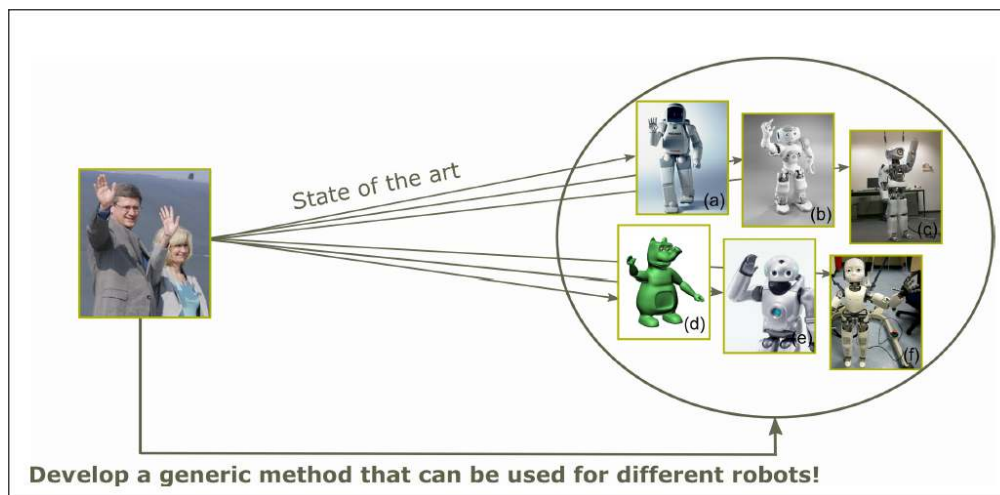

Figure 3: In the state of the art, gestures are always implemented for a specific robot platform. Our method aims to facilitate implementing gestures for a new robot platform by storing gestures independently of a morphology, and mapping them on a specific configuration. Robots: (a) ASIMO [10], (b) NAO [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 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 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 will 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, 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. 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) will be defined together with a Body Action Coding System (BACS), 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) [21].

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 (DOF), 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.
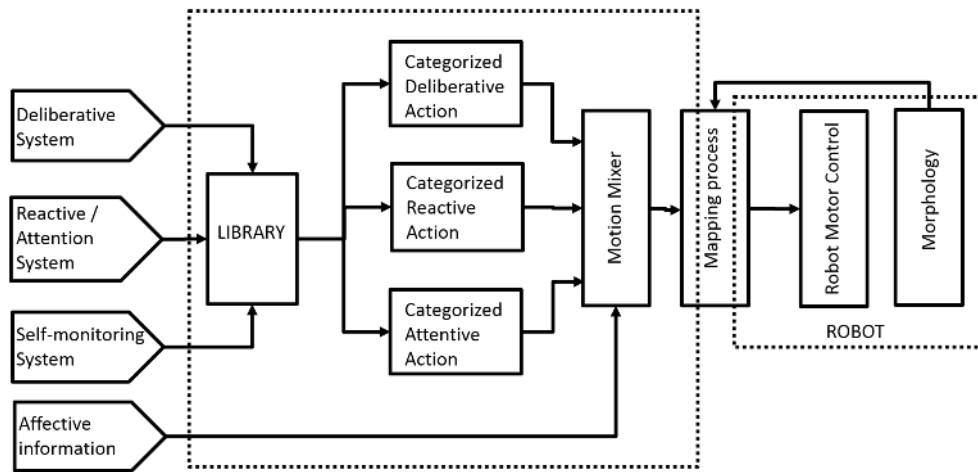

Figure 4: Overview of the Expression and Actuation subsystem. This subsystem receives inputs from several sources, categorizes them using the Library module and mixes them up to create a unique behaviour. Such behaviour is mapped into the joint configuration of the corresponding robot. This last process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

2.3   Actual work performed

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 already 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. A new

version of Probo is currently under development, see Annex 4.3 for further details.

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.
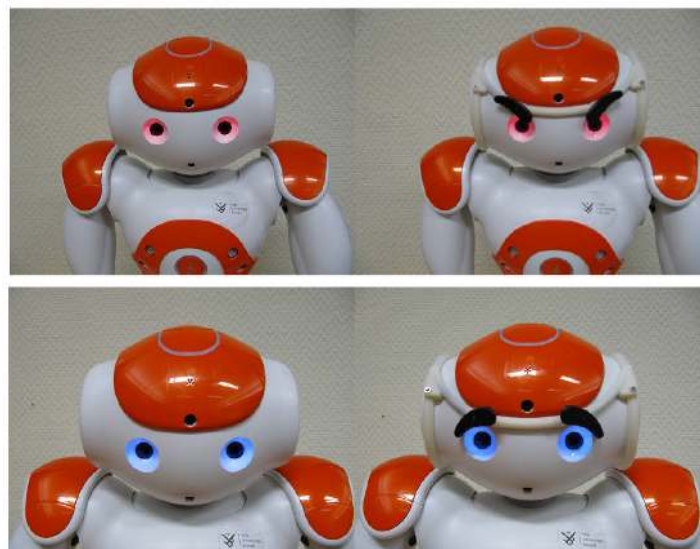

Figure 6: Comparison of the NAO robot expressing anger and sadness with (right) and without (left) the eyebrow system.

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 relative 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.

As mentioned before, the Expression and Actuation subsystem outputs the gesture to-be-performed by the robot. Such a gesture or behaviour is the result of the combination of several suggestions made by the other subsystems. Once we have the activated FAUs and BAUs, the subsystem should mix them using the Motion Mixer in Figure 4. We have already made some developments regarding this module. Based on insights from the animation industry it combines motions commands that are triggered by a human operator with motions that originate from different units of the cognitive control architecture of the robot, see [27] for further details.
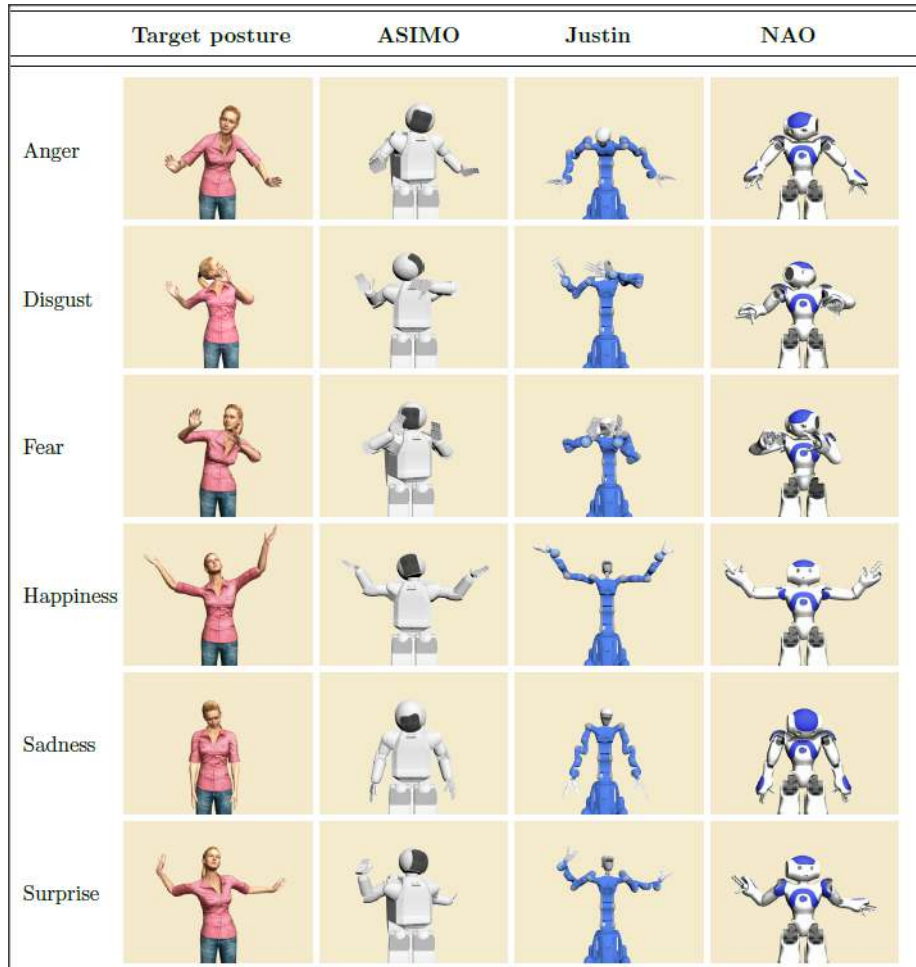
Figure 7: End postures of the gestures used in the survey. The first column shows the end posture of the target gestures for expressing the six basic emotions, while columns 2, 3 and 4 respectively show the mapped end postures for the robots ASIMO, Justin and NAO.

2.3.1 Action Primitives and Motor Execution

Once the desired 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). The preliminary version of this component has been implemented and is running as part of the turn-taking WoZ system deployed as part of T2.1.

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
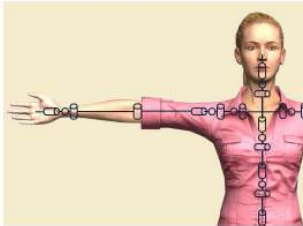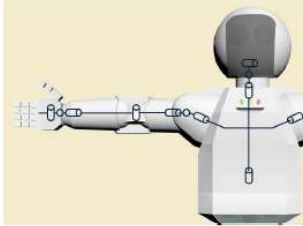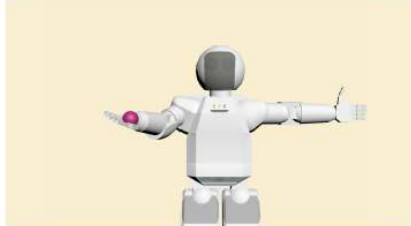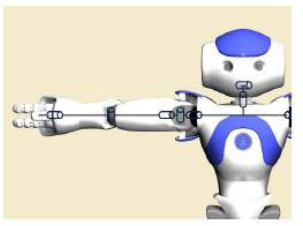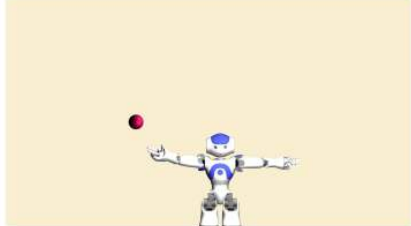
Configuration	Calculated posture
Config 1: 9 DOF arm	
Config 2: ASIMO	
Config 4: NAO	

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 $(34, -34, 38)$.

for different robot embodiments, without having to manually encode each specific action.

2.3.2 Preliminary 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 has been formulated. This first version of the component is 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, 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.

A description of the ports of this preliminary version of the component may be see in Figure 9. As per the software engineering standards, each of the port names shown is prefixed by the system and subsystem names. For the script manager, this prefix is "*/cognitiveController/expressionActuationSubsystem/*". In this preliminary version, each of the ports shown are of type *BufferedPort<Bottle>*. This affords

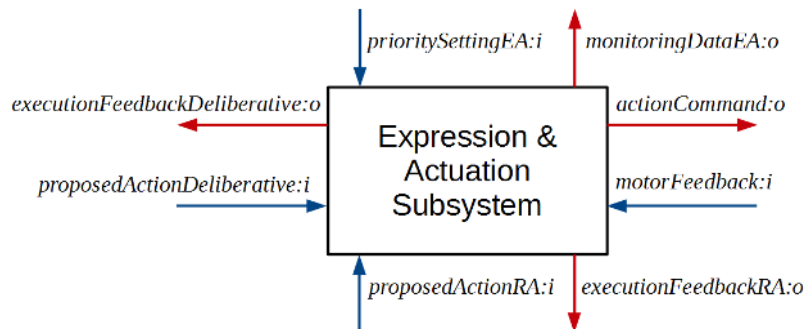Figure 9: Expression and Actuation subsystem component yarp ports: the prefix for the port names is listed in the main text. The functional description of this component may be found above.

maximum flexibility in the type of information that can be sent to/from this component (and indeed the others), although this then requires that a protocol is defined for the exchange of information within the *Bottle*.

