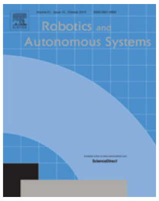
Robotics and Autonomous Systems 83 (2016) 32–43

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



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, Bram Vanderborght

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

HIGHLIGHTS

- The proposed method can be used to generate gestures for an arbitrary social robot.
- This paper focuses on how reaching and pointing gestures are calculated.
- DH-parameters, orientation of the base frames and joint limits are used as input.
- Joint angles are calculated using IK with a cost-function for natural postures.
- The method was validated on several robots, including NAO, Justin and ASIMO.

ARTICLE INFO

Article history:
Received 4 November 2015
Accepted 14 June 2016
Available online 22 June 2016

Keywords:
Generic gesture system
Pointing
Gestures
Upper body postures

ABSTRACT

Since the implementation of gestures for a certain robot generally involves the use of specific information about its morphology, these gestures are not easily transferable to other robots. To cope with this problem, we proposed a generic method to generate gestures, constructed independently of any configuration and therefore useable 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.

1. Introduction

In today's robotics, motions are mostly preprogrammed off-line for a specific robot configuration [1–3], or generated by mapping motion capture data to the robot's configuration [4–6]. Since both techniques use specific information about the robot's morphology, these motions cannot be easily transferred to other robots. This issue is known as the correspondence problem [7,8]. 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. Such a generic gesture system can be useful for different research teams investigating different topics of human–robot interaction, since it allows a fast and easy switch between robot platforms. This work fits in the challenge of the EU-project DREAM of building a complete platform-independent cognitive architecture, which will allow to extend this flexibility of changing between robot platforms for a complete experimental protocol.

An alternative technique to generate gestures in a flexible way was proposed by Stanton et al. [9], by using neural networks to teleoperate a humanoid robot without an explicit kinematic modelling. However, this technique requires training while the method proposed here is very straightforward in use. In both [10,11], 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. For a specified gesture, the end

[*] Corresponding author.
E-mail address: Greet.Van.de.Perre@vub.ac.be (G. Van de Perre).

http://dx.doi.org/10.1016/j.robot.2016.06.006
0921-8890/© 2016 Elsevier B.V. All rights reserved.

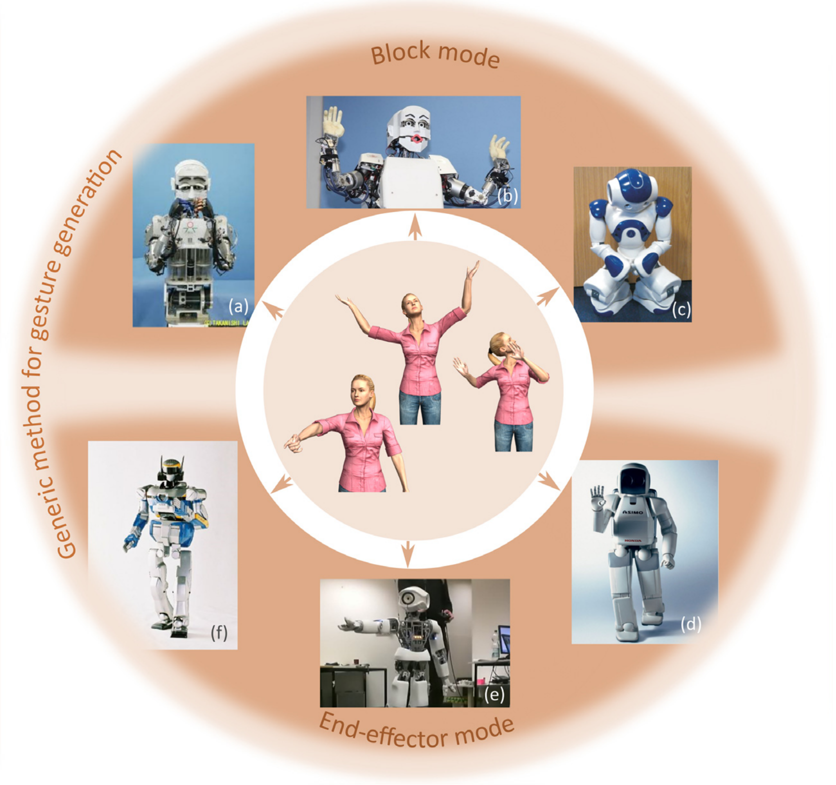


Fig. 1. In the state of the art, gestures are implemented for a specific robot. We propose to use a generic method to generate gestures for different robots. The method uses a human base model to store target gestures independently of any configuration in a database, and to calculate a mapping at runtime, 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 [3], (c) NAO [16], (d) ASIMO [17], (e) Myon [18], (f) HRP-2 [19].

