An Augmented Interface to Display Industrial Robot Faults

Francesco De Pace(✉), Federico Manuri, Andrea Sanna, and Davide Zappia

Dipartimento di Automatica e Informatica, Politecnico di Torino,
C.so Duca degli Abruzzi 24, 10129 Torino, Italy
francesco.depace@polito.it

Abstract. Technology advancement is changing the way industrial factories have to face an increasingly complex and competitive market. The fourth industrial revolution (known as industry 4.0) is also changing how human workers have to carry out tasks and actions. In fact, it is no longer impossible to think of a scenario in which human operators and industrial robots work side-by-side, sharing the same environment and tools. To realize a safe work environment, workers should trust robots as well as they trust human operators. Such goal is indeed complex to achieve, especially when workers are under stress conditions, such as when a fault occurs and the human operators are no longer able to understand what is happening in the industrial manipulator. Indeed, Augmented Reality (AR) can help workers to visualize in real-time robots' faults. This paper proposes an augmented system that assists human workers to recognize and visualize errors, improving their awareness of the system. The system has been tested using both an AR see-through device and a smartphone.

Keywords: Industry 4.0 · Industrial robots
Human-machines interfaces · Augmented reality

1 Introduction

The fourth industrial revolution is bringing both new opportunities and challenges. An increasing number of devices is connected and it is capable of exchanging data in real-time. In an industrial context, modern factories are composed by many automated systems, such as industrial robots, that can perform different tasks, improving the overall production. As the market is becoming increasingly competitive, factories are required not only to enhance the products quality but also to reduce manufacturing and maintenance times. As industrial robots are becoming more powerful and efficient, it is possible to imagine a scenario in which robots collaborate actively with human workers. Development in the Artificial Intelligence's (AI) field have allowed the creation of machines able to work in completely autonomy and to recognize the human workers. Whilst these improvements represent important steps to the realization of machines increasingly complex and sophisticated, there is the risk that workers will not be able

L. T. De Paolis and P. Bourdot (Eds.): AVR 2018, LNCS 10851, pp. 403–421, 2018.
https://doi.org/10.1007/978-3-319-95282-6_30

to understand and recognize what machines are doing, compromising the realization of a real active collaboration. In order to understand how to achieve a true collaboration, a new scientific discipline is born: the so-called Human Robot Collaboration (HRC)[1]. HRC tries to understand how to improve the human-robot collaboration using innovative interfaces. Several works have investigated the effectiveness of original technologies in the human-robot collaboration context. From the development of Cascade Convolutional Network [5] to the ingenious use of a motion capture system [2,6], an increasing number of researches are investigating this new topic. In an industrial context, workers should trust robots to obtain a real collaborations system. Creating such a context is a complex challenge: a human-human collaboration system is considered safe because one human can naturally understand the intention of another human. Thus, understanding the robot's intentions becomes a crucial issue. Intentions can be expressed through the actions that the robot is doing (movements, task, and so on): if workers could visualize them, they would be able to understand the robot's purposes, improving their awareness of safety.

AR can indeed be used to achieve this purpose, since it is able to show information contextualized in the real environment. The origin of this visualization technology dates back to the last years of the sixties, when the first AR prototype was proposed by Sutherland [28]. It was not until the early years of the nineties that the AR concept was formalized by Milgram and Kishino [24]: these authors introduced the definition of *Mixed Reality* as a continuum space going from full reality to full virtuality; within this definition it is possible to identify AR as a category where real elements are dominant and are supplemented by virtual elements. Until few years ago, there was a lack of low cost AR devices and this technology was used only in a limited number of cases. Thanks to the technology improvements, not only smartphones provide all the sensors necessary to implement AR applications, but it is also possible to find on the market several AR see-through devices such as the Meta 2 AR headset [15] or the Microsoft Hololens [16]. Thus, the number of AR applications is greatly increased, from educational or cultural heritage applications [30] to industry ones, such as maintenance-assembly-repair processes or product inspection and building monitoring [29]. It should also be noticed that the effectiveness of a new technology can be measured by its market penetration and AR potentially has a much larger market than Virtual Reality (VR), since it allows users to interact with the real world, which is, at least so far, much more complex than the fictitious environments provided by VR.