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] Jelle Saldien, Bram Vanderborght, Kristof Goris, Michael Van Damme, and Dirk Lefeber. A motion system for social and animated robots. *INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS*, 11, 2014.

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.

Method - 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. To convey emotional expressions, the overall pose of the arms is important, and not (only) the exact position of the end effector. To ensure a good overall posture, it is necessary to take into account the relative orientations of every joint complex the robot has in common with a human. A base human model was defined, and the target postures were described by the orientation of the different joint complexes in the model. To make a certain model or robot perform a desired emotional expression, this information is mapped to its joint configuration. The necessary joint angles corresponding to the desired posture are calculated by using an inverse kinematics algorithm.

Results and Conclusion - 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. In general, the obtained postures are very recognizable, even for configurations with a limited number of joints. 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 and a separate movie for every robot performing each gesture and six additional movies showing a human virtual model performing the target gestures were made. 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. The overall rates for the correct linking of the gestures are relatively high, whereof we can conclude that the calculated gestures in general well resemble the target gestures from the database and therefore, that our method to map gestures to different robot configurations gives good results.

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. 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 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.

Method - 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). Beatboxs afterwards offered the official source code to make the robotic toy completely hackable. My Keepon uses two microprocessors (PS232 and PS234) to control the movements and the sounds, which communicate with each other via I^2C 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^2C bus.

Results - 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. 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.

Conclusion - We present a method to modify My Keepon to be a programmable research platform for HRI studies. 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.

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 De Beir, A. (2016), Evolutionary Method for Robot Morphology: Case Study of Social Robot Probo

Bibliography - De Beir, A., and Vanderborght, B. (2016). Evolutionary Method for Robot Morphology : Case Study of Social Robot Probo. *In Proceedings of the Annual ACM/IEEE International Conference on Human Robot Interaction*

Abstract - The appearance of robots is often made arbitrary as it relies more on guidelines than on a rigorous methodology. This paper presents a novel method of using genetic algorithms (GA) to improve the appearance of social robots with human feedback. Such general methods are interesting as they do not require prior artistic experience from the designer and can integrate the end-user in the loop. As a proof of concept, we carry out a case study by applying this method to the new design of the social robot Probo. Using designer feedback, the robot is evolved from its original design over five populations composed of 15 individuals. An online survey shows that the evolved designs are significantly improved compared to the original. These results indicate the feasibility of the method employed and gives rise to the possibility of non-technical end-users influencing the design of robot morphologies adapted for specific human-robot interaction requirements.

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

4.4 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.

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.
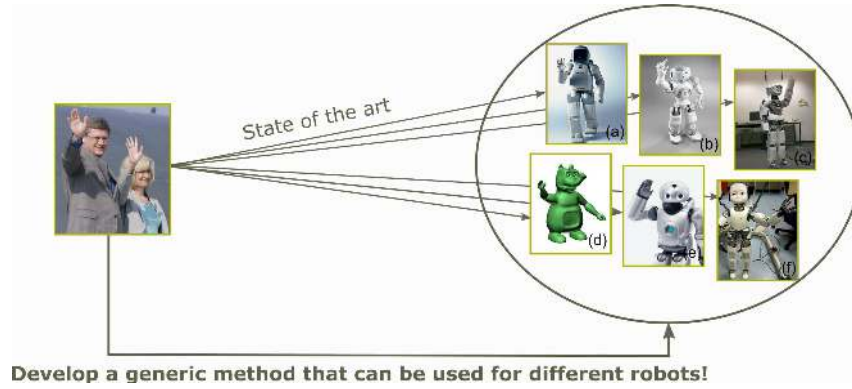
Figure 1. In the state of the art, gestures are always implemented for a specific robot platform. Our method aims to facilitate implementing gestures for a new robot platform by storing gestures independently of a morphology, and mapping them on a specific configuration. Robots: (a) ASIMO [19], (b) NAO [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 2 3	Flexion/extension of neck Abduction/adduction of neck Rotation of neck
Body	Body	4 5 6	Flexion/extension of spinal column Lateral flexion of spinal column Transversal rotation of spinal column
Arm	Clavicle	7 8	Abduction/adduction of shoulder girdle Elevation/depression of shoulder girdle
	Shoulder	9 10 11	Flexion/extension of shoulder Abduction/adduction of shoulder Inward/outward medial rotation
	Elbow	12	Flexion/extension of elbow
	Wrist	13 14 15	Pronation/supination of elbow Flexion/extension of wrist Abduction/adduction of wrist
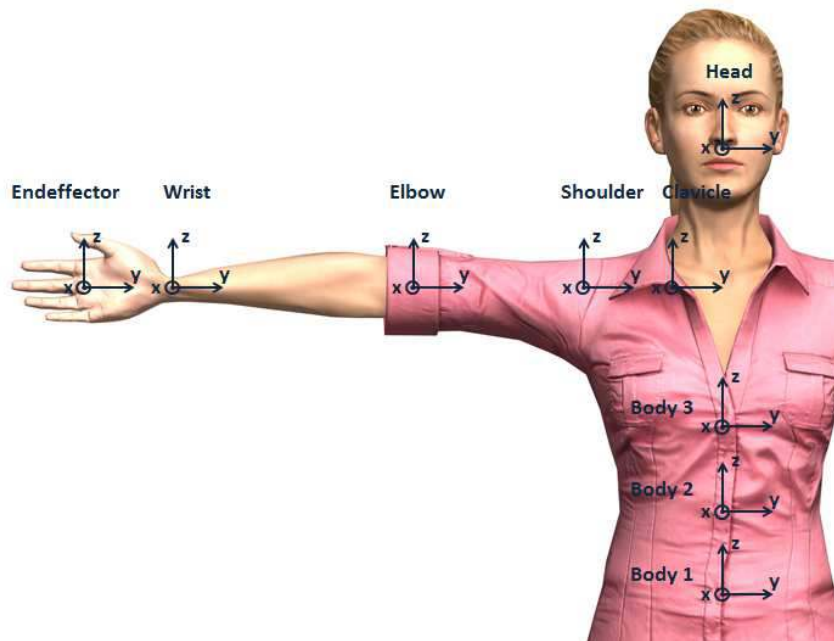
Figure 2.   A reference frame was assigned to each block. For the body 1 block, the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle's reference axes are the body 3 - embedded axes. For all other blocks 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:
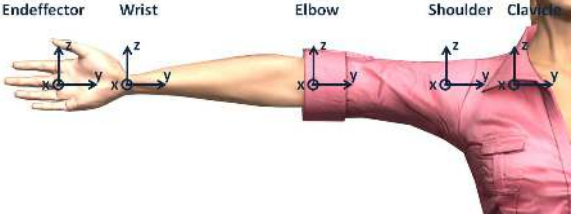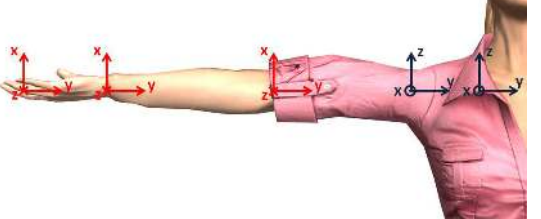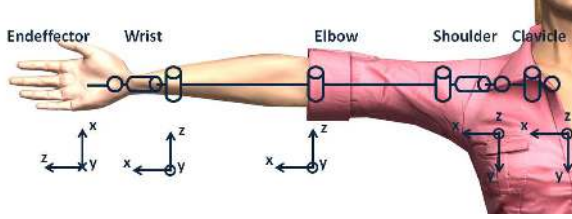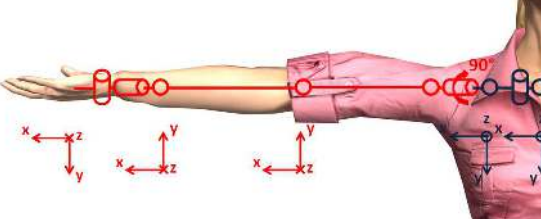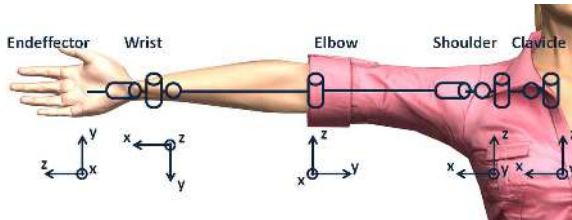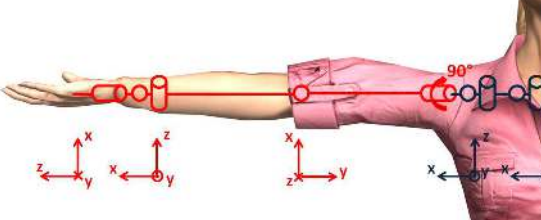
$$R_i = {}^{b,i}R_{st} \cdot R_{i,des} \cdot {}^{st}R_{e,i} \tag{1}$$

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:

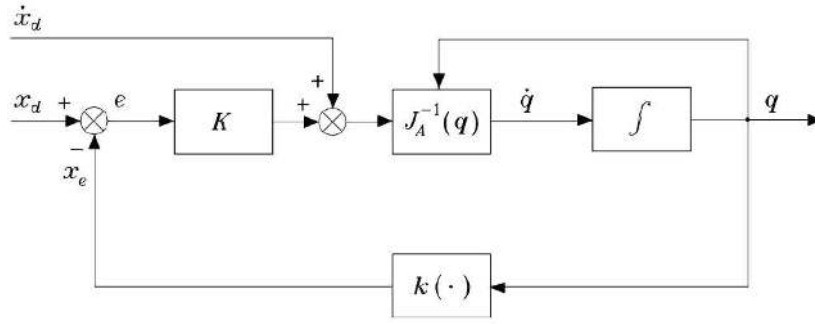$$\dot{q} = J_A^{-1}(q)(\dot{x_d} + K(x_d - x_e)) \tag{2}$$

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
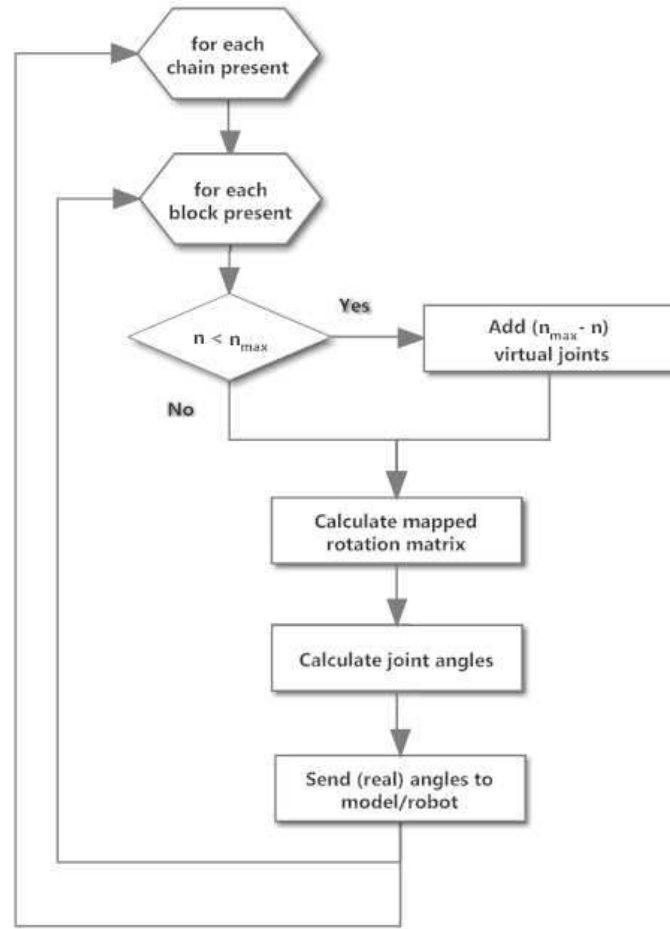
Figure 4. Schematic representation of the program flow. For every block present in the configuration, a mapped rotation matrix is calculated. The corresponding joint angles are calculated by inverse kinematics and sent to the virtual model or robot.

