Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-11 (2018) 66–71

Survey on Human-Robot Interaction for Robot Programming in Industrial Applications

Valeria Villani, Fabio Pini, Francesco Leali, Cristian Secchi, Cesare Fantuzzi

University of Modena and Reggio Emilia, Modena, Italy.
(e-mail: name.surname@unimore.it)

Abstract: The recent trends in modern industry highlight an increasing use of robots for a wide range of applications, which span from established manufacturing operations to novel tasks characterized by a close collaboration with the operators. Although human-robot collaboration allows to relieve operators of exhausting works, an effective collaboration requires a straightforward interaction to foster the use of robot assistants. This paper provides a comprehensive survey on human-robot interaction approaches and related interfaces addressed to robot programming. An overview of on-line and off-line robot programming techniques is first presented. Then, novel intuitive interaction means, such as those based on multi-modal interaction, virtual and augmented reality, are considered. The paper aims at pointing out that collaborative robotics can effectively reduce operator's physical workload if easy to use interfaces for robot programming are provided.

Keywords: Intelligent interfaces, Robotics technology, Human operator support, Multi-modal interaction, Design methodology for HMS.

1. INTRODUCTION

Robots play a pivotal role for today's manufacturing industry to be competitive. The International Federation of Robotics has estimated that until 2019 the worldwide annual supply of industrial robots will increase, on average, of 13% per year (IFR, 2016). The use of industrial robots has been increasing also in small and medium sized companies given the availability of affordable solutions and easy-to-use collaborative robots (IFR, 2016). In this regard, to allow a pervasive diffusion of such robots, they should be quickly and intuitively (and safely) operated by humans.

However, despite the reported positive trend, one of the difficulties of using robots in industrial processes is often related to the way the human operator is supposed to interact with the robot, since it usually requires specialized knowledge. Indeed, in addition to guaranteeing the physical safety of human operators interacting with a collaborative robot, issues related to mental safety, intended as mental stress and anxiety induced by close interaction with robot, need also to be considered. Thus, the availability of intuitive ways to interact with robots and program them is one of the key enablers for letting also small companies make use of (collaborative) robotic technologies. Specifically, simplified ways to interact with industrial robots in a reduced time, while minimizing user's errors and preserving situation awareness, are needed. In this regard, together with several other advantages, collaborative robots used in industrial processes prove beneficial since they allow for intuitive programming approaches. On the contrary, traditional non collaborative robots often need expert specialist engineers to program the robot since instructions to robots have to be explicit and motion oriented, basically specifying a set of points which the robot must pass through.

To reduce mental workload and increase reliability in robotic agents, human-robot interfaces based on the principles of human-centred design and cognitive engineering should be considered (Hancock et al., 2011; Keyes et al., 2010). Accordingly, the design of human-robot interfaces can be enhanced considering human cognitive information processing, decision making, perceiving and other capabilities or limitations (Nachreiner et al., 2006; Sabattini et al., 2017; Villani et al., 2018b). Generally speaking, to let human operators easily interact with robots and take full advantage of their skills, it is important that intuitive user interfaces are properly designed. This requires that, on the one hand, providing inputs to the robot and programming it should be intuitive for the worker so that she/he is less concerned with how to communicate and is free to concentrate on the tasks and goals at hand. On the other hand, the information provided as a feedback by the robot should be adequate to provide the user with situation awareness, needed to comprehend the current system behavior and facilitate intervention in dynamic and unforeseen situations. To enable these features, the use of novel programming approaches, such as walk-through programming or programming by demonstration, and interaction modes, such as gestures or speech, and augmented reality have been introduced to overcome the limitations of traditional interaction approaches (Gupta and Arora, 2009; Bascetta et al., 2013; Pan et al., 2012).