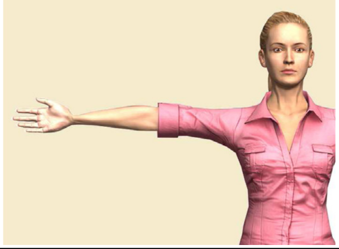
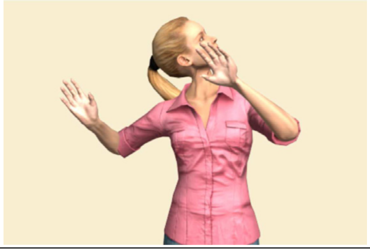
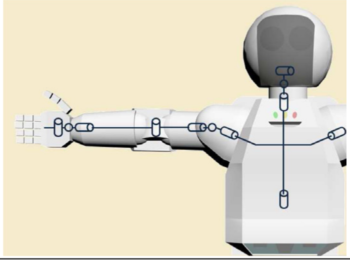
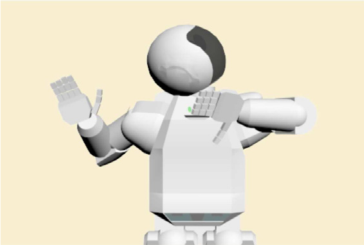
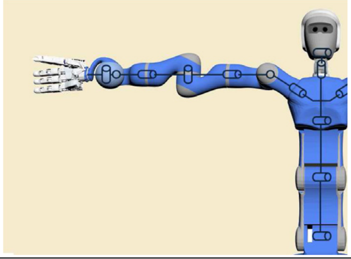
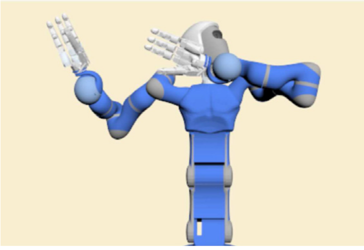
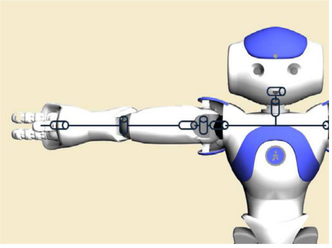
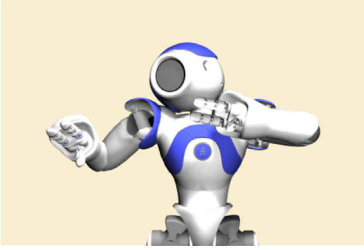
effector positions and orientations are calculated by the MAX system and used as input for ASIMO's whole body motion controller [12]. Similarly, in [11], gestures are described independently of the embodiment by specifying features as the hand shape, wrist position and palm orientation. The specifications 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, [13] presented a semi-general approach for 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 [14] 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 and 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 (Fig. 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 [20] 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 unrecognizable 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 consists 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 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 Fig. 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

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		

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 a set of robots with significant differences in morphology to illustrate the flexibility of the method. 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 [21], Justin [22] and NAO [23], 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 [20].

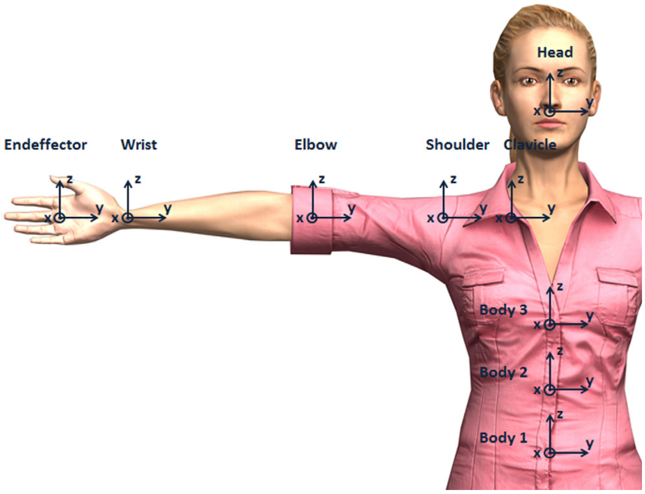


Fig. 2. A reference frame was assigned to each block. For the body 1 block, the reference frame is the standard reference frame. The body 2 and body 3 axes are respectively, the body 1 and body 2 embedded axes. The head and clavicle's reference axes are the body 3-embedded axes. For all other blocks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

Table 2
Minimum values for the joint cost functions. The angles are defined in the reference frames connected to the human base model (see Fig. 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

1.2. End-effector mode

This paper will focus on the novel developed 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.

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 Fig. 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 [24]:

$$\dot{q} = J_A^{\dagger}(q)\,(\dot{x_d} + K\,(x_d - x_e)) + \left(I - J_A^{\dagger}(q)J_A(q)\right)\dot{q_0}. \tag{1}$$

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 $(I - J_A^{\dagger}(q)J_A(q))$ will differ from zero, activating the influence of $\dot{q_0}$ on the calculated joint speeds. $\dot{q_0}$ introduces the cost function $w(q)$ (see Section 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 [25].