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

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
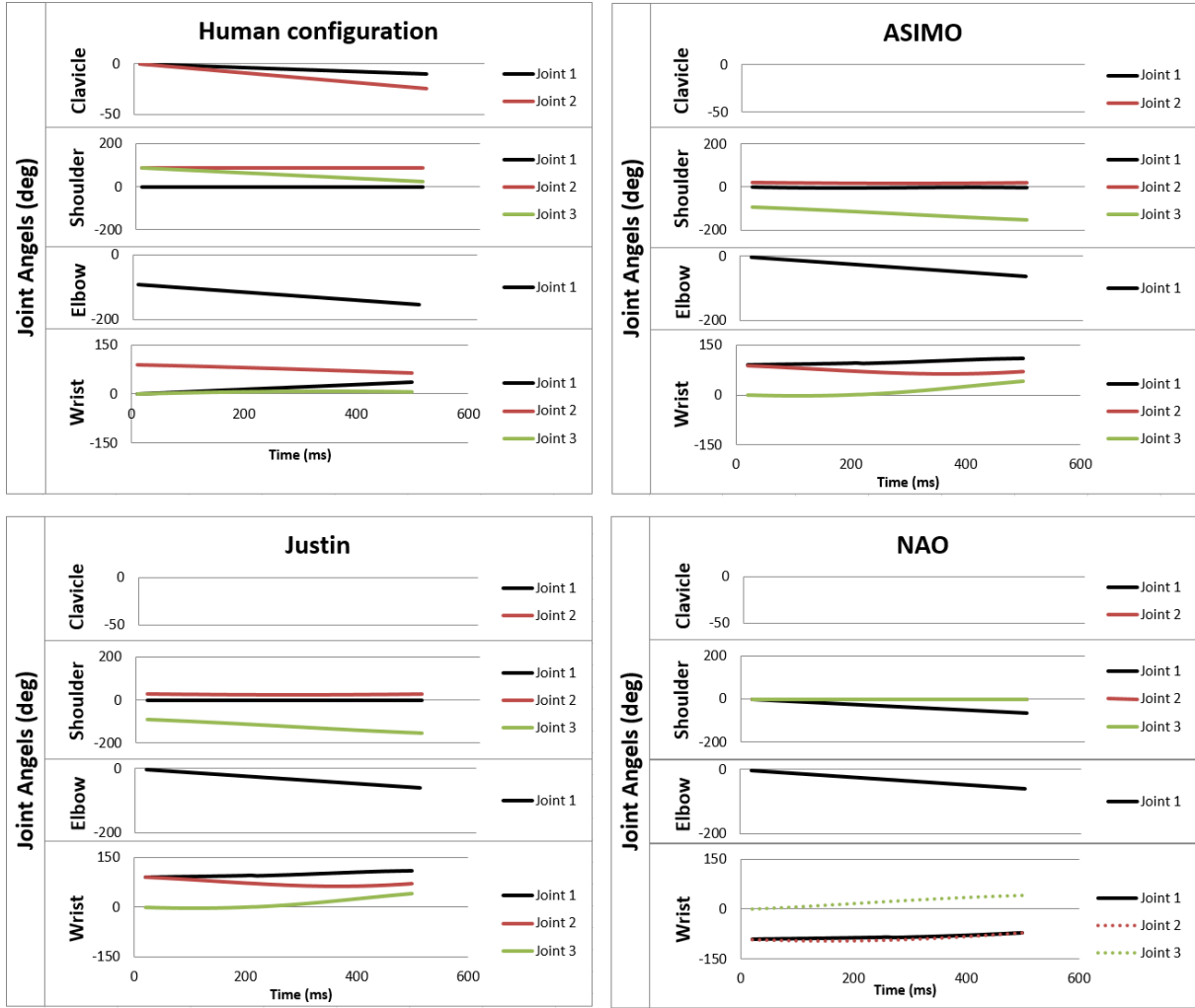
Figure 5.   This figure shows the calculated joint trajectories for the left arm chain for the four configurations listed in Table 3 when going from the T-pose to the end posture for the expression for *happiness*. As missing blocks are ignored by the method, no trajectories are calculated for the clavicle 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.
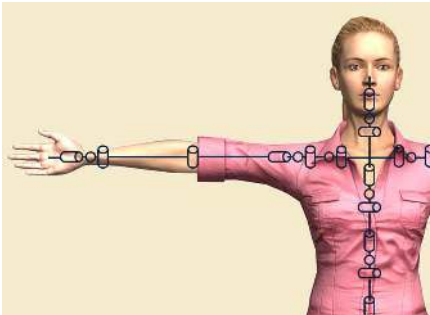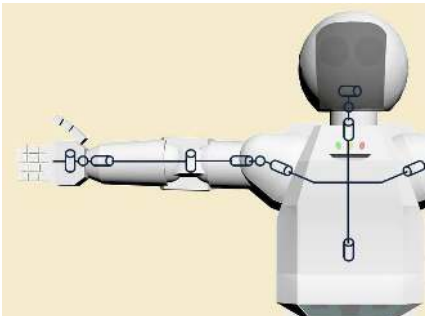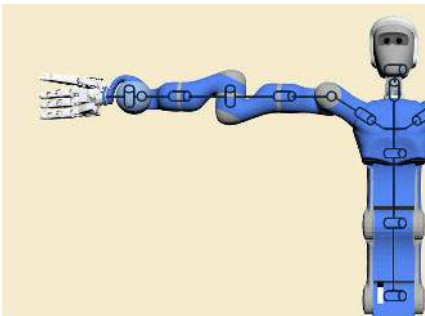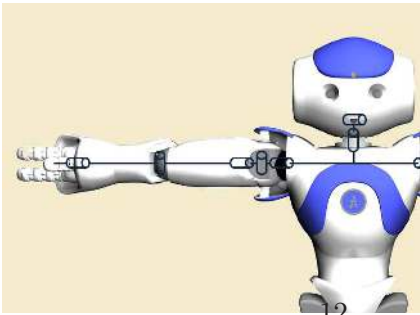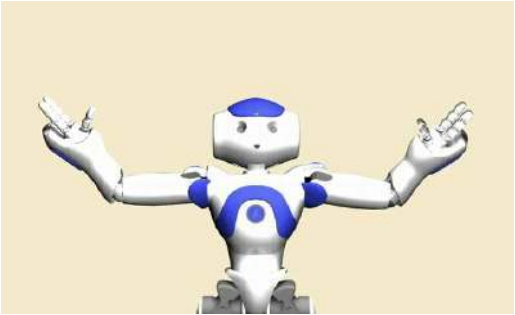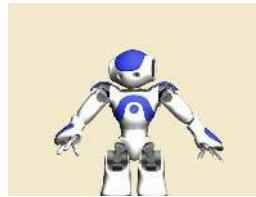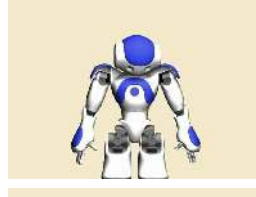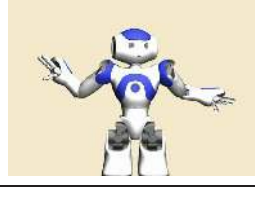
Configuration	Calculated posture
Base model	
Config 1: full configuration	
Config 2: ASIMO	
Config 3: Justin	
Config 4: NAO	

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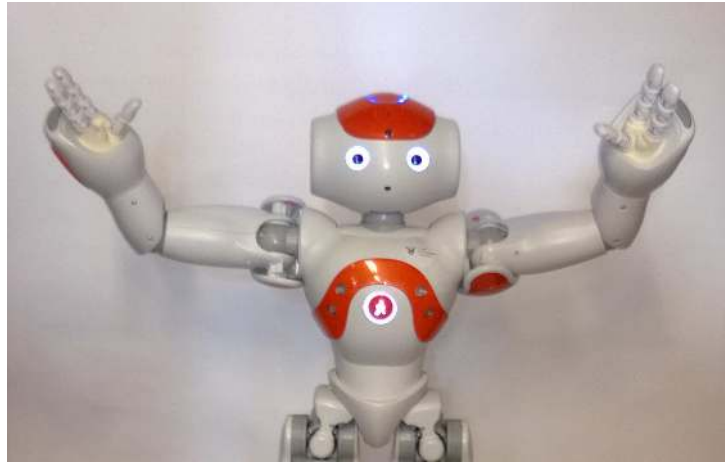
Figure 7. Current work includes validating the method on the real robots instead of their virtual model.

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 behaviors have recently been developed for Human-Robot Interaction (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]. According 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 platforms. Successful robots such as Asimo [4], Nao [5], iCub [6], HRP-4C [7], Probo [8], Keepon Pro [9], are expensive

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

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 recently 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 Emotiv neuroheadset will be introduced as a general example

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 $30,000 research model developed by Hideki Kozima, and is sold commercially to Beatbots. It appears as a small yellow creature-like robot designed for simple, natural, nonverbal interaction with children to study social development of autistic children [9]. According to Beatbox website [21], Keepon Pro structure has four degrees of freedoms (DOFs): turning ($\pm180°$), nodding ($\pm40°$), rocking side-to-side ($\pm25°$), 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)
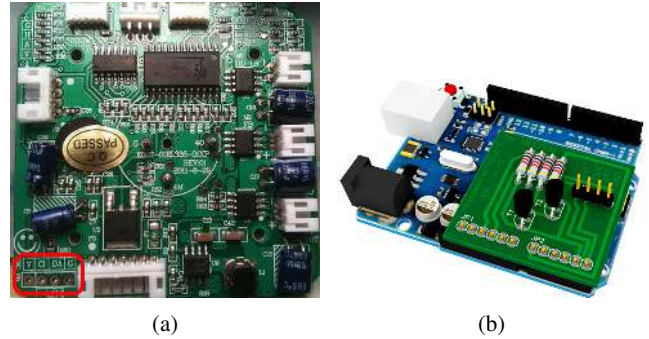
Fig. 2: My Keepon controller board (left) and Arduino shield for the modification (right).

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
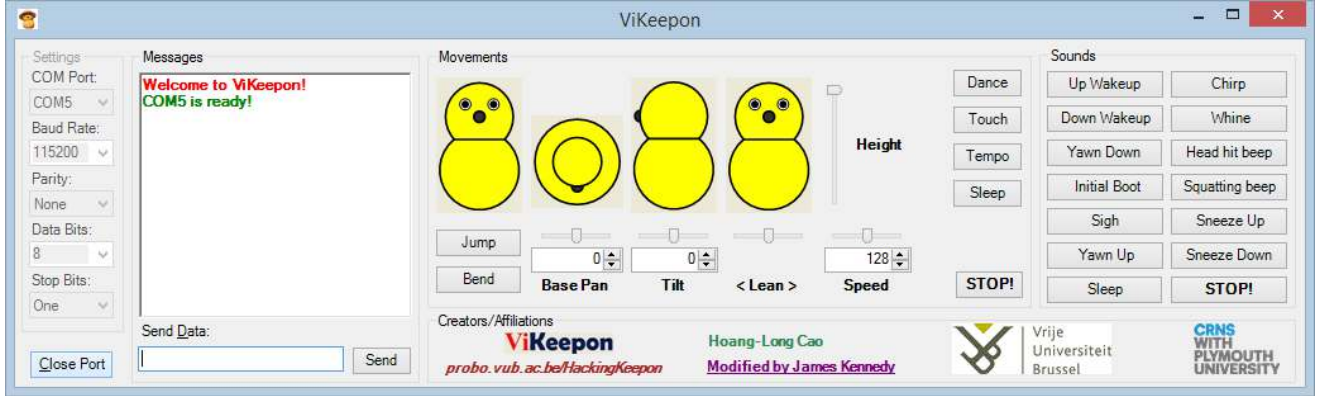
Fig. 3: Basic GUI to control the modified My Keepon. Adapted depending on specific experiments.

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
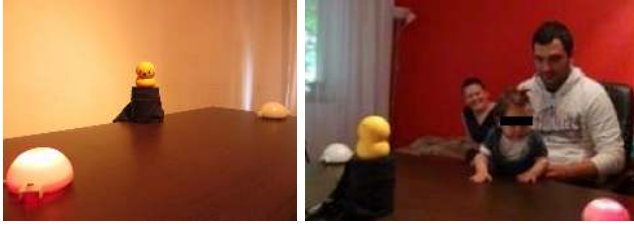
Fig. 6: Joint attention study with the modified My Keepon.

