

5th CIRP Global Web Conference Research and Innovation for Future Production

High level robot programming using body and hand gestures

Panagiota Tsarouchi^a, Athanasios Athanasatos^a, Sotiris Makris^a, Xenofon Chatzigeorgiou^a, George Chryssolouris^{a*}

^aUniversity of Patras, Department of Mechanical Engineering and Aeronautics, Laboratory for Manufacturing Systems and Automation, 26500, Patras, Greece

* Corresponding author. Tel.: +30-2610-997-264; fax: +30-2610-997-744. E-mail address: xrisol@lms.mech.upatras.gr

Abstract

Robot programming software tools are expected to be more intuitive and user friendly. This paper proposes a method for simplifying industrial robot programming using visual sensors detecting the human motions. A vocabulary of body and hand gestures is defined, allowing the movement of robot in different directions. An external controller application is used for the transformation between human and robot motions. On the robot side, a decoder application is developed translating the human messages into robot motions. The method is integrated within an open communication architecture based on Robot Operating System (ROS), enabling thus the easy extensibility with new functionalities. An automotive industry case study demonstrated the method, including the commanding of a dual arm robot for single and bi-manual motions.

© 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 5th CIRP Global Web Conference Research and Innovation for Future Production

Keywords: Gestures, Programming, Human Robot Interaction, Dual arm robot.

1. Introduction

Robot programming tools are expected to be more user friendly involving the latest trends of artificial intelligent, function blocks, universal programming languages and open source tools, anticipating the traditional robot programming methods [1]. Even non-expert robot programmers will be able to use those through shorter training processes. The intuitive robot programming includes demonstration techniques and instructive systems [2]. Different methods and sensors have been used for implementing more intuitive robot programming frameworks. Some examples include vision, voice, touch/force sensors [3], gloves, turn-rate and acceleration sensors [4], as well as artificial markers [5]. The main challenge faced in this area concerned the development of methods for the sensor knowledge data gathering and reproduction of these data from robots with robustness and accuracy [6].

An extended review on the use of human robot interaction (HRI) techniques was presented in [1]. The main direction of such systems includes both the design and development of multimodal communication frameworks for robot

programming simplification. Among these research studies, a significant research attention has been paid on the use of gestures. For example, hand gestures are recognized by using data gloves in [7-8], where in [4] the sensor Wii is used. Examples using 3D cameras can be found in [9-12]. In [13] the human-robot dialogue is achieved with the help of a dynamic vocabulary. Kinect has been used as the recognition sensor in [14-18] enabling the online HRI. Kinect has also been used for ensuring safety in human robot coexistence [19]. Apart from HRI, the use of Kinect can also be found in studies for human computer interaction (HCI) [20-21]. Last but not least, the leap motion sensor was also selected for HRI [22-23].

Despite the wide research interest on intuitive interfaces for interacting and robot programming, there are still challenges related to the need for universal programming tools, advanced processing, transformation methods and the environment uncertainties. Related to the environment uncertainties, the natural language interactions, such as the gestures and voice commands, are not always applicable to industrial environments, due to their high level of noise, the lightning conditions and the dynamically changing environment.

Additionally, the development of gesture based vocabularies with complex and numerous gestures seems not to be user friendly for the human.

Taking into consideration the above challenges, this paper proposes a high level robot programming method using sensors detecting both body and hand gestures. The defined body gestures are static and the hand gestures are based on the dynamic movements of the hands. The user has two different options for interacting with the robot. There is no need for special robot programming training, even for commanding an industrial robot in this case. Safety can be ensured using the traditional emergency buttons, during the robot programming. The selected sensors for implementing body and hand gestures are the Microsoft Kinect and the leap motion respectively. Kinect sensor has become significantly popular in replacing the traditionally 3D cameras in applications where recognition accuracy is not needed. The introduction of leap motion in robot programming is relatively new allowing the detection of hand gestures for commanding the robot.