In this paper, building upon our recent extensive review on human-robot collaboration (Villani et al., 2018a), we present the state-of-the art on robot programming, with

10.1016/j.ifacol.2018.08.236

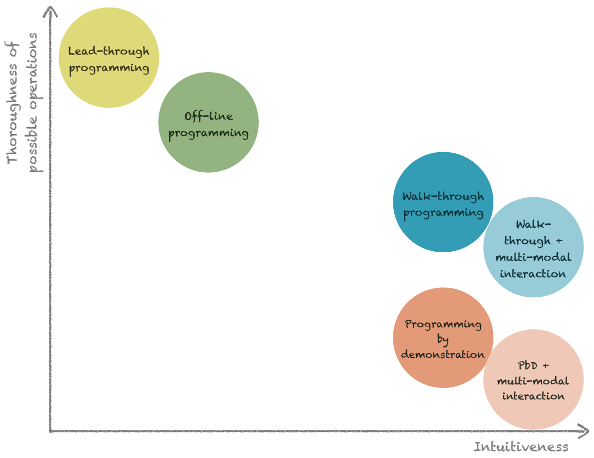

Fig. 1. Comparison of approaches to the programming of industrial robots.

specific focus on industrial applications. Specifically, classical and novel approaches to programming are firstly reviewed, highlighting the main features of traditional lead-through and off-line programming and novel walk-trough programming and programming by demonstration. Then, we discuss how these methods can be further enhanced by introducing natural human-friendly multi-modal interaction modes. Finally, the use of virtual and augmented reality for robot programming is discussed. These methods are compared in Fig. 1 in terms of intuitiveness and ease of use, which are features of the most novel approaches, and completeness of possible operations to perform and working scenarios, which compares favorably for traditional approaches, such as lead-through and off-line programming.

2. ROBOT PROGRAMMING APPROACHES

In practical industrial applications, most of the operator's cognitive effort during the interaction with a robot is devoted to task programming. Differently from instructing a (skilled) human worker how to carry a task, programming a robot requires providing the robot with explicit motion-oriented instructions and detailing the points and trajectories to follow.

2.1 Traditional lead-through programming

The standard approach relies on the use of the teach pendant for on-line moving the robot through the required motion cycle by jogging. Trajectories and endpoints are then recorded into controller memory for later playback. When played back, the end effector appears to follow a continuous smooth path. During the programming session, the robot's control is placed in a "teach" mode and the person performing the teach function can be within the robot's working envelope, with operational safeguarding devices deactivated or inoperative.

Although the concept is simple and does not require strong technical expertise, some programming skills are still required and teaching trajectories to the robot in this way turns out to be a tedious and time-consuming task, as shown in usability assessments reported, e.g., in (Gray et al., 1992; Morley and Syan, 1995). Moreover, it is only suitable for programming simple tasks on workpieces with a simple geometry, with programming complexity dramatically increasing when complex geometries are involved. Further, this method requires reprogramming for each new task, even in case of little changes, thus stopping the production every time. As a consequence, in industry this type of robot programming can be justified economically only for producing large lot sizes and is not suited for small and medium sized enterprises, where small production batches require frequent task reprogramming and such a time-consuming and demanding procedure is unaffordable (Dietz et al., 2012).

To overcome these limitations, several other approaches, which are addressed below, have been proposed. Nevertheless, lead-through programming is still necessary in some specific situations: this is the case, for example, when it is needed to *in situ* verify and manually adjust programs generated off-line or when 3D models are unavailable (see Subsec. 2.2), or still in presence of complex tasks that can be only be programmed by the human operator close to the robot (Hägele et al., 2016; Hein et al., 2008). To this end, a few new programming methods have been proposed to alleviate the burden of jogging assisted by implementing additional sensors and control technologies (Pan et al., 2012; Hein et al., 2008; Hein and Wörn, 2009).

2.2 Off-line programming

Given the disadvantages listed above, nowadays on-line robot programming by teach pendant has become quite unusual and is being replaced by off-line programming (OLP) (Neto and Mendes, 2013). This approach consists in the remote simulation of the task in a 3D model of the complete robot workcell. Tools for OLP come with a set of modelling and simulation functions that allow for graphical representation of the robot cell, automated program generation and simulation of robotic tasks, with the possibility to check for possible collisions (Zha and Du, 2001; Neto and Mendes, 2013). Moreover, most advanced today's OLP tools offer modules for specific processes, such as coating, welding or polishing. After simulation and testing, the program is then exported from the computer to the robot and some final tuning of the program with the teach pendant might be required.