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.

ACKNOWLEDGMENT

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.

Keywords Facial Expressions · Emotions · Nao Robot · Eyebrow · Humanoid Robot · Eyes Color

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
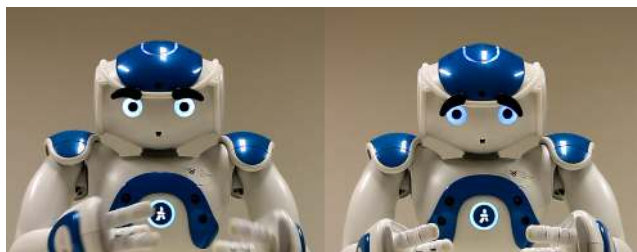
A. De Beir (✉)
Vrije Universiteit Brussel, Robotics & Multibody Mechanics Research Group, Pleinlaan 2, 1050 Brussels, Belgium
E-mail: albert@debeir.com
Website: http://www.dream2020.eu

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
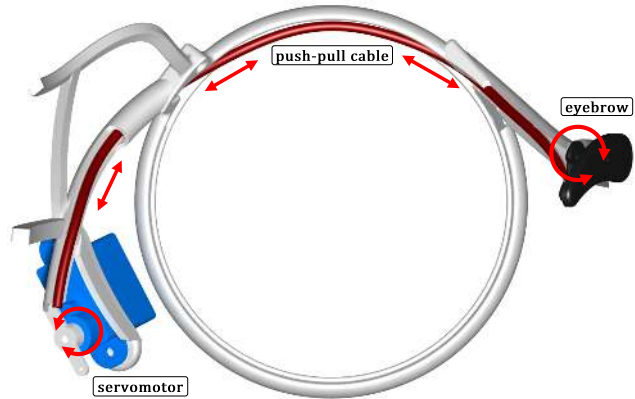

Fig. 2: Actuation of the eyebrows. The arrows show the conversion between rotation and translation. The white structure is 3D printed and can be cliped on the robot's face.

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
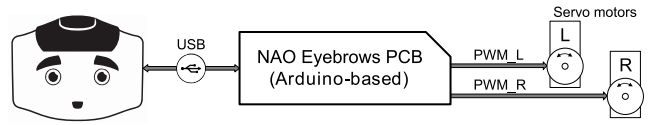
(a) An Arduino-based PCB is connected to the USB port of NAO robot to control two servo motors. Each servo motor can be controlled separately.

(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.
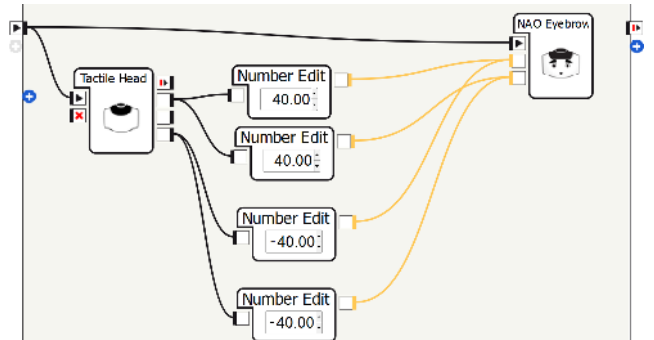

Fig. 4: An example of using tactile head sensors to control two movements of the eyebrows in Choregraphe. On the top right the *"NAO Eyebrows"* box allows users to easily control the positions of the NAO eyebrows.

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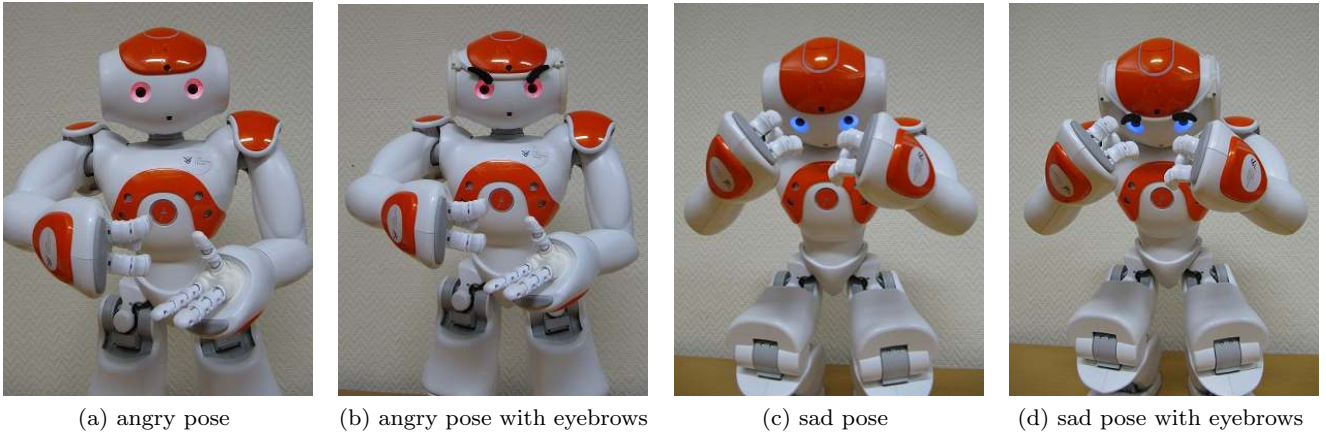
Fig. 5: Pictures used in the first experiment. For each picture, participants had to guess the robot emotion. We used poses expressing emotions from [7]. For each pose, the robot was either with or without eyebrows.

`["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 variated 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.
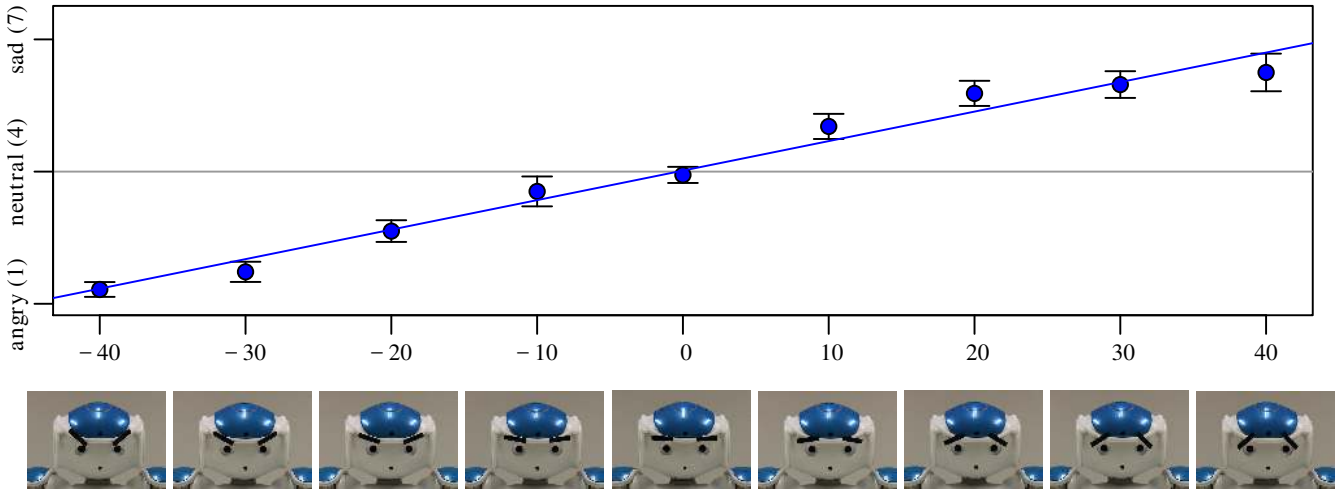
Fig. 6: Emotion expressed by NAO robot in function of the eyebrows' angle (in deg). Emotion was rated on a Likert scale from 1 (*anger*) to 7 (*sadness*). Dots represent the mean of the Likert scale value for each angle. Error bars indicate 95% confidence intervals. The blue line is the linear regression.

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).
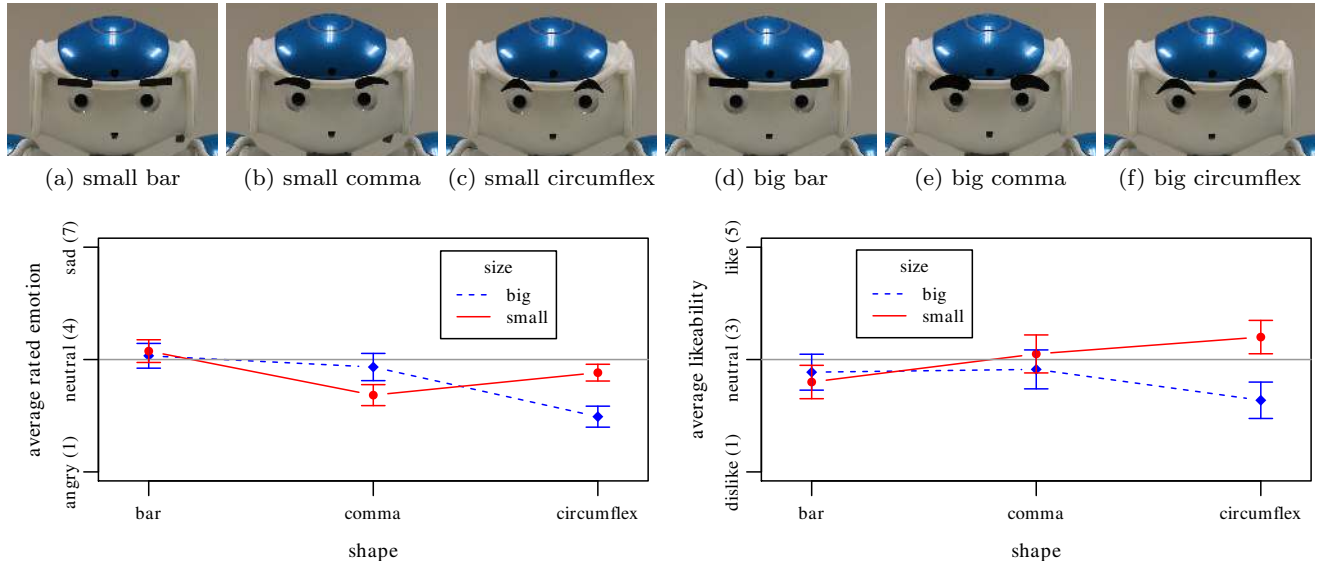
Results and discussion

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
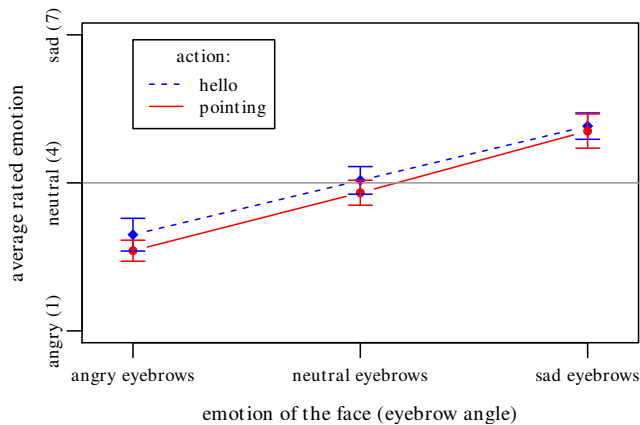

Fig. 8: Experiment 4: Interaction plot of the rated emotions for the videos. Error bars indicate 95% confidence intervals. We consider two factors: action performed and emotion of the face.