2. Method overview

The proposed high level robot programming method is based on the open architecture presented in Figure 1. In this architecture, the high level robot programming is achieved by the implementation of high level commands in the form of a gestures' vocabulary. This vocabulary includes both body and hand gestures, that are recognized using data from two different tracking devices. The first device is an RGB-d camera allowing the recognition of body gestures, while the hand gestures are recognized using the similar device called leap motion. This sensor includes two monochromatic IR cameras and three infrared LEDs.

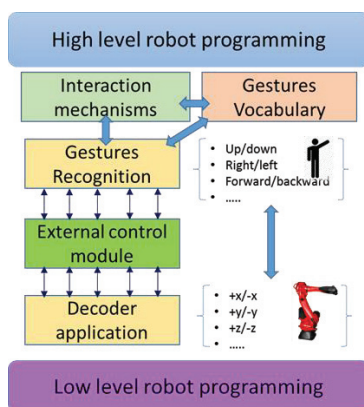


Fig. 1. System architecture.

The available interaction mechanisms (sensors) are represented within a software module (Fig. 1), allowing the detection and recognition of the gestures ('Gestures Vocabulary', 'Gestures Recognition'). The recognized gestures are published in "rostopic" as messages, where third applications can subscribe to listen these messages through ROS middleware. These messages represent the result of recognition in each topic e.g. /Left_gesture. These two modules do not communicate with each other but they are managed by

the external control module. The control of the gestured based programming is managed by this module as well. This module is directly connected to the robot controller, by exchanging messages through TCP/IP protocol. Robot starts the movement that the human has commanded in order to allow the easy programming in tasks that high accuracy is not required.

A body and hand gestures vocabulary has been defined. The body gestures involves human body postures that are able to control the motion of a robot arm in different directions along a user frame. The defined gestures are static and allow to an operator to move the robot in the 6 different directions, namely $+/-x$, $+/-y$ and $+/-z$ around any selected frame (Fig. 2).

Similarly, to the body gestures, hand gestures are defined within this vocabulary (developed as functions using Python). These gestures are dynamic motions of the human hands, involving different number of fingers in each of them in order to secure the uniqueness of each motion. The use of these 6 different hand gestures enables the same motions as the body gestures. The hand gestures are more useful in the case that the human is closer to the robot for interaction of programming purposes.

The movement of the robot in the $+x$ direction is achieved with the up gesture, where the operator extends the right arm upwards and holds the left hand down. The robot will move in the $-x$ with down gesture and it can be done when the left hand is extended upwards and the right one held down. Using the hand gestures, the robot will go up if the operator uses one hand and swipes it upwards contrariwise it will go down if the finger is swiped downwards.

The movement along the $+y$ direction is accomplished with the left gesture which includes the left hand extended at the height of the shoulder pointing to the left direction. On the contrary the robot will move along the $-y$ with the left gesture involving the right hand extended with the same way but pointing to the right direction. Using the hand gestures, the gestures for right and left are recognized using two fingers swiping from left to right and the opposite.

The movement along the $+z$ direction is achieved using the forward gesture, where the right hand is extended upwards and the left hand is at the height of the shoulder pointing to the left direction. The movement in the opposite direction is achieved by using the opposite hands, the left upwards and the right at the height of the shoulder pointing to the right (backward gesture).

The same movements are achieved using the relevant gestures, where three fingers swiping from left to right and the opposite. Finally, in order to stop the robot during a motion, two static body and hand gestures are available. In this way, the robot motor drivers are switched off and the robot will not remain active.

Following the recognition results of the gestures and how these are expected to be executed in the robot controller, communication between the external controller module and the decoder application is established. In this way, the robot receives the human commands, allowing the execution of them, translated to the described motions.

This method is offered for online interaction with an industrial robot, even in the case of multi-arm robotic systems, enabling the coordinated motions execution.

Download English Version:

<https://daneshyari.com/en/article/5470008>

Download Persian Version:

<https://daneshyari.com/article/5470008>

[Daneshyari.com](https://daneshyari.com)