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/

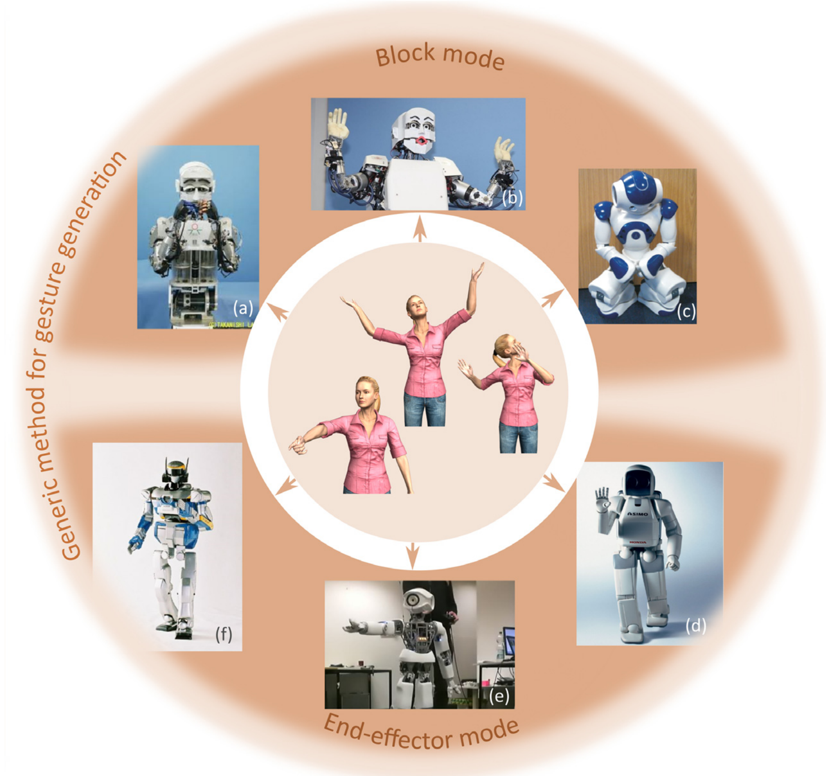

Fig. 1. In the state of the art, gestures are implemented for a specific robot. We propose to use a generic method to generate gestures for different robots. The method uses a human base model to store target gestures independently of any configuration in a database, and to calculate a mapping at runtime, 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