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.
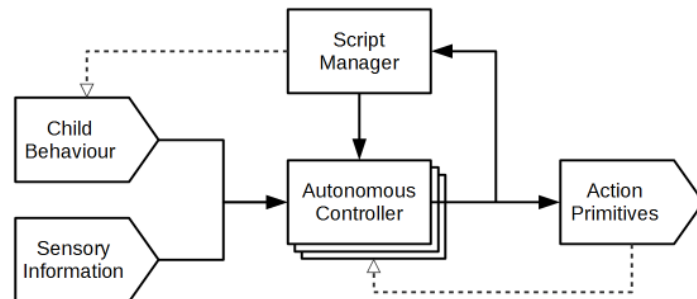

Figure 1: High level description of the robot control system; arrows denote information flow on YARP ports. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with 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.
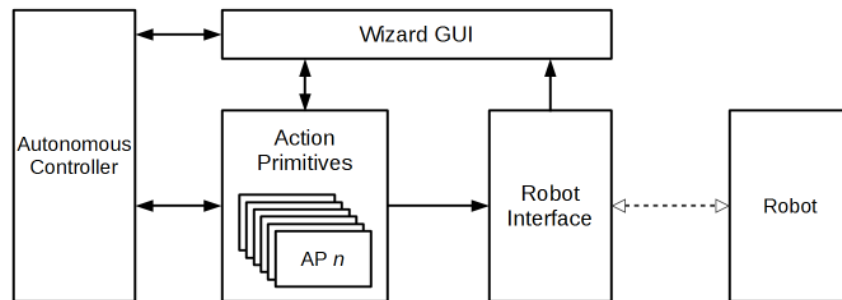
Figure 2: The Robot Interface coordinates motor control for the robot, and provides information to the Wizard GUI (and autonomous controller - not shown) regarding robot/behaviour state. Solid arrows denote information flow on YARP ports, dashed arrow shows communication using robot API (e.g. naoqi API 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
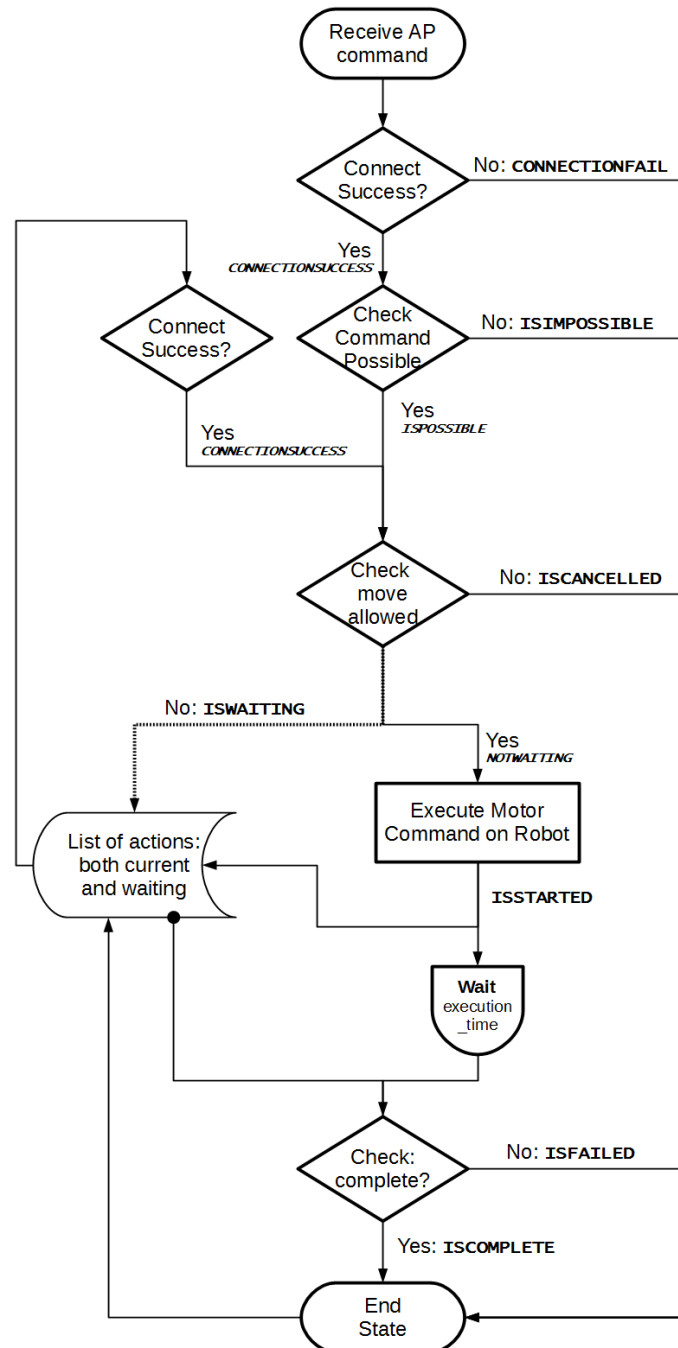
Figure 3: Low level motor control, showing feedback states (in bold, italicised shows additional debug feedback) from various stages emitted on a YARP port. Input is the request (from autonomous action selection or wizard interface) for execution of a single Action Primitive (AP). Dotted arrow denotes 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).
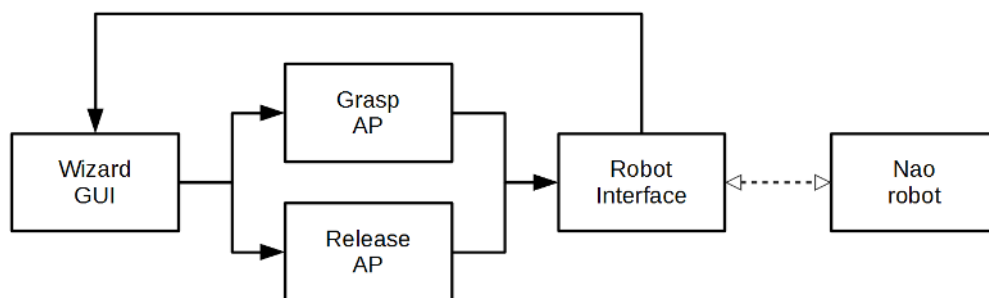

Figure 4: Basic system demonstration of low-level control described in section 3. The Wizard GUI has controls and displays feedback from the Robot Interface (see table 1).

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 inter-actions: 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.
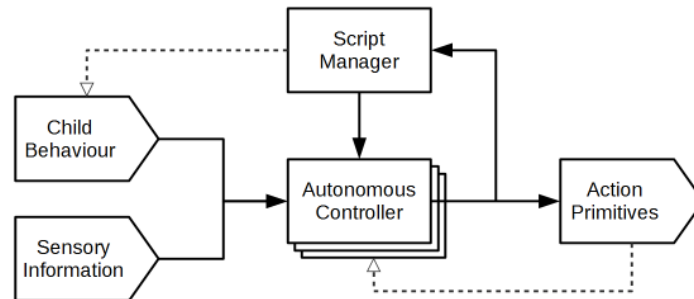

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied.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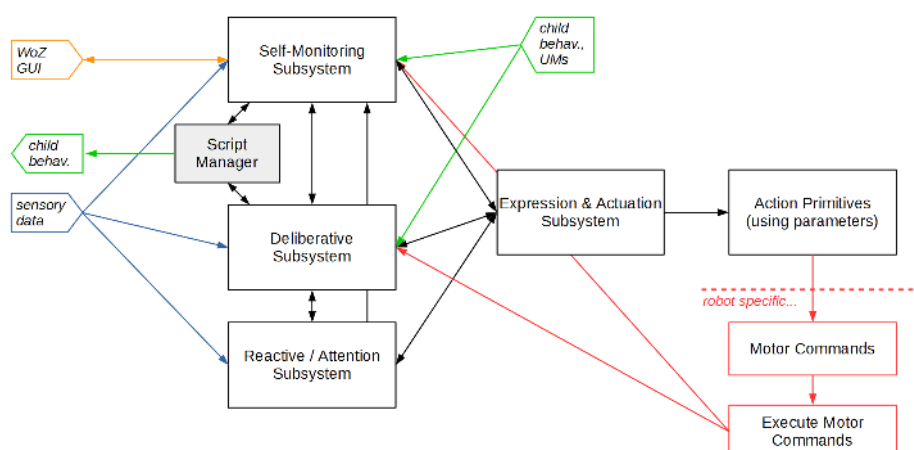

Figure 2: Description of the cognitive controller subsystems. The script manager is separate from, but tightly interacts with, the deliberative subsystem to enable the robot control system to generate appropriate social/interaction behaviour even in the absence of an explicit interaction script. UMs: 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.
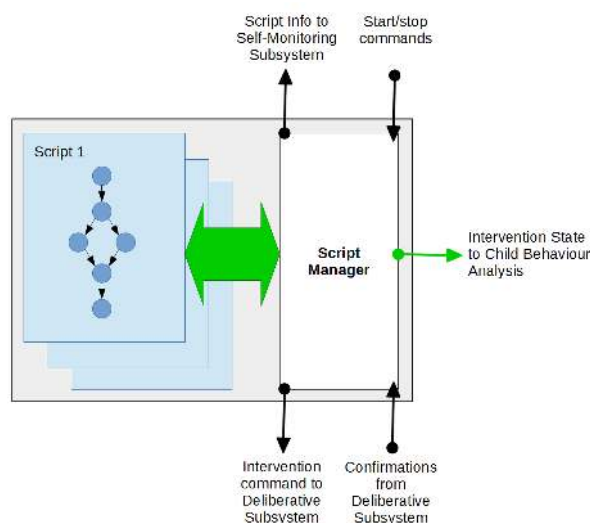
Figure 3: Overview of the script manager subsystem. The scripts are defined independently of the script manager, which is responsible for stepping through the script as appropriate and communicating with the other subsystems as required.

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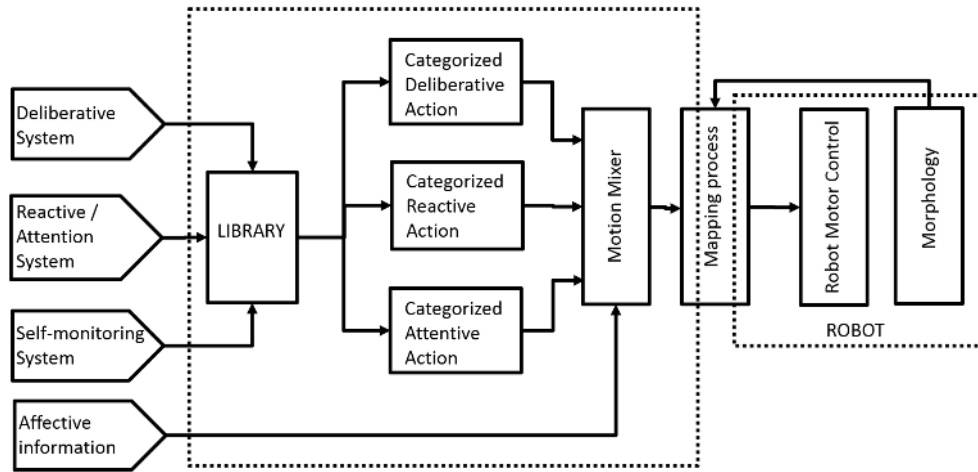
Figure 4: Overview of the Expression and Actuation subsystem. This subsystem receives inputs from several sources, categorizes them using the Library module and mixes them up to create a unique behavior. Such behavior is mapped into the joint configuration of the corresponding robot. This last process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

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
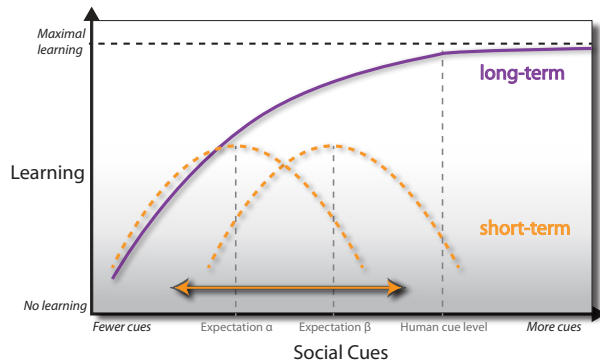email: {james.kennedy, paul.baxter, tony.belpaeme}@plymouth.ac.uk