Software tools for OLP can be classified in two main categories (Gan et al., 2013). The first one includes OLP plugins and standalone OLP software, which can manage different robot brands. OLP plugins are based on existing software for computer aided design or manufacturing (CAD/CAM), and rely on the modelling functionalities of the hosting CAD/CAM tool. On the contrary, standalone OLP software tools have their own graphical interface and specific modelling capabilities. For both cases, these tools exploit the kinematic model of the considered robot and perform basic analysis, such as positions reachability, definition of the robot movements and collision assessment. They also provide the robot programs through a post-processor based on the coding language of the selected robot brand. The second category includes OLP proprietary software tools, which are developed directly by the robot manufacturers. These tools are based on a virtual replica of the robot controller and return a realistic robot simulation with an accurate calculation of robot trajectories and processing times, as well as the native robot code.

OLP approaches move the burden of programming from the robot operator in the shop floor to the software engineer in the office (Pan et al., 2012). Time required to program the robot is still remarkably long, but the production does not need to be stopped during programming, thus the uptime can be maximized. Moreover, it is fundamental to perform a robot calibration step when off-line generated program is transferred onboard the robot in order to compensate for any positioning error due to a mismatch of coordinate systems between real and virtual world. The calibration of robotic solutions to exploit robot off-line programmed codes aims to align both the position of contact point of the robot tool, also known as tool center point (TCP) and the origin of the part being processed, also defined as work frame (WF). The traditional calibration approach relies on manual teaching of both TCP and WF using sharp tips and calibration algorithms. To improve the accuracy of the calibration process and consequently reduce alignment errors, automatic methods have been proposed, which exploit measurement tools, such as touch probes or laser trackers. Automatic calibration approaches for off-line programmed robotic finishing workcells for deburring and machining process have been presented in (Leali et al., 2013a,b). Further improvements to calibration operations with a behavioral simulation of the measurement tool directly in the OLP virtual environment have been presented in (Pini et al., 2014).

2.3 Walk-through programming

This approach consists in letting the user physically move the end-effector of the robot through the desired positions in a free way. In the meantime, the robot's controller records the trajectory and the corresponding joints coordinates, and is then able to reproduce the trajectory. Thus, the robot can be programmed in a very intuitive manner and no knowledge of the programming language is requested to the operator. Clearly, in this scenario safety issues related to physical human-robot interaction (HRI) become of paramount importance and appropriate motion control strategies are needed (Gupta and Arora, 2009). Most control approaches rely on the use of a force/torque sensor typically mounted on the robot wrist to measure forces and torques during the interaction. These are then exploited as inputs to the control system of the robot, in order to accommodate the forces applied by the operator (Ferretti et al., 2009). This is typically achieved by means of compliant control schemes, such as admittance/impedance control (Bascetta et al., 2013; Talignani Landi et al., 2017b,a).

Recently (Ferraguti et al., 2017), walk-through programming for spray painting with industrial collaborative robots has been proposed. Moreover, walk-through programming in welding applications has been considered in (Ang Jr et al., 1999), exploiting the impedance control with zero stiffness but without taking into account the tool emulation or compensation. Indeed, the main limitation of these methods is that they require a robust dynamic model of the robot and its tool.

To partially overcome these issues, the concept of virtual tool has been introduced, which gives the operator an impression the closest possible to that felt when the task is performed without the robot assistance (see, e.g., (Ferretti et al., 2009)). However, the use of a virtual tool is not appropriate in some industrial applications, where the operator needs to program the robot by moving directly the real tool, in order to see the final result of the operation. In this case the end-effector of the robot may have to carry a not negligible payload (Aghili and Namvar, 2009).

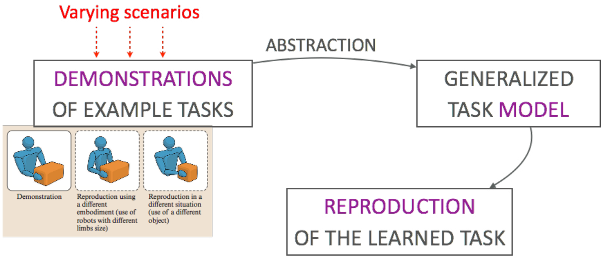

Fig. 2. Overview of robot programming by demonstration.

Other approaches resort to variations of admittance and impedance control (Tee et al., 2010; Ferraguti et al., 2013; Talignani Landi et al., 2017b,a). Finally, some approaches not relying on the use of a force/torque sensor should be mentioned. Basically, they consist in detecting human intentional interaction, which can ultimately be used to achieve manual guidance in a scenario of walk-through programming. Examples can be found in (Geravand et al., 2013; Chen and Kazanzides, 2013).