Since AR technology is becoming more widespread, several researches are trying to figure out how to use it in the HRC context. In [3,7,8], systems based on projected AR have been developed to visualize in real-time the future motions (trajectories, occupied space) of Automated Guided Vehicles (AGV). Moreover, authors in [9,10], are exploring the use of AR for visualizing the robot's arm movement in the real environment. Ameri et al. [9] developed an AR system in which the worker is not only able to detect which is the object that is going to be manipulated by the robot but also the trajectory that the robot will follow.

Being aware of the robot's movements is indeed useful, but there may be circumstances that require the visualization of other data. In [11], forces applied by the robot arm are represented in the real space as 3D vectors centered in the application point. However, the works described above do not take in consideration situations in which robots are affected by faults. In a human-robot collaboration context, humans work side-by-side with manipulators and faults may increase anxiety in the workers because they are not able to understand in real-time which is the cause of the error. It is then quite important to be able to visualize robots' faults in real time.

Since HRC is a quite new scientific discipline, there is still a lack of researches that have tried to figure out how to develop AR systems for the robots faults visualization.

In this paper, a preliminary study regarding the use of AR for detecting and visualizing faults on robotic arms is proposed. The paper is organized as follows: the background of this work and the problem of fault detection in the context of industrial robots are briefly introduced in Sect. 2, followed by a general classification of any possible fault. The subset of faults considered in this work with the set of 3D assets used to describe the faults, are presented in Sect. 3. The hardware and software architecture of the proposed application is explained in Sect. 4. Tests and the analysis of the gathered results are presented in Sect. 5. Finally, conclusions and possible perspectives are discussed in Sect. 6.

2 Human-Robot Augmented Collaborative Environment

Fault detection is a subject that has been studied since the first use of industrial robots. However, there is still a lack of straightforward techniques to detect and visualize in real-time shortcomings on the manipulators. As described in [4], factory productivity may be widely affected by faults in industrial manipulators. Nowadays, when a fault occurs on an industrial manipulator, one common procedure used to solve the error is represented by the following work-flow:

- a text file, containing the corresponding error, is saved as a log file;
- technicians use this log file to understand the nature of the problem and try to solve the fault using technical manuals and their experience.

This procedure takes a long time and it is not possible to understand in real-time the nature of the fault. In an Human-Robot Augmented Collaborative (HRAC) environment, human workers can work side-by-side with industrial manipulators, visualizing faults directly in the real environment without the necessity of the procedure explained above. Figure 1 shows this scenario. To achieve this purpose, successful strategies to recognize and solve errors in the shortest possible time must be pursued. Many approaches exist to control the robot's state: in [17–19] neural networks have been used to monitor and isolate industrial manipulator's faults. However, these methods suffer for the requirements of high computational power; moreover, workers are not able to visualize the errors on the robot and consequently they cannot recognize the location of the faults.

AR is indeed suitable to achieve this purpose: thanks to its intrinsic capability to enrich the real environment with additional information, faults can be clearly recognized by workers in real-time. Nowadays, industrial robots work inside security cells, completely separated from the human workers to avoid any possible injury. Normally, when a mechanical robot's fault occurs, the manipulator suddenly stops its movements: this is a safety procedure, used to ensure that unprogrammed actions are not performed by the robot. In a HRAC context, however, the arm robot and the human worker carry out tasks side-by-side and unexpected interruptions in the manipulator's movements may increase anxiety and stress in the operators. This paper represents a preliminary work that investigates which 3D metaphors best represent some faults on industrial manipulators. To achieve this purpose, an AR system has been designed and developed: it is able to correctly align the 3D assets on the robot to highlight the nature of the faults.