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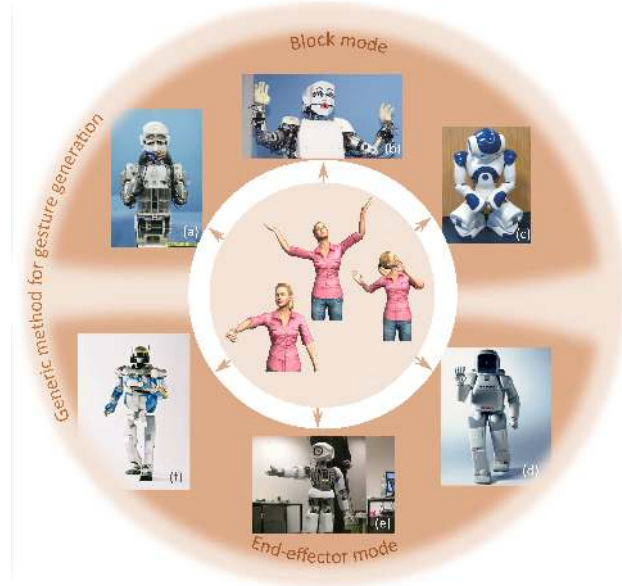 *Email address:* `Greet.Van.de.Perre@vub.ac.be` (Greet Van de Perre)

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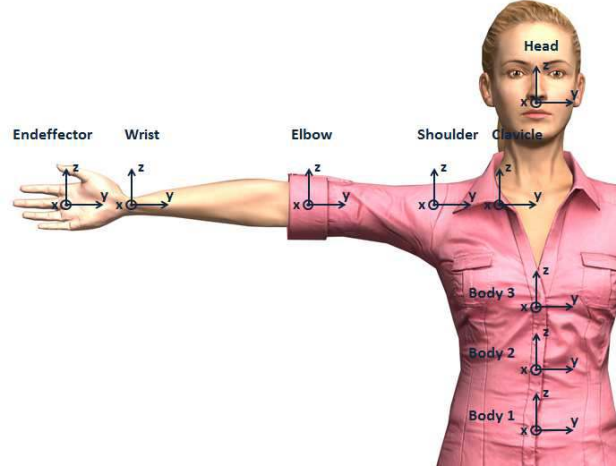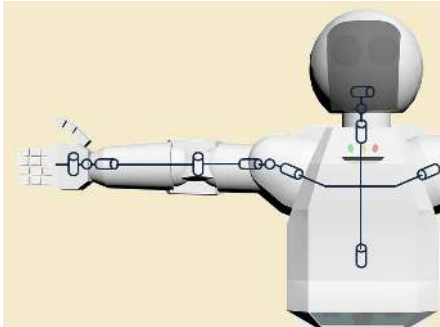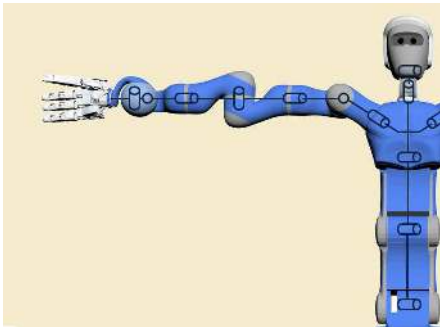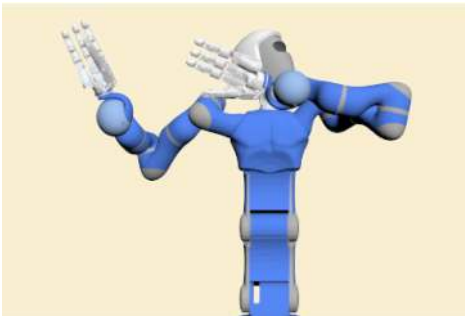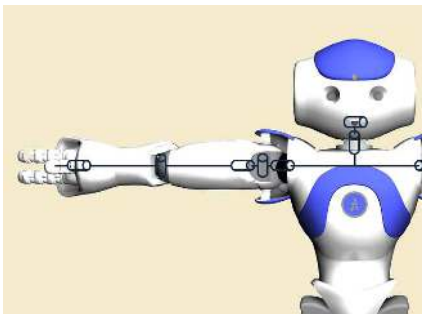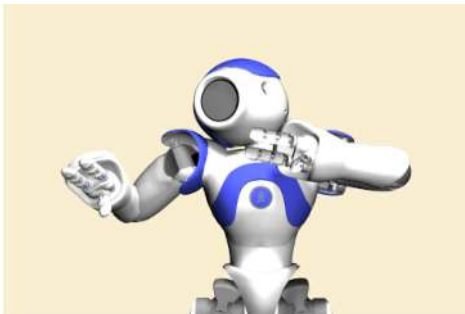
Figure 2: A reference frame was assigned to each block. For the body 1 block, the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle's reference axes are the body 3 - embedded axes. For all other blocks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

5

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
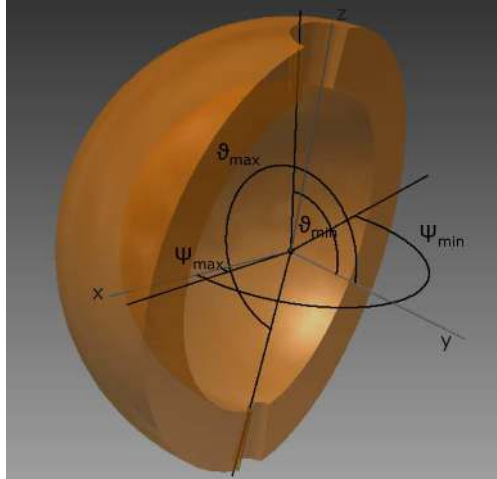
Figure 3: Example of an approximated workspace.

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
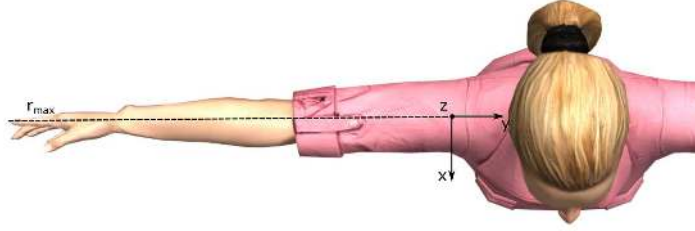
Figure 4: Determination of the maximum radius r_{max}: the maximum reachable distance of the end effector, measured from the shoulder base frame origin.
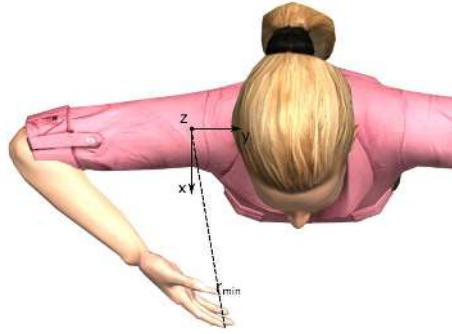

Figure 5: Determination of the minimum radius r_{min}: the elbow joint is placed in maximum flexion while the other joint angles correspond to the T-pose angles. The distance between the shoulder base frame and end-effector corresponds to r_{min}.
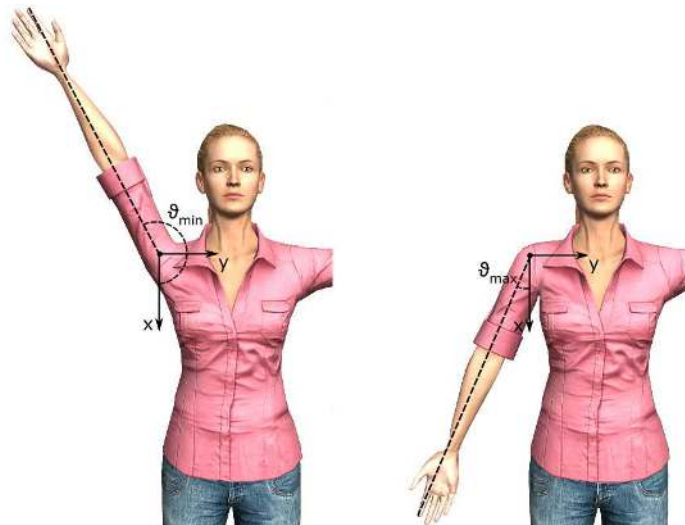

Figure 6: Calculation of the minimum and maximum value for θ; the arm is placed in respectively, maximum abduction and maximum adduction and the angle formed by the end-effector is calculated.
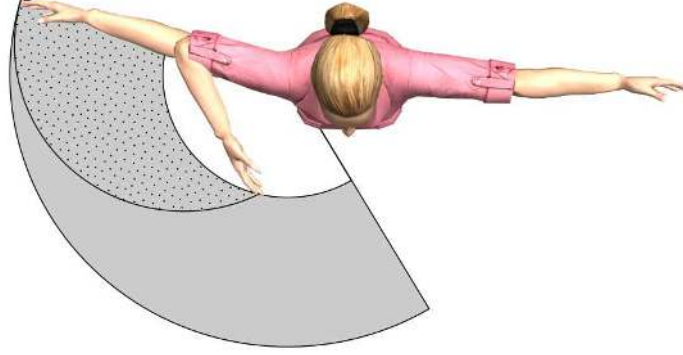
Figure 7: View of the calculated workspace in the horizontal plane crossing the shoulder base frame. The area covered by a circle with the centre point located in the elbow base frame and radius equal to the length of the lower arm (dotted surface) needs to be subtracted from the workspace (grey surface). 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
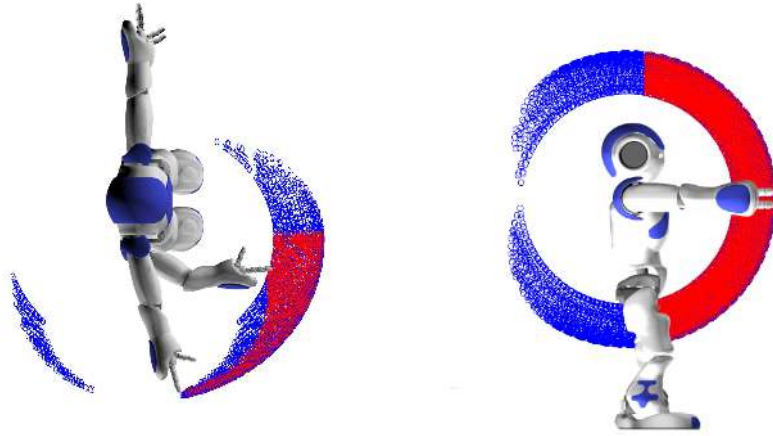
Figure 8: Calculating an approximate workspace. Blue dots indicate the real workspace, red dots the approximation. Left: xy-cross section passing the shoulder base frame. Uncovered blue dots in the top right result from not taking into account the configurations involving elbow flexion for maximum horizontal 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.
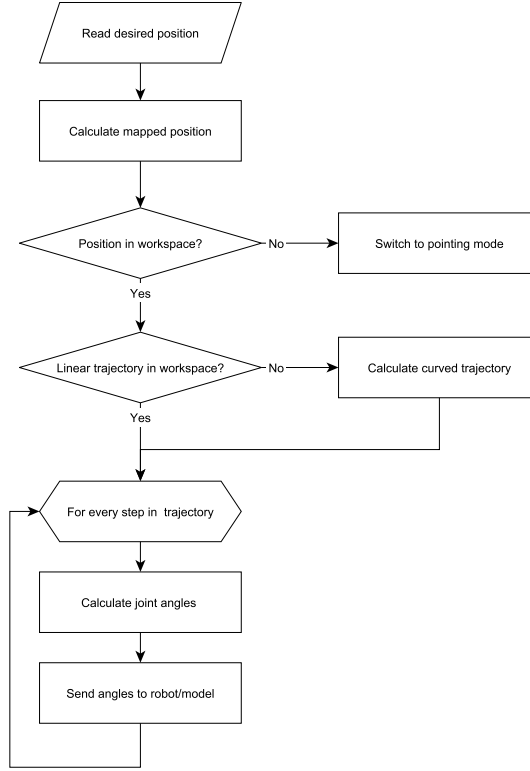
Figure 9: Schematic representation of the work flow for the place-at condition.