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 [26], jerk [27], angular displacement (MAD) [28] or torque [29,30,26]. Another possibility is to use the joint range availability (JRA) criterion [31]. Here, the algorithm will try to find an optimal humanlike posture by keeping the joints close to their central position, away from their limits [32]:

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

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

Cruse et al. [33] 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 was 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 centre values:

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

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

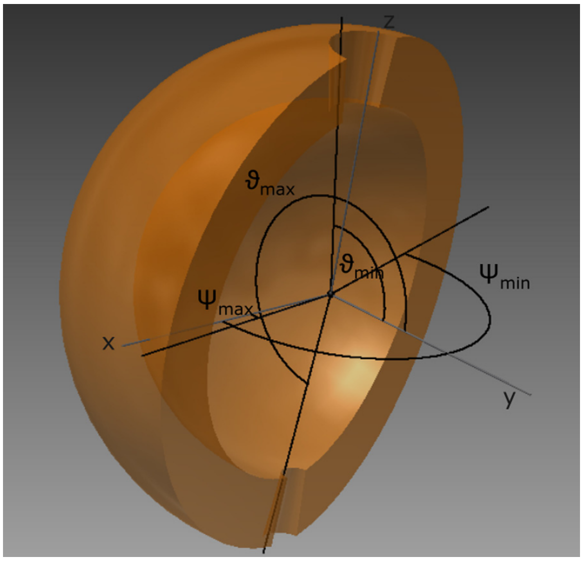


Fig. 3. Example of an approximated workspace.

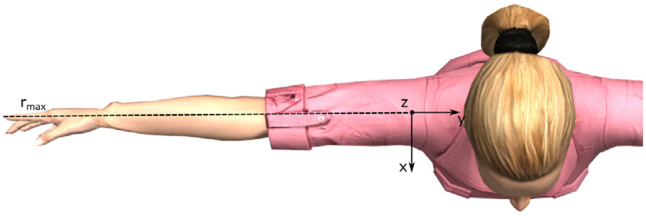


Fig. 4. Determination of the maximum radius $r_{\max}$: the maximum reachable distance of the end effector, measured from the shoulder base frame origin.

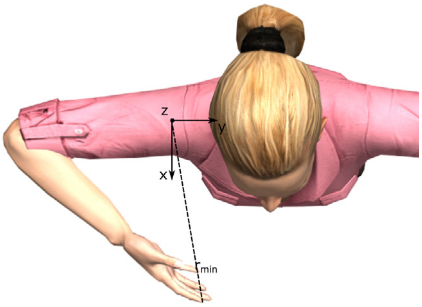


Fig. 5. Determination of the minimum radius $r_{\min}$: the elbow joint is placed in maximum flexion while the other joint angles correspond to the T-pose angles. The distance between the shoulder base frame and end-effector corresponds to $r_{\min}$.

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 maximum and minimum values for the three spherical coordinates specifying the sphere part. Fig. 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, maximum and minimum values are 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 Fig. 4). In 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 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 (Fig. 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 (Fig. 6). In a similar way, the minimum and maximum values for the azimuthal angle ϕ is calculated by considering the maximum horizontal extension 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 radii 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 need to fall out of the workspace (see Fig. 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.

Fig. 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 in the left part of Fig. 8 result from not taking into account the configurations involving elbow flexion for maximum horizontal shoulder flexion/extension. The eliminated circle around the elbow is clearly visible in the right bottom corner of the *xy*-cross section (left of Fig. 8). However, some blue dots are visible in this region. They originate from non-human like postures and do not contribute to proper natural trajectories. In the right part of Fig. 8, the posture of NAO reaching maximum extension is superimposed on the *xz*-cross section passing the shoulder base frame. For the specific configuration and joint limits of NAO, a small region of reachable points, highlighted by a black circle, is neglected by the approximate calculation. These are however points that are not of great interest for pointing and reaching gestures; most probably, such a gesture is directed towards the space in front of the body. Other uncovered blue dots result from configurations involving elbow flexion for maximum flexion/extension and do not contribute to natural human postures.

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

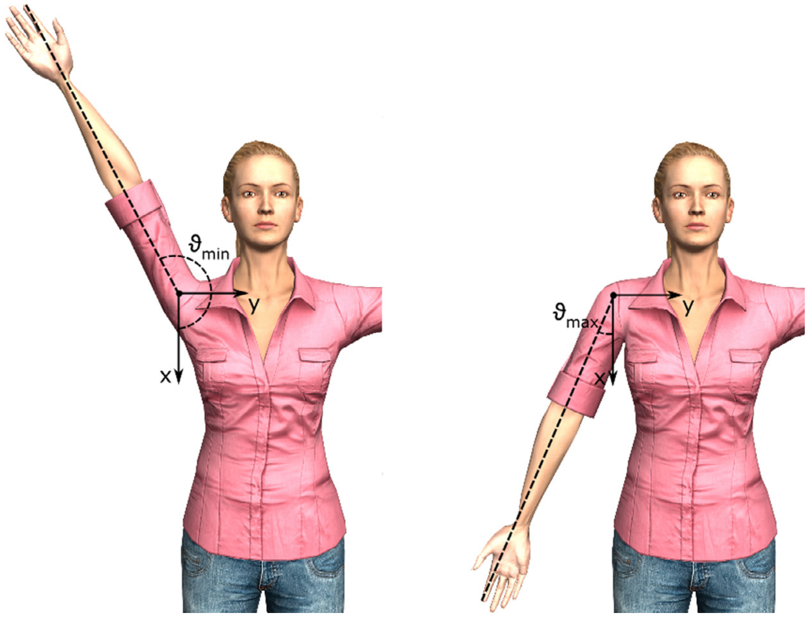


Fig. 6. Calculation of the minimum and maximum values for θ; the arm is placed in respectively, maximum abduction and maximum adduction and the angle formed by the end-effector is calculated.