2.4 Programming by demonstration

An extension to walk-through programming is provided by the concept of programming by demonstration (PbD). While the former allows the mere reproduction of motions performed by the human operator, the latter considers the possibility for the robot to learn the movements to perform under varying conditions and to generalize them in new scenarios, as shown in Fig. 2. To this end, the robot is endowed with some learning skills, rather than pure imitation. This approach allows an easy and natural interaction, without requiring any experience in robot programming, as for walk-through programming.

The most investigated issues in PbD refer to how to generalize across demonstration in the demonstration phase and how to generalize the movement to new situations during reproduction. As regards the first one, mainly two approaches have been proposed to extract the relevant features of a given task, namely symbolic and trajectory encoding. In symbolic encoding, a set of task-dependent primitives is derived and a task is expressed as a sequence of symbolic primitives (Abbas and MacDonald, 2011). In the case of trajectory encoding, the demonstrated trajectory is directly transformed to the robot motion (Ijspeert et al., 2002; Calinon et al., 2007). The choice of which encoding approach to consider strongly depends on the the task to perform.

3. MULTI-MODAL INTERFACES

Using interaction modes that make robots behave like humans or complement human abilities simplifies the communication with the robot and allows people with no previous experience or knowledge in HRI to easily and effectively interact with robots. Multi-modal interfaces resort to human-friendly input modes, such as speech, gesture, eye tracking, facial expression, haptics, that let users control and program a robot by means of high-level behaviors (Tsarouchi et al., 2016; Villani et al., 2017).

3.1 Vision based

Vision systems can be used for recognising the demonstrator's actions and tranferring them to the robot for motion imitation. As an example, in (Solvang et al., 2008) the authors have proposed an approach based on the recognition of marks manually made by the human operator on the workpiece: these marks are detected by a vision system and translated in instructions to the robot for additional processing. In (Bonilla et al., 2012), the path is shown to the robot by using a laser that projects structured light on the surface. A similar approach has been proposed in (Madsen et al., 2007) for teaching welding trajectories to the robot. Structured light 3D machine vision is used also for object profile perception in (Hu et al., 2007), where the problem of automated leather surface roughing has been addressed.

A markerless vision-based HRI system has been proposed in (Du and Zhang, 2015) to control dual robot manipulators by tracking the motion of operator's hands, without any contact devices or markers.

Moreover, stereo vision has been used also in this context: for example, in (Takarics et al., 2008) coordinates for welding robot programming are acquired by means of stereo vision. The system uses two cameras for 3D coordinates and edge detection with other image processing algorithms to find the welding path in the image.

Visual commands are combined with voice commands in (van Delden et al., 2012) to program a pick-and-place application simply by pointing to objects in the work area and speaking simple and intuitive natural language commands. Then, a camera is used to recognize deictic gestures and implement finger pointing.

3.2 Vocal commanding

Voice guidance proves very useful when hands-free interaction is required. That is the case when the user's hands are not free or when classical interaction systems do not fit the situation, such as when interacting with service mobile robots. However, very few systems for speech recognition and natural language processing in industrial scenario exist, mainly due to the lack of reliable solutions and the severe consequences of any misrecognition.

The ultimate goal of speech interfaces is establishing a natural bidirectional communication that allows natural language to be understood and generated by the robot. This poses great challenges that are not yet solved. However, in industrial applications a vocal communication based on quasi-natural language might be sufficient, since the lexicon to be used is quite limited and users should be (at least partially) expert of the interaction. Despite of this, the existing approaches are usually based on a very limited number of simple voice commands (Veiga et al., 2009), which is quite limiting, as reported in (Rogowski, 2012). A web-based remote voice control of robotic cells has been proposed in (Rogowski, 2013) and it is based on quasi-natural language. However, the implementation and validation of the approach are still at a laboratory level. Voice command is used in (van Delden et al., 2012) in combination with finger-pointing commands. Recognized voice commands trigger the vision component to capture what a user is pointing at. In (Chan et al., 2012) the problem of environmental noise in industrial robotic control is considered and a multichannel signal enhancement methodology has been proposed to improve the performance of commercial speech recognizers.