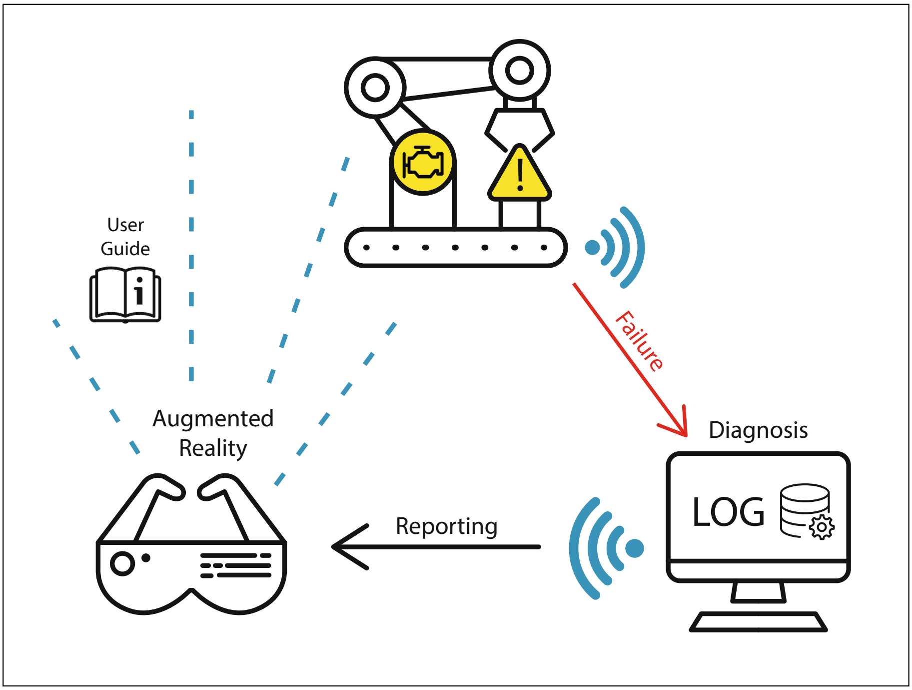

Fig. 1. The HRAC scenario: when a fault occurs on an industrial manipulator, it is both saved on a database and it is possible to visualize the augmented fault in real-time.

2.1 Classification of the Faults

An industrial manipulator is defined as a n-degree-of-freedom (nDoF) arm robot: joints are controlled by using either DC or brushless electric motors, their position is sampled by means of encoders whereas the joints' velocity is measured with tachometers. Arm robots are made of sensors, mechanical parts and actuation systems, thus it is possible to classify errors in three main categories [12]:

- faults on the sensors;
- faults on the mechanical structure;
- faults on the actuation system.

The first category regards errors that may occur during the acquisition of data by sensors: it may happen that the values provided by sensors are wrong even if the physical quantity is actually not affected by any error. The second category refers to faults that may occur on mechanical components; for example, a joint is blocked due to a fault in the brakes or a collision among the robot and an unexpected object occurs and the manipulator suddenly stops. Finally, faults on the actuation system may involve the electrical components, such as the motor drivers and the motors themselves. A fourth category can be added considering the overloading fault [13]. Each manipulator is able to raise weights up to a predefined limit; however, if this limit has overcome, the manipulator suddenly stops its movements. This fault may cause stress and displacements in the structure; hence it should be considered as well as the others introduced above.

3 AR Interface for Fault Visualization

To improve the sense of safety and reliability of the human workers, errors should be identified and highlighted by graphic metaphors that represent the real problems. In a human-robot collaborative scenario, when the robot suddenly stops for one (or more) of the faults introduced in the previous section, the human worker should be immediately able to both understand the typology of the faults and to visualize them in the proper location. In this project, just the collaborative robots are considered and, moreover, a subset of the faults categories introduced above is contemplated:

- fault on the velocity sensor: velocity can be measured by a tachometer; a fault in the tachometer circuit causes the read velocity to be null;
- fault on the actuation system: fault on a motor may stop the rotation of one of the joints;
- collision detection: a collaborative robot is able to foresee an imminent collision and in that case it comes to a sudden stop; it is then important for the human worker to understand that it has stopped not because of an internal error but in order to avoid the collision;
- overloading fault: the industrial manipulator is not able to raise the payload because the object's weight overcomes the robot's limit.

Each of these faults is represented by a 3D asset, superimposed on the fault's location:

– a 3D circular arrow: this model rotates as long as the angular velocity sensor reads correct data, while it stops (also changing color) when the sensor reads null velocity;
– a 3D motor: when an error occurs on a joint's motor, the 3D model starts to blink;
– a 3D sphere: this asset represents the working-area of the manipulator; when a collision is detected, it starts to blink;
– a 3D anvil with a 3D warning signal: when the manipulator stops its movement due to overloading problems, these assets are superimposed on the payload.

4 The System Architecture

In this section the system architecture including hardware and software elements is presented.

4.1 Hardware Architecture

The hardware architecture is composed by three different elements: a Personal Computer (PC), an AR Android device and the industrial manipulators. On the PC, the Ubuntu 16.04 LTS distribution has been installed along with the Robot Operating System (ROS) Kinetic version [20]. The PC works as a server, sending both the instruction to control the robot to the industrial manipulator and the data used to correctly align the 3D assets to the Android device (acting as a client) over a TCP connection. Both devices have to be connected to the same LAN network.