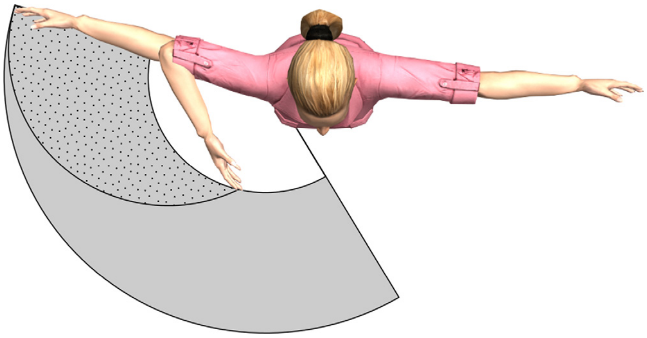


Fig. 7. View of the calculated workspace in the horizontal plane crossing the shoulder base frame. The area covered by a circle with the centre point located in the elbow base frame and radius equal to the length of the lower arm (dotted surface) needs to be subtracted from the workspace (grey surface). N.b.: this is the calculated workspace used in the pointing-condition, where the end-effector is the finger.

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 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. Different research has shown that, when asked

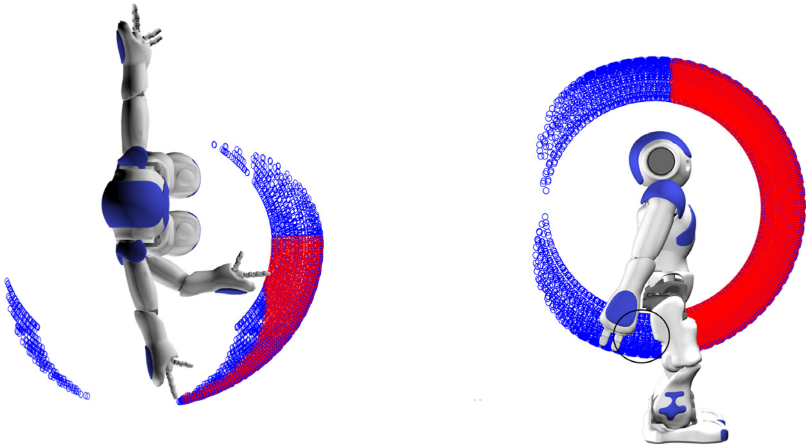


Fig. 8. Calculating an approximate workspace. Blue dots indicate the real workspace, red dots the approximation. Left: xy-cross section passing the shoulder base frame. Uncovered blue dots in the top right result from not taking into account the configurations involving elbow flexion for maximum horizontal 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) originate from non-human like postures and therefore do not contribute to proper natural postures. Right: xz-cross section passing the shoulder base frame. A small region of reachable points, highlighted by a black circle, is neglected by the approximate calculation. Other uncovered blue dots result from configurations involving elbow flexion for maximum flexion/extension and do not contribute to natural human postures. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

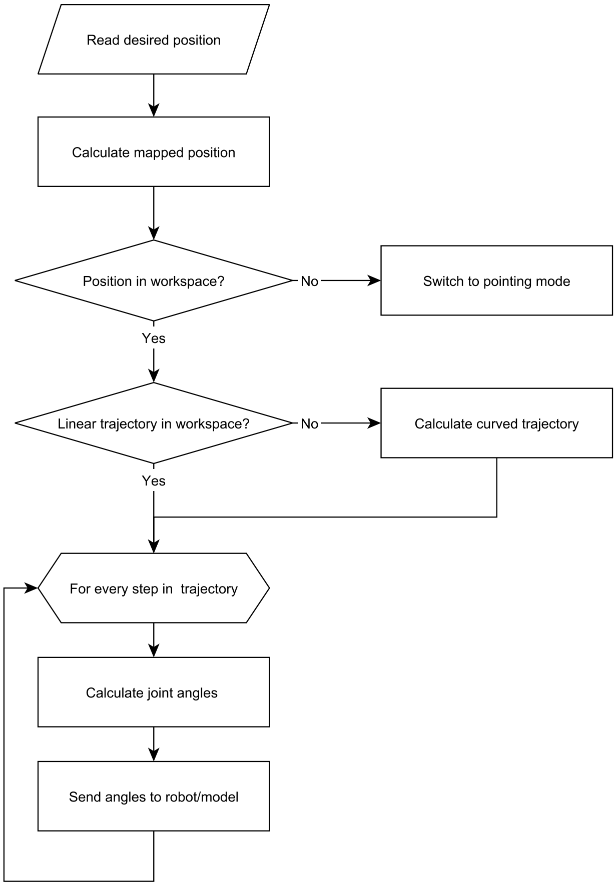

Read desired position
Calculate mapped position
Position in workspace?
No
Switch to pointing mode
Yes
Linear trajectory in workspace?
No
Calculate curved trajectory
Yes
For every step in trajectory
Calculate joint angles
Send angles to robot/model