4. AUGMENTED AND VIRTUAL REALITY

In recent years, a lot of interest in robot interfaces has been devoted to the application of augmented reality (AR) and virtual reality (VR) in manufacturing practice. The first results of the integration of these methodologies in traditional interaction approaches have shown that they can increase system productivity while enhancing human safety (Michalos et al., 2016; van Krevelen and Poelman, 2010). AR and VR in industrial applications have been applied to design, assembly and maintenance since they allow to display synoptic information onboard the robot and in the field of vision, such as performance values, catalogue spare part codes, work instructions and optimal operation sequence (Michalos et al., 2016; van Krevelen and Poelman, 2010). Moreover, they have been applied also to robot programming (Ong et al., 2006; Chong et al., 2009). The first attempts in this regard considered VR as an alternative approach to OLP, allowing safe robot programming in a more intuitive manner than traditional OLP. However, approaches based on VR require to extensively model the environment entities and to calibrate the model when it is applied in real environment. To overcome this, robot programming using AR (RPAR) techniques were introduced, which implement a sort of OLP without the need for a model of the workpiece in the virtual environment (Pan et al., 2012). RPAR brings the same advantages as walk-through programming, such as intuitive programming and spatial interaction, and OLP, namely the possibility to run simulations of the planned paths to check for collision and to program the robot without stopping the production (Pan et al., 2012). In addition, RPAR allows the programming of large robots where the walk-through method is unfeasible, such as is the case of airplane washing robots considered in (Chong et al., 2009).

In (Lambrecht and Krüger, 2012) AR is combined with gestures for a very intuitive spatial programming, whereas RPAR is proposed jointly with modular multimodal inputs, such as mouse and speech, in (Akan et al., 2011), and is tested with pick-and-place tasks of different complexity. Finally, in (Stadler et al., 2016) the perceived workload of industrial robot programmers and their task completion time were investigated when using a tablet-based AR approach. The mental demand was found to be decreased with respect when not using AR, but an increase in task completion time was however found.

5. CONCLUSION

Robot programming is a key activity in the development of robotic systems since it allows the success of a complex solution in real manufacturing scenarios. Such success increases as the effort and skills required for robot programming are as limited as possible, while, on the contrary, complex and long programming processes are not effective and increase the time required for each production change. It follows that the development and adoption of for user-friendly robot programming approaches are essential for an effective use of robots for industrial applications. The present work provides an extensive analysis of the current scenario of robot programming interfaces. On the one hand, traditional interfaces such as lead-through and off-line programming are still based on coding of required robot movement instructions, respectively on real and virtual robots. On the other hand, novel robot programming approaches, as walk-through and programming by demonstration, aim at freeing user from coding. Furthermore, multi-modal interfaces and the integration of both virtual and augmented reality promise to further simplify robot programming.

REFERENCES

(2016). Executive summary world robotics 2016 industrial robots. Technical report, International Federation of Robotics (IFR).

Abbas, T. and MacDonald, B.A. (2011). Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 3816–3821. IEEE.

Aghili, F. and Namvar, M. (2009). Scaling inertia properties of a manipulator payload for 0-g emulation of spacecraft. *The Int. J. Robotics Research*, 28(7), 883–894.

Akan, B., Ameri, A., Cürüklü, B., and Asplund, L. (2011). Intuitive industrial robot programming through incremental multimodal language and augmented reality. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 3934–3939. IEEE.

Ang Jr, M.H., Lin, W., and Lim, S.Y. (1999). A walk-through programmed robot for welding in shipyards. *Industrial Robot: An Int. J.*, 26(5), 377–388.

Bascetta, L., Ferretti, G., Magnani, G., and Rocco, P. (2013). Walk-through programming for robotic manipulators based on admittance control. *Robotica*, 31(7), 1143–1153.

Bonilla, I., Mendoza, M., Gonzalez-Galvan, E.J., Chavez-Olivares, C., Loredo-Flores, A., and Reyes, F. (2012). Path-tracking maneuvers with industrial robot manipulators using uncalibrated vision and impedance control. *IEEE Tran. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1716–1729.

Calinon, S., Guenter, F., and Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2), 286–298.

Chan, K.Y., Yiu, C.K., Dillon, T.S., Nordholm, S., and Ling, S.H. (2012). Enhancement of speech recognitions for control automation using an intelligent particle swarm optimization. *IEEE Trans. Industrial Informatics*, 8(4), 869–879.