4.2 Software Architecture

The software architecture is divided in three different parts: the first one is represented by the ROS system used by the server to send data both to the manipulator and to the Android client. Information sent to the robot is used to control it, while data sent to the Android device is used to correctly align the 3D assets in the real environment. In fact, using the ROS system the server is able to get some precious information from the robot (such as its joint's orientation or velocity) and send them to the Android device for properly visualize the 3D models. The second one is represented by the robot controller, used for managing the manipulator's behaviour. Finally, there is the software layer used by the Android client to visualize the 3D assets. It has been developed using Unity3D as Integrated Development Environment (IDE) and the Vuforia Software Development Kit (SDK). With Unity3D is possible to manage 3D objects in a relative simple way whereas the main task of Vuforia is to detect and recognize the marker for correctly positioning the 3D models in the real world. Another advantage of using Unity3D is that it is able to build the developed application into an APK for the Android devices.

4.3 Implementation

Since this research project represents a preliminary work, for the development of this project it has been used a 3D model of the Smart-5 Six Comau manipulator (Fig. 2): it is a 6-DoF arm robot, employed for welding operations. The 3D model is directly managed by the client's software, hence only the Android application and the server software using ROS have been developed. To correctly align the 3D robot in the real space, a target (marker), printed on a sheet of format A0, has been used. When the AR device detects the marker, the system can extract some essential information (such as orientation and distance from the camera) to correctly align the 3D assets in the augmented scene.

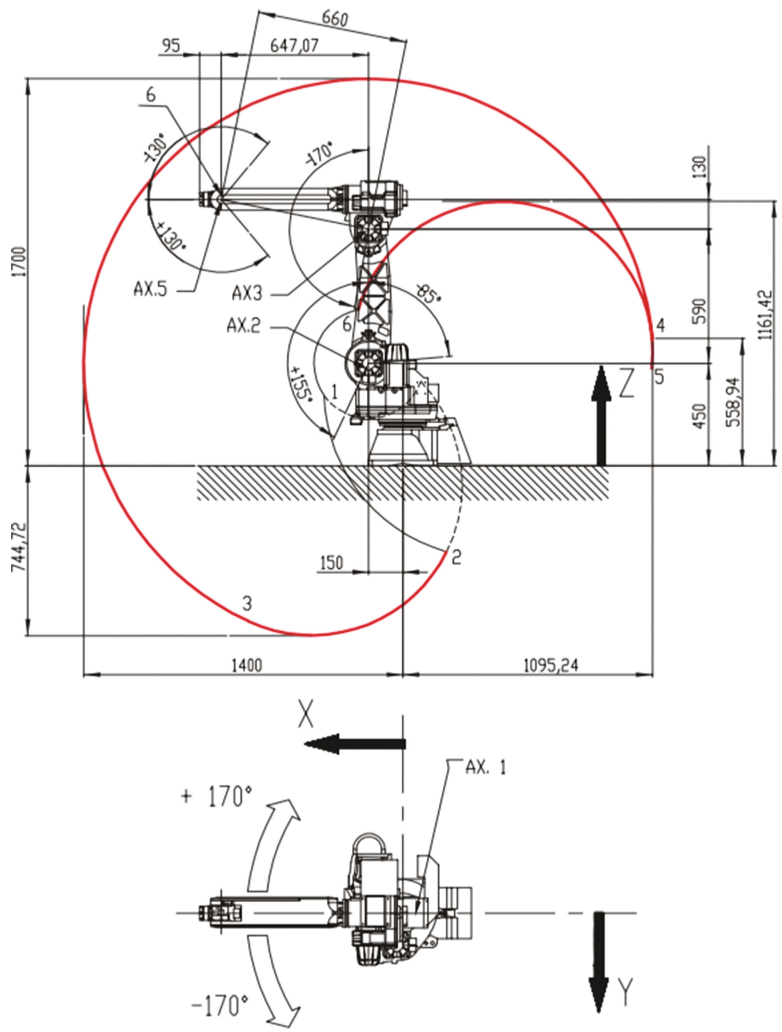

Fig. 2. Operational area (red line) of the Comau Smart-5 Six. (Color figure online)