Fig. 9. Schematic representation of the work flow for the place-at condition.

to perform a point to point hand gesture, humans tend to move their hand along a straight path [34–37]. Therefore, a logical first trial for the path is a linear interpolation between the start and end positions. However, even if the start and end points fall within the workspace, it is possible that a part of the trajectory falls out the reachable range. For a human, this is for example the case when reaching a point close to the chest, starting from the T-pose. [38] reported that, when a subject was required to produce curved hand trajectories, a curve with low curvature elements was tried to be approximated. To verify if a linear trajectory is possible, our developed method checks a set of points on the trajectory to lie in the workspace. In case one of these points falls out of the range, a curved trajectory to reach the desired location is calculated. This trajectory consists of a circular arc connecting the start and end positions of the end-effector, whereof the exact shape, i.e. the radius and mid-point corresponding to the circular arc, depends on which amount the straight path is situated in the non-reachable zone. If only a small section of the straight path is not reachable by the configuration, a small correction with respect to the linear trajectory is possible. However, when a large portion of the straight path falls in the non-reachable zone, to be able to reach the desired end position, the corresponding arm chain needs to go around this zone, resulting in a path with higher curvature. In that way, a trajectory with a minimal amount of curvature is calculated for the specified gesture, as close as possible to the linear path.

Fig. 9 summarizes 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

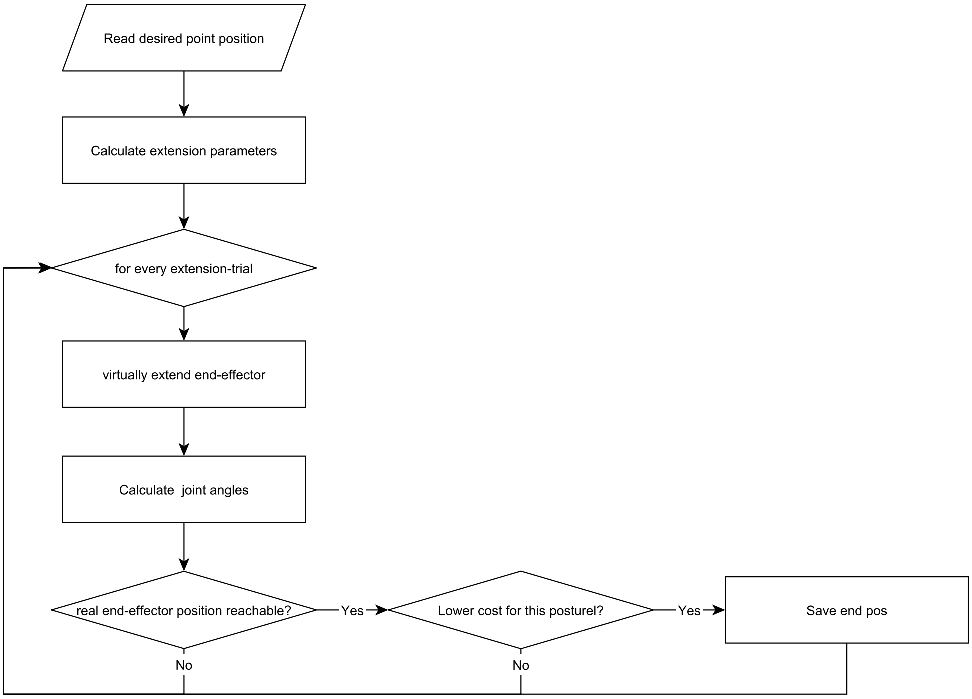

Read desired point position
Calculate extension parameters
for every extension-trial
virtually extend end-effector
Calculate joint angles
real end-effector position reachable?
Yes
Lower cost for this posturel?
Yes
Save end pos
No
No

Fig. 10. Schematic representation of the work flow for the pointing condition.