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.
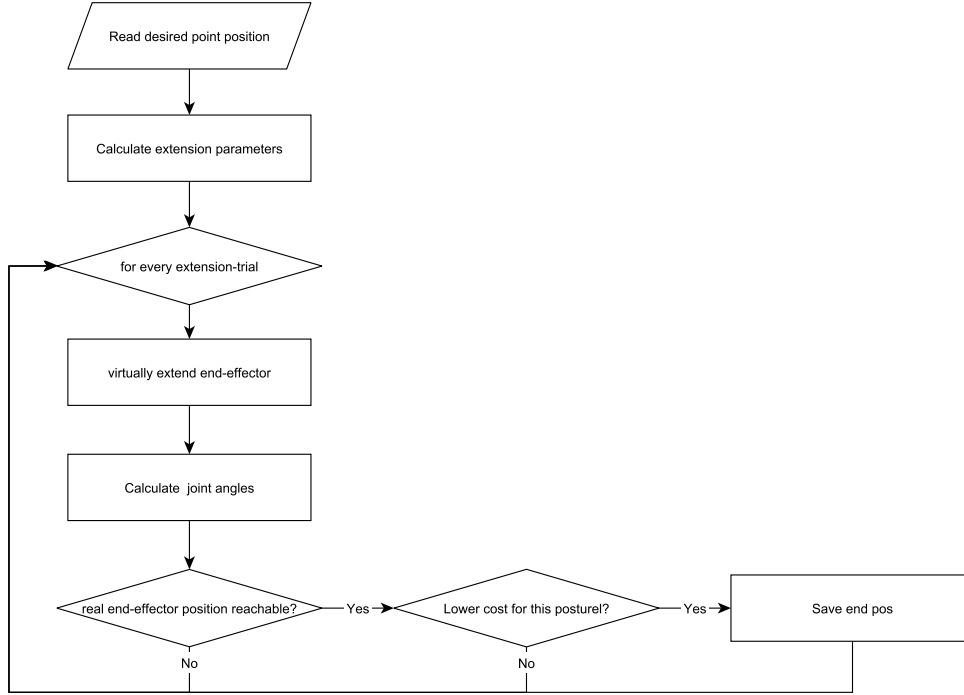
Figure 10: Schematic representation of the work flow for the pointing condition.

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.
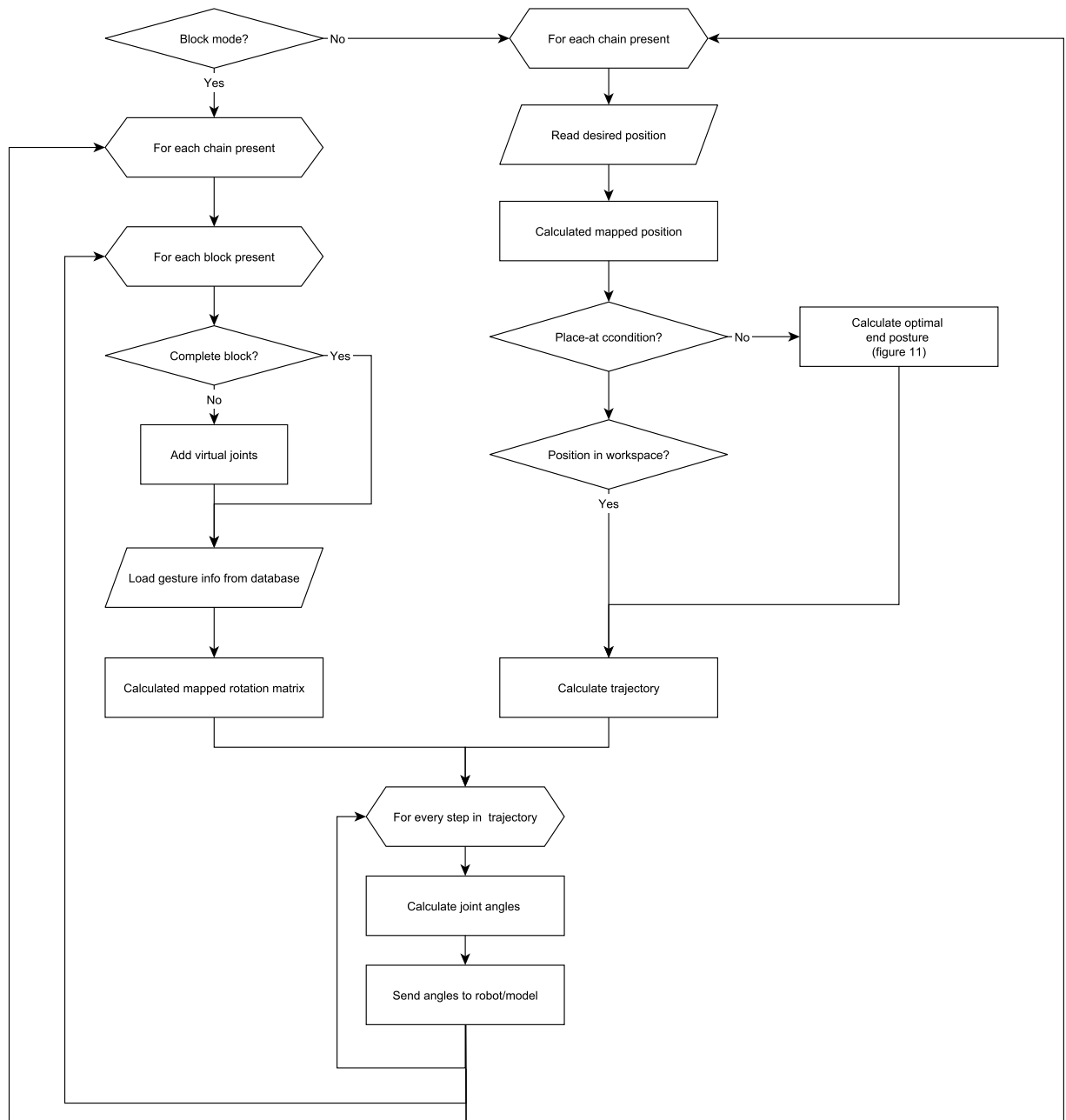
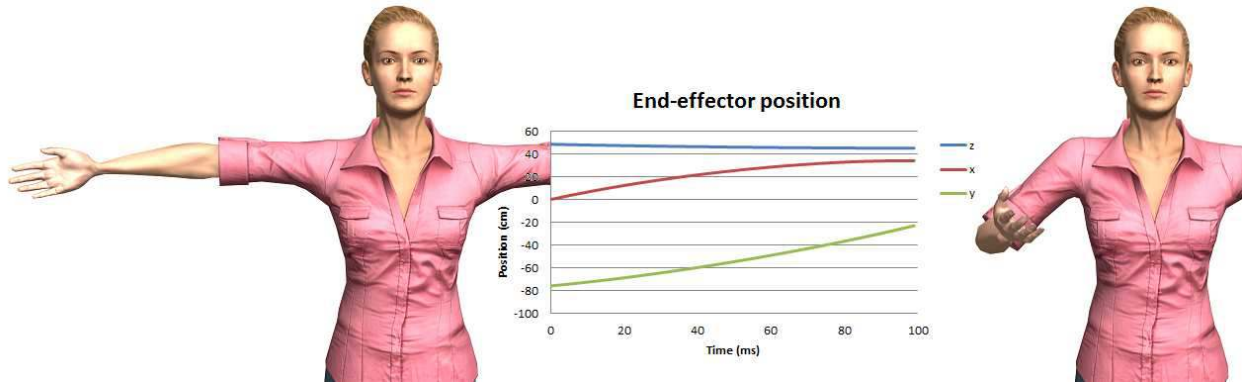Figure 11: Simplified work flow of the complete method.

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.
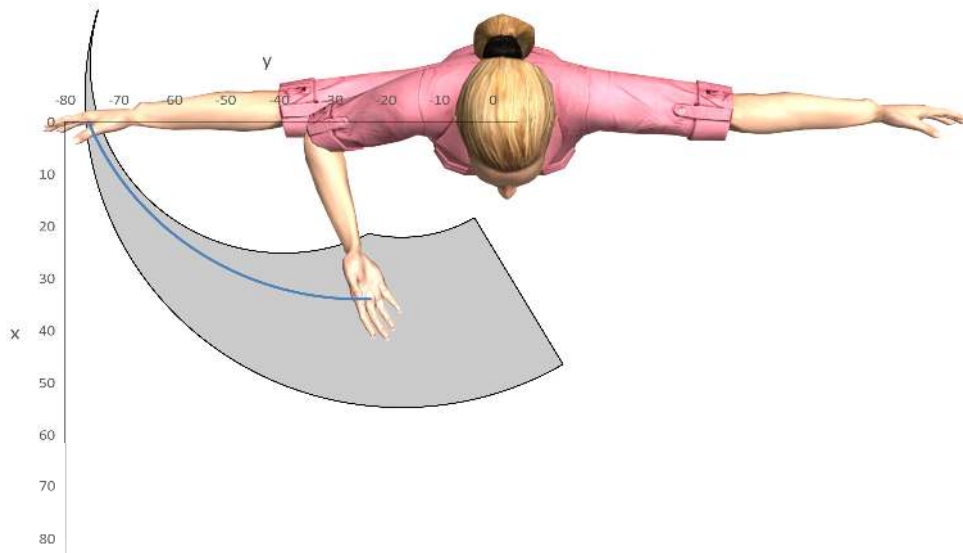

Figure 13: Top view of the calculated trajectory for a place-at task for the right hand with a start position of $(0, 76, 48)$ cm and end position of $(34, -23, 45)$ cm, superposed on the xy-cross section of the right arm workspace.
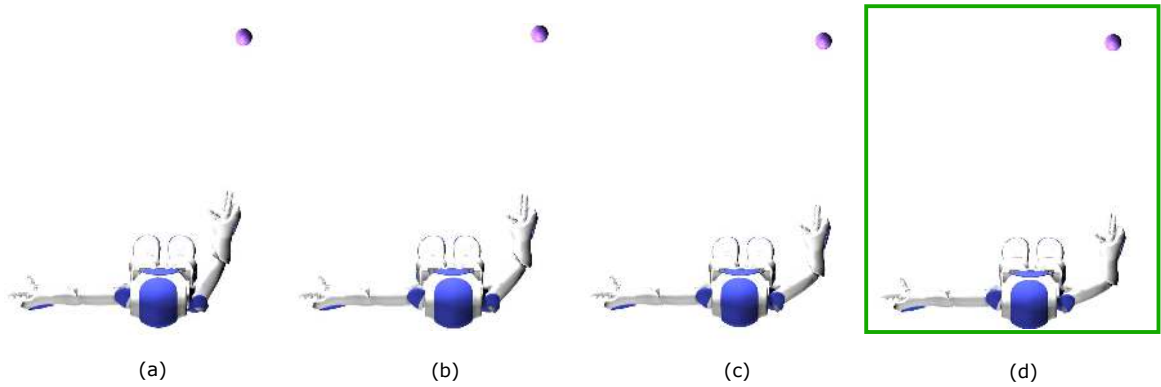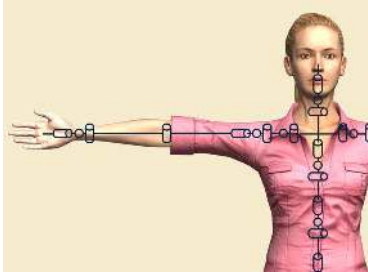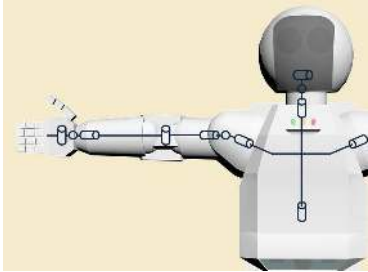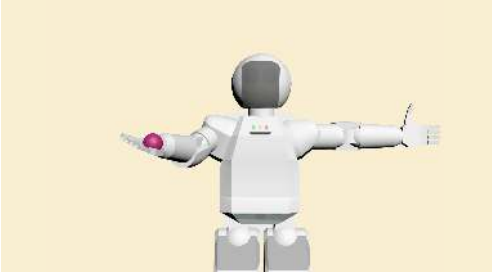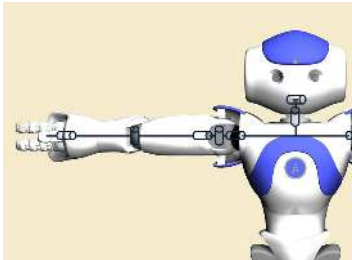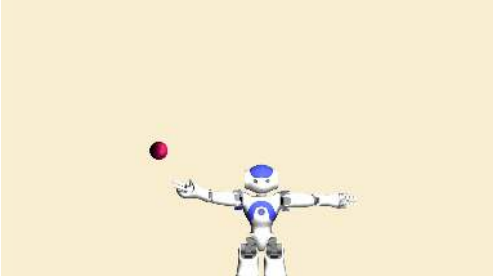

Figure 14: In the pointing condition, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function.(a) 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.