The main purposes of the server software are to wait for connections from a client and to send the robot simulation to it. To establish the connection, the rosbridge_server package has been used. It is part of the rosbridge suite and it provides a WebSocket transport layer. To create a suitable set of animations for the virtual robot, four nodes have been developed using the C++ language. To represent the Smart-5 Six as a 3D asset, an URDF file describing all the characteristics of the robot has been used [31]. At the system bootstrap, the rosbridge_server node is initialized and it creates a websocket connection on a specific IP-port couple. When the client has established the connection on the websocket, the server can send data for controlling the virtual robot starting the corresponding node. Each of the nodes has a similar structure: it publishes on the "join_states" topic a message of type "sensor_msgs/JointStateMessage" containing the data used to describe the state of the robot joints. In this way, it is possible to change the orientation of each of the joints of the robot. The Android device's software manages the visualization of both the 3D metaphors and the virtual industrial robot. For managing a 3D model consistent with the one used in the server, it has been necessary to use the URDF file that describes the characteristics of the Smart-5 Six robot. Unity3D does not support .urdf extensions and the use of an external plugin has been mandatory [21]. With this plugin, it is possible to create a GameObject from the URDF file, obtaining a real 3D representation of the arm robot. It also allows to develop publisher and

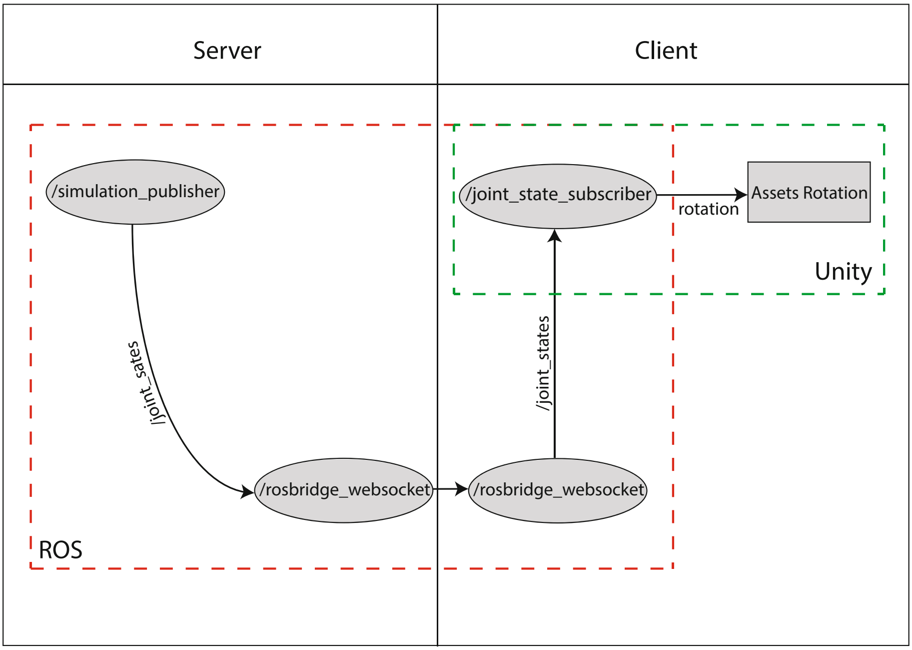

Fig. 3. The ROS nodes architecture.

subscriber nodes using the C# language, allowing the Unity3D application to be compatible with the ROS system of the server. When the Android application starts, it establishes a connection with the server using a websocket. Then, a ROS node subscribes to the "joint_state" topic, waiting for incoming data. When data arrive, they are used to rotate the arm robot GameObject created previously from the URDF file. Figure 3 shows the nodes architectures.

5 Tests and Results

In order to assess the framework usability and the clarity of the assets, some tests have been held at Politecnico di Torino. The tests were focused on some objective and subjective parameters: specifically, the user's understanding of each problem, the clarity of representation of each problem and the different user experience using AR glasses and smartphone.

Users were students and members of the Polytechnic of Turin. There were 10 testers, 8 men and 2 women, with ages that ranged between 20 and 30 years. A computer science laboratory has been used for the test, with an artificial lighting comfortable for using AR devices. During the test, users had to visualize four different scenes using an AR see-through device, the Epson Moverio BT-200 (with Android 4.0 as operating system) [22] and an Asus Zenfone