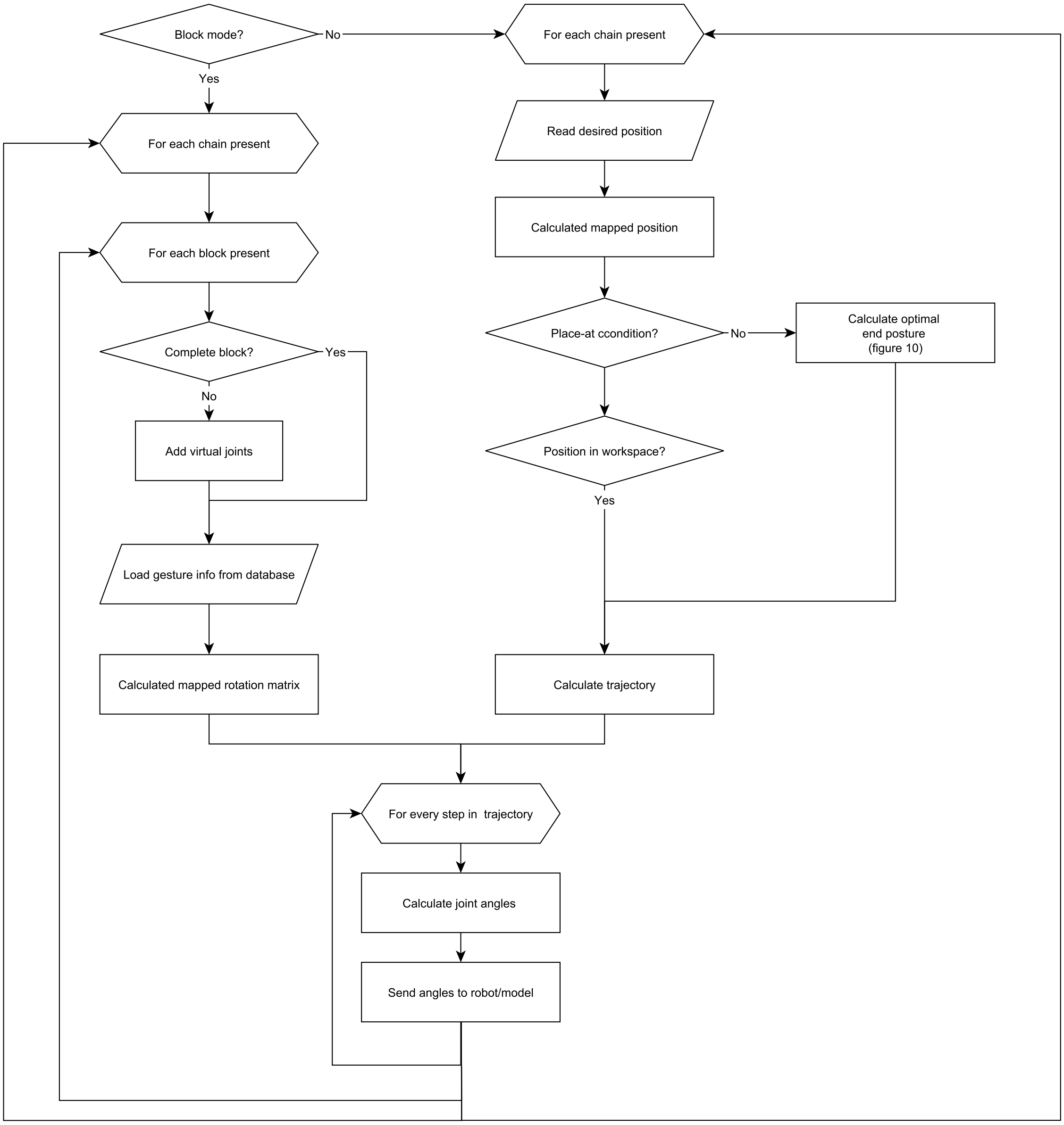


Fig. 11. Simplified work flow of the complete method.

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 in 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. Fig. 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.

A simplified diagram of the complete work flow of the program is visualized in Fig. 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

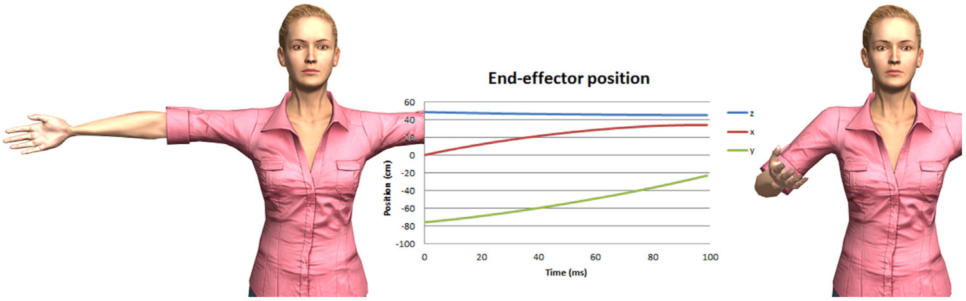


Fig. 12. Calculated trajectory for a place-at task for the right hand with a start position of (0, 76, 48) cm and end position of (34, −23, 45) cm. Left: start pose. Right: end pose. Middle: plot of the calculated end-effector position with respect to the time.