Chen, Z. and Kazanzides, P. (2013). Force control of a non-backdrivable robot without a force sensor. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 3570–3575. IEEE.

Chong, J.W.S., Ong, S., Nee, A.Y., and Youcef-Youmi, K. (2009). Robot programming using augmented reality: An interactive method for planning collision-free paths. *Robotics and Computer-Integrated Manufacturing*, 25(3), 689–701.

Dietz, T., Schneider, U., Barho, M., Oberer-Treitz, S., Drust, M., Hollmann, R., and Hägele, M. (2012). Programming system for efficient use of industrial robots for deburring in SME environments. In *7th German Conf. Robotics (ROBOTIK)*, 1–6. VDE.

Du, G. and Zhang, P. (2015). A markerless human–robot interface using particle filter and Kalman filter for dual robots. *IEEE Trans. Industrial Electronics*, 62(4), 2257–2264.

Ferraguti, F., Secchi, C., and Fantuzzi, C. (2013). A tank-based approach to impedance control with variable stiffness. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 4948–4953. IEEE.

Ferraguti, F., Talignani Landi, C., Secchi, C., Nolli, M., Pesamosca, M., and Fantuzzi, C. (2017). Walk-through programming for industrial applications. In *27th Int. Conf. Flexible Automation and Intelligent Manufacturing (FAIM)*.

Ferretti, G., Magnani, G., and Rocco, P. (2009). Assigning virtual tool dynamics to an industrial robot through an admittance controller. In *IEEE Int. Conf. Advanced Robotics (ICAR)*, 1–6. IEEE.

Gan, Y., Dai, X., and Li, D. (2013). Off-line programming techniques for multirobot cooperation system. *Int. J. Advanced Robotic Systems*, 10.

Geravand, M., Flacco, F., and De Luca, A. (2013). Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 4000–4007. IEEE.

Gray, S., Wilson, J., and Syan, C. (1992). Human control of robot motion: orientation, perception and compatibility. In *Human-Robot Interaction*, 48–64. Taylor & Francis, London.

Gupta, A. and Arora, S. (2009). *Industrial automation and robotics*. Laxmi Publications.

Hägele, M., Nilsson, K., Pires, J.N., and Bischoff, R. (2016). Industrial robotics. In B. Siciliano and O. Khatib (eds.), *Springer Handbook of Robotics*, chapter 54, 1385–1418. Springer, 2nd edition.

Hancock, P.A., Billings, D.R., Schaefer, K.E., Chen, J.Y., De Visser, E.J., and Parasuraman, R. (2011). A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors*, 53(5), 517–527.

Hein, B., Hensel, M., and Worn, H. (2008). Intuitive and model-based on-line programming of industrial robots: A modular on-line programming environment. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 3952–3957. IEEE.

Hein, B. and Wörn, H. (2009). Intuitive and model-based on-line programming of industrial robots: New input devices. In *IEEE/RSJ Int. Conf. Intelligent Robots and*

Systems (IROS), 3064–3069. IEEE.

Hu, Z., Marshall, C., Bicker, R., and Taylor, P. (2007). Automatic surface roughing with 3D machine vision and cooperative robot control. *Robotics and Autonomous Systems*, 55(7), 552–560.

Ijspeert, A.J., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, volume 2, 1398–1403. IEEE.

Keyes, B., Micire, M., Drury, J.L., and Yanco, H.A. (2010). Improving human-robot interaction through interface evolution. In *Human-robot interaction*. InTech.

Lambrecht, J. and Krüger, J. (2012). Spatial programming for industrial robots based on gestures and augmented reality. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 466–472. IEEE.

Leali, F., Pellicciari, M., Pini, F., Vergnano, A., and Berselli, G. (2013a). A calibration method for the integrated design of finishing robotic workcells in the aerospace industry. *Communications in Computer and Information Science*, 371(37-48).

Leali, F., Pini, F., and Ansaloni, M. (2013b). Integration of CAM off-line programming in robot high-accuracy machining. In *IEEE/SICE Int. Symp. System Integration (SII)*, 580–585.

Madsen, O., Pires, J., Godinho, T., and Araújo, R. (2007). Using digital pens to program welding tasks. *Industrial Robot: An Int. J.*, 34(6), 476–486.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., and Chryssolouris, G. (2016). Augmented reality (AR) applications for supporting human-robot interactive cooperation. *Procedia CIRP*, 41, 370–375.