Evolutionary Method for Robot Morphology: Case Study of Social Robot Probo

Albert De Beir and Bram Vanderborght

Robotics & Multibody Mechanics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Abstract—**The appearance of robots is often made arbitrary as it relies more on guidelines than on a rigorous methodology. This paper presents a novel method of using genetic algorithms (GA) to improve the appearance of social robots with human feedback. Such general methods are interesting as they do not require prior artistic experience from the designer and can integrate the end-user in the loop. As a proof of concept, we carry out a case study by applying this method to the new design of the social robot Probo. Using designer feedback, the robot is evolved from its original design over five populations composed of 15 individuals. An online survey shows that the evolved designs are significantly improved compared to the original. These results indicate the feasibility of the method employed and gives rise to the possibility of non-technical end-users influencing the design of robot morphologies adapted for specific human-robot interaction requirements.**

I. Introduction & Related Work

The appearance of social robot is important as it creates expectations about the robot's behavior and mental state [1]. For example, a baby morphology will create an air of cuteness and low expectations concerning the robot's capabilities. Additionally, interaction is enhanced if humans are attracted to or interested by the appearance of the robot [2].

Researchers have identified key elements in attractive robots such as the mimicking of human traits (eyebrows, mouth, eyes, etc..) or newborn proportions (big eyes, round face) [1], [2]. However these guidelines are difficult to implementand and require experienced designers capable of implicitly fine-tuning these attractiveness features. However, there are actually no scientific methods or tools to generally improve robot morphology.

In this work, we use a genetic algorithm (GA) with human judgement being part of the algorithm to evaluate robot morphology. The selection process is performed by the end-users in order to improve the robot appearance according to their taste. GAs have been used for Computed Aided Design (CAD) [3] and the generation of iconic faces [4], but never to improve the appearance of a robot.

II. Method

In the proposed GA (see Figure 1), the following steps are iteratively performed: 1–A population of robots with a variation of appearances is presented to humans. This robot appearance (phenotype) is coded through a set of variables (genotype). 2–Humans select the robots with the appearance they like the most. 3–The features of these selected robots
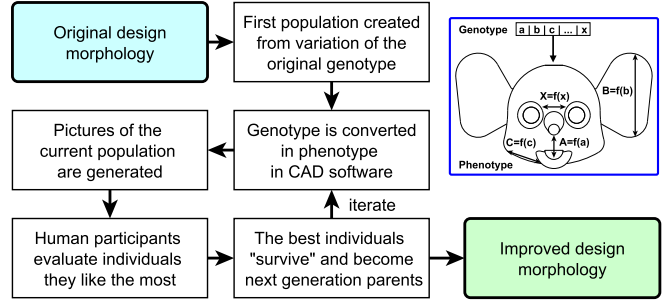
Fig. 1. Working principle of the GA to improve appearance of robots.

are recombined to create a new generation using cross combination and mutations. This Darwinist selection is similar to the morphological evolution of domestic animals. For example cute dogs are more likely to be feed and breed by humans and therefore more likely to reproduce [5].

This method has many advantages. First, it is capable of improving robot's appearance without requiring any artistic skills from the roboticist. A such GA is made for optimization and does not create a robot from scratch, the roboticist can focus on technical design difficulties then let the robot evolves to a proper good looking shape. Second, randomness generated by evolutionary algorithms allows to explore unexpected solution and go out of designer's preconceived ideas. Third, evolution works through local comparisons of the current population and do not require knowing the "optimal" solution to progress. Finally, GAs can converge toward multiple local maxima, therefore providing many suitable alternatives to the roboticist.

III. Experiment

In order to demonstrate the feasibility of the method, we implemented it in a case study to design a new version of the social robot Probo [6]. The genotype of the individual is composed of 17 variables that determine the phenotype. For example, if X is the ratio between the eyes (compared to the size of the head), then $X = x + \beta$; where x is a variable of the genotype bounded between 0 and 1, and β is a positive constant to ensure that the ratio X is strictly superior to 0. If x is small, the eyes of the robot will be close, similarly if x is big, the eyes will be distant.

Recombination and mutation of the genetic algorithm is performed with a *MATLAB R2014b* script, and the genotype of the population is exported to an excel file. This excel file

(a) original individual   (b) individual 1   (c) individual 2

(d) individual 3   (e) individual 4   (f) individual 5
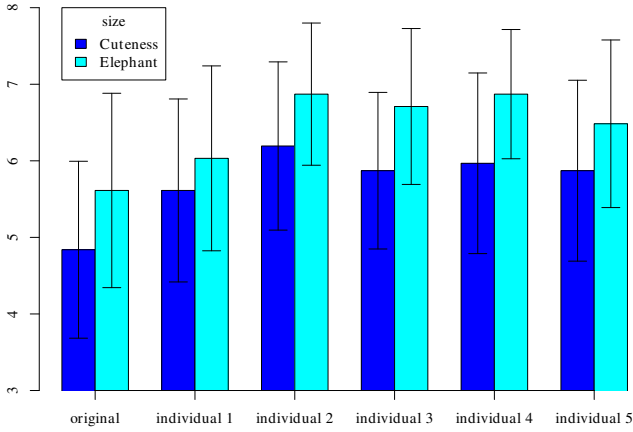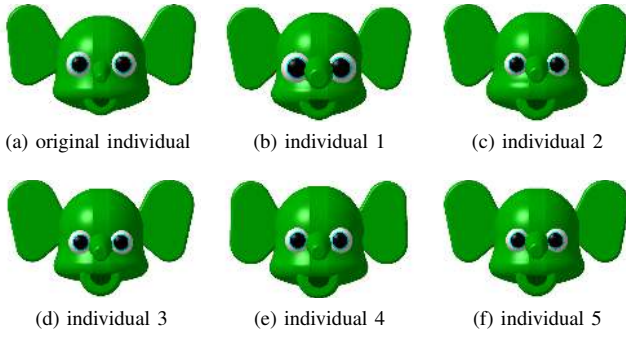
(g) Mean rating of the 6 Probo faces for *"cuteness"* (dark blue) and *"elephant resemblance"* (light blue). Error bars indicate SD.

Fig. 2. Evolution after 5 iterations and a population of 15 individuals

is later imported into *CATIA V5-6R2014 Student Edition* and, using a custom-made Macro, generates the new population from the parametric shape of the designed robot (see figure 2a). In this case study, we performed one-point cross-overs [3] to generate children. Additionally, mutations were added to increase variability: for each child, one variable of the genotype is randomly modified in such way that $x_i = x_{i-1} + \alpha * rd$. Where α is the mutation rate, and rd is a random number between -1 and $+1$.

We performed GAs for five iterations with populations of 15 individuals. For each iteration, the five best individuals were selected to reproduce and mutate. Selection was performed by the author according to two criteria: 1–selecting the cutest individuals. 2–selecting the individuals that look the most similar to an elephant. These two criteria come from the requirement for the new Probo design: we want to achieve a cute attractive robot that still looks like an elephant. Of course, other criteria could be used, depending on the application.

An online questionnaire was conducted using *LimeSurvey* with 31 participants that had no prior experience with social robots. Participants were presented 6 pictures of Probo appearances in a randomized order. Figure 2 shows the picture used: Figure 2a is the original Probo while Figure 2b through 2f are the evolved appearances selected at the end of the process. For each picture, participants had to answer two questions using

a Likert scale between 1 and 10. The questions were : *Is this individual cute?* (1 being not at all and 10 being very cute) and *Does this individual looklike an elephant?*.

A repeated measures ANOVA with contrast was performed using R (see Figure 2g). It shown a significant increase in cuteness for the five evolved individuals compared to the original, $F(1,180)=5.628$, $p=.019$. Similarly, the five evolved individuals where looking closer to an elephant compare to the original, $F(1,180)=5.418$, $p=.021$.

IV. DISCUSSION AND FUTURE WORK

We have presented an innovative method for collecting and using human feedback for improving robot appearance, helping the designer to optimize the robots proportions to target end-user needs. A case study suggests that significant improvements can already be obtained with few iterations and small populations. Yet, there are limitations that we would like to overcome in future work. First, we would like to perform evaluations with different end-user types. We would also like to generate more iterations with bigger populations. In addition, we will need to create measurement procedures for quantifying convergences and evolutions of the robot appearance.

We believe that the design of robots could and should be adapted with novel manufacturing possibilities. We are currently working on a new low-cost version of Probo that will mostly be produced by additive manufacturing (3D printing). This robot will be parametrized in CAD, allowing its appearance to change rapidly while maintaining mechanical functionality. In this context, we would like to apply this method toward different end-users. The same design will evolve to adapt to different groups. For example, we could imagine fitting the robot to meet the expectations of the elderly [7] or children [8]. The goal would then be to produce theses robots. We want to show that one design can create many robots adapted to each situation or end-user, enhancing stronger HRI.

REFERENCES

[1] I. Leite, C. Martinho, and A. Paiva, "Social robots for long-term interaction: a survey," *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013.

[2] C. F. DiSalvo, F. Gemperle, J. Forlizzi, and S. Kiesler, "All robots are not created equal: the design and perception of humanoid robot heads," in *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, 2002, pp. 321–326.

[3] G. Renner and A. Ekárt, "Genetic algorithms in computer aided design," *Computer-Aided Design*, vol. 35, no. 8, pp. 709–726, 2003.

[4] M. Lewis, "Aesthetic evolutionary design with data flow networks," in *in Proc. Generative Art*. Citeseer, 2000.

[5] J. A. Serpell, "Anthropomorphism and anthropomorphic selection beyond the cute response," *Society & Animals*, vol. 10, no. 4, pp. 437–454, 2002.

[6] 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.

[7] Y.-H. Wu, C. Fassert, and A.-S. Rigaud, "Designing robots for the elderly: appearance issue and beyond," *Archives of gerontology and geriatrics*, vol. 54, no. 1, pp. 121–126, 2012.

[8] A. Peca, R. Simut, S. Pintea, C. Costescu, and B. Vanderborght, "How do typically developing children and children with autism perceive different social robots?" *Computers in Human Behavior*, 2014.

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.
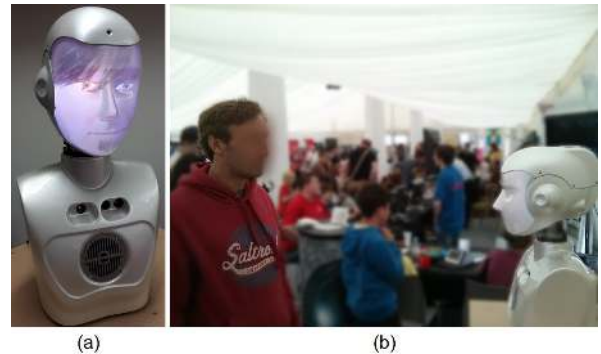

Fig. 1. (a) The robot used in the experiment with a retro-projected face; (b) the robot set up near the entrance of the event space to maximise potential interactions with the public.

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).
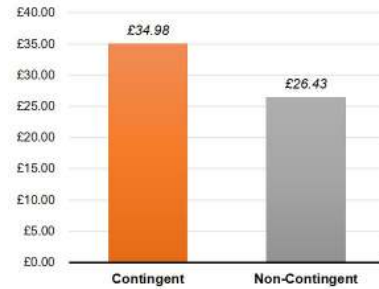

Fig. 2. Comparison of money collected in the two conditions, demonstrating a 32.2% increase for the socially contingent condition over the control.

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.