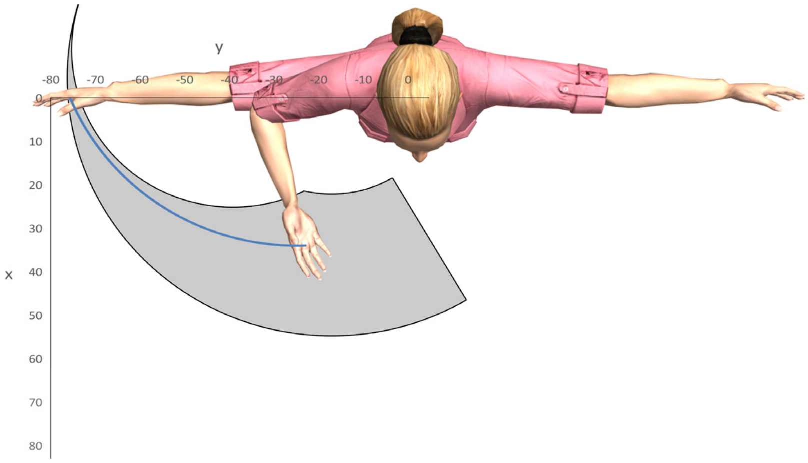


Fig. 13. Top view of the calculated trajectory for a place-at task for the right hand with a start position of (0, 76, 48) cm and end position of (34, −23, 45) cm, superposed on the *xy*-cross section of the right arm workspace. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

arm is shown in Fig. 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 [39]). 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. Fig. 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 positions passes a non-reachable zone. Therefore, a curved trajectory was used to reach the desired end-effector position.

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 [23] with the T-pose as the starting posture. Fig. 14 shows the calculated end posture for the different iteration steps. The end-effector is virtually extended between minimum and maximum values. 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. Fig. 14(a) shows the calculated end posture for the minimum virtual extension whereby the pointing position is visualized by a sphere. Fig. 14(d) visualizes the end posture for the maximum virtual extension, while Fig. 14(b) and (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 [21]. 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 [23]. 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.

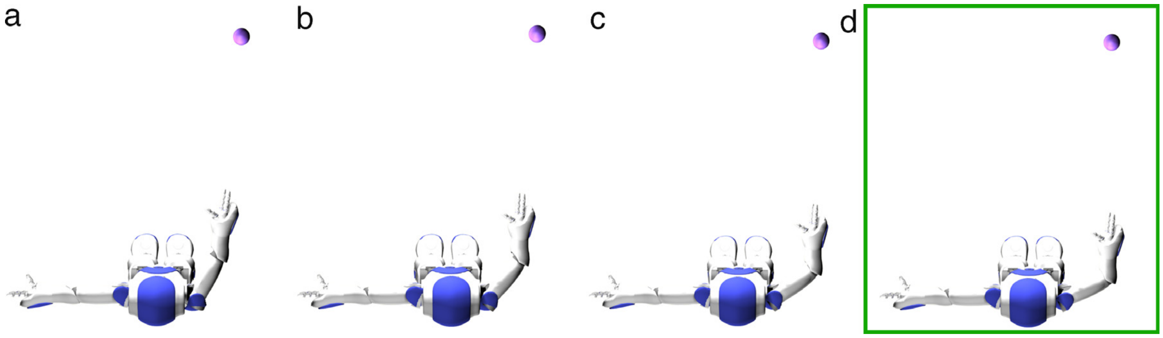
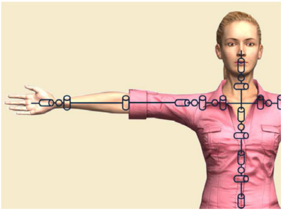
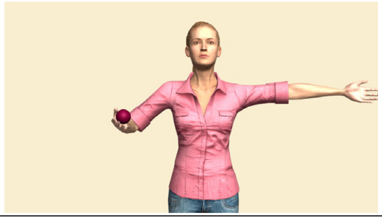
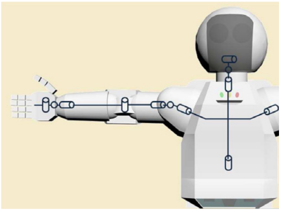
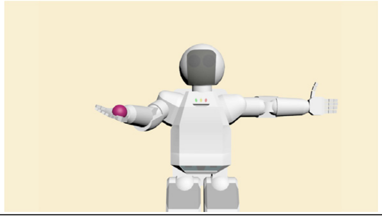
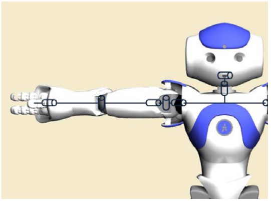
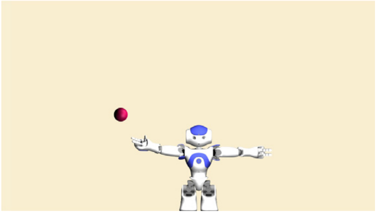


Fig. 14. In the pointing condition, an optimal posture corresponding to a desired pointing position is determined by extending the end-effector gradually between two predefined boundaries, calculating the corresponding end postures, and selecting the optimal posture according to the cost-function. (a) 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 which 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 [20]. 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).

Acknowledgements

The first author is funded by the Fund for Scientific Research (FWO) Flanders (11F2315N). This work is partially funded by the EU-project DREAM (611391). The authors would like to thank DLR for sharing the virtual model of Justin.

References

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

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

[3] M. Zecca, Y. Mizoguchi, K. Endo, F. Iida, Y. Kawabata, N. Endo, K. Itoh, A. Takanishi, Whole body emotion expressions for kobian humanoid robot: preliminary experiments with different emotional patterns, in: The 18th IEEE Int. Symp. on Robot and Human Interactive Communication. RO-MAN 2009, 2009, pp. 381–386.