Morley, E.C. and Syan, C.S. (1995). Teach pendants: how are they for you? *Industrial Robot: An Int. J.*, 22(4), 18–22.

Nachreiner, F., Nickel, P., and Meyer, I. (2006). Human factors in process control systems: The design of human–machine interfaces. *Safety Science*, 44(1), 5–26.

Neto, P. and Mendes, N. (2013). Direct off-line robot programming via a common CAD package. *Robotics and Autonomous Systems*, 61(8), 896–910.

Ong, S.K., Chong, J., and Nee, A.Y. (2006). Methodologies for immersive robot programming in an augmented reality environment. In *4th Int. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH)*, 237–244. ACM.

Pan, Z., Polden, J., Larkin, N., Duin, S.V., and Norrish, J. (2012). Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2), 87 – 94.

Pini, F., Leali, F., and Ansaloni, M. (2014). Offline workpiece calibration method for robotic reconfigurable machining platform. In *19th IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA)*.

Rogowski, A. (2012). Industrially oriented voice control system. *Robotics and Computer-Integrated Manufacturing*, 28(3), 303–315.

Rogowski, A. (2013). Web-based remote voice control of robotized cells. *Robotics and Computer-Integrated Manufacturing*, 29(4), 77–89.

Sabattini, L., Villani, V., Czerniak, J.N., Mertens, A., and Fantuzzi, C. (2017). Methodological approach for the design of a complex inclusive human-machine system. In *13th IEEE Conf. Automation Science and Engineering (CASE)*. IEEE.

Solvang, B., Sziebig, G., and Korondi, P. (2008). *Robot programming in machining operations*. INTECH Open Access Publisher.

Stadler, S., Kain, K., Giuliani, M., Mirnig, N., Stollnberger, G., and Tscheligi, M. (2016). Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control. In *25th IEEE Int. Symp. Robot and Human Interactive Communication (RO-MAN)*, 179–184. IEEE.

Takarics, B., Szemes, P.T., Nemeth, G., and Korondi, P. (2008). Welding trajectory reconstruction based on the intelligent space concept. In *IEEE Conf. Human System Interactions (HSI)*, 791–796. IEEE.

Talignani Landi, C., Ferraguti, F., Sabattini, L., Secchi, C., and Bonfé, Marcello Fantuzzi, C. (2017a). Variable admittance control preventing undesired oscillating behaviors in physical human-robot interaction. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE.

Talignani Landi, C., Ferraguti, F., Sabattini, L., Secchi, C., and Fantuzzi, C. (2017b). Admittance control parameter adaptation for physical human-robot interaction. In *IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE.

Tee, K.P., Yan, R., and Li, H. (2010). Adaptive admittance control of a robot manipulator under task space constraint. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 5181–5186. IEEE.

Tsarouchi, P., Makris, S., and Chryssolouris, G. (2016). Human-robot interaction review and challenges on task planning and programming. *Int. J. Computer Integrated Manufacturing*, 29(8), 916–931.

van Delden, S., Umrysh, M., Rosario, C., and Hess, G. (2012). Pick-and-place application development using voice and visual commands. *Industrial Robot: An Int. J.*, 39(6), 592–600.

van Krevelen, D. and Poelman, R. (2010). A survey of augmented reality technologies, applications and limitation. *Int. J. Virtual Reality*, 9(2), 1–20.

Veiga, G., Pires, J., and Nilsson, K. (2009). Experiments with service-oriented architectures for industrial robotic cells programming. *Robotics and Computer-Integrated Manufacturing*, 25(4), 746–755.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018a). Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*.

Villani, V., Sabattini, L., Czerniak, J.N., Mertens, A., and Fantuzzi, C. (2018b). MATE robots simplifying my work: benefits and socio-ethical implications. *IEEE Robot. Automat. Mag.*

Villani, V., Sabattini, L., Riggio, G., Secchi, C., Minelli, M., and Fantuzzi, C. (2017). A natural infrastructure-less human–robot interaction system. *IEEE Robotics and Automation Letters*, 2(3), 1640–1647.

Zha, X. and Du, H. (2001). Generation and simulation of robot trajectories in a virtual CAD-based off-line programming environment. *The Int. J. Advanced Manufacturing Technology*, 17(8), 610–624.