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

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

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

[7] K. Dautenhahn, C.L. Nehaniv, The Correspondence Problem, MIT Press, 2002.

[8] A. Alissandrakis, C.L. Nehaniv, K. Dautenhahn, Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments, IEEE Trans. Syst. Man. Cybern. A 32 (4) (2002) 482–496.

[9] C. Stanton, A. Bogdanovych, 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.

[10] M. Salem, S. Kopp, I. Wachsmuth, F. Joublin, Generating multi-modal robot behavior based on a virtual agent framework, in: Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots, ICAIR, 2010.

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

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

[13] Y. Koga, K. Kondo, J. Kuffner, J.-C. Latombe, Planning motions with intentions, in: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1994, pp. 395–408.

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

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

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

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

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

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

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

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

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

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

[24] L. Sciavicco, Robotics: Modelling, Planning and Control, Springer, 2009.

[25] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, Vol. 61, SIAM, 1998.

[26] A. Biess, D.G. Liebermann, T. Flash, A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics, J. Neurosci. 27 (48) (2007) 13045–13064.

[27] J. Zhao, B. Xie, C. Song, Generating human-like movements for robotic arms, Mech. Mach. Theory 81 (2014) 107–128.

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

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

[30] J.M. Hollerbach, K.C. Suh, Redundancy resolution of manipulators through torque optimization, IEEE J. Robot. Autom. 3 (4) (1987) 308–316.

[31] E.S. Jung, D. Kee, M.K. Chung, Upper body reach posture prediction for ergonomic evaluation models, Int. J. Ind. Ergon. 16 (2) (1995) 95–107.

[32] C.A. Klein, B.E. Blaho, Dexterity measures for the design and control of kinematically redundant manipulators, Int. J. Robot. Res. 6 (2) (1987) 72–83.

[33] H. Cruse, M. Brüwer, J. Dean, Control of three-and four-joint arm movement: Strategies for a manipulator with redundant degrees of freedom, J. Mot. Behav. 25 (3) (1993) 131–139.

[34] N.A. Bernstein, The co-ordination and regulation of movements, 1967.

[35] J. Kelso, D.L. Southard, D. Goodman, On the nature of human interlimb coordination, Science 203 (4384) (1979) 1029–1031.

[36] P. Morasso, Spatial control of arm movements, Exp. Brain Res. 42 (2) (1981) 223–227.

[37] C.G. Atkeson, J.M. Hollerbach, Kinematic features of unrestrained vertical arm movements, J. Neurosci. 5 (9) (1985) 2318–2330.

[38] W. Abend, E. Bizzi, P. Morasso, Human arm trajectory formation, Brain: J. Neurol. 105 (Pt 2) (1982) 331–348.

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

Greet Van de Perre studied Mechanical Engineering at the Vrije Universiteit Brussel. After graduating in 2011, she started her Ph.D. at the Robotics & 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.

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 Universite Libre de Bruxelles. He is now pursuing a Master in neuropsychology. His research interest focus on the development of artificial intelligence for social robots.

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

Pablo Gómez Esteban received, in March 2014 from the Rey Juan Carlos University of Madrid, a Ph.D. degree for his thesis entitled "Cooperation and Competition in Emotional Robot Societies" with highest distinction. He is currently a post-doctoral researcher working at Vrije Universiteit Brussel in Belgium. His research was addressed towards decision making processes within competitive and cooperative environments; the impact that affective factors have in such processes; and designing and implementing those models to be used in non-expensive platforms within domestic environments. But his research interests also lie on cognitive architectures for social robots and human–robot interaction. He is the author or co-author of several chapters in books and published articles in international journals and conferences. He has also been a reviewer for journals and conferences including Frontiers in Psychology, Autonomous Robots and ICSR 2015.

Dirk Lefeber was born in 1956. He received a degree in Civil Engineering in 1979 and a Ph.D. in Applied Sciences in 1986 from the Vrije Universiteit Brussel. Currently he is full professor, former head of the dept. of Mechanical Engineering and head of the Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, which he founded in 1990. At that time the research focused on dynamical simulations of walking, hopping and running of mono-, bi- and quadruped robots. From the beginning, the group had a strong interest in adaptable compliant actuators. The group invented 2 different adaptable compliant actuators: the pleated pneumatic artificial muscle and the Maccepa actuator. With these they have built the walking robots Lucy and Veronica. In the past 7 years the main research focus moved to physical and cognitive human–robot interaction. The group developed the AMP-foot, an actuated transtibial prosthesis and the Hekta, a passive transfemoral prosthesis. The ALTACRO-rehabilitation robot is a 12 active-DOF lower-limb rehabilitation exoskeleton. Probo is a social robot with 19 DOF in the head, allowing for emotional interaction with people. Currently the R&MM group is partner in 5 FP-7 projects.

The research interests of the group are new actuators with adaptable compliance, dynamically balanced robots, robot assistants, rehabilitation robotics and multibody dynamics, all in the context of physical and cognitive human–robot interaction with emphasis on improving the quality of life of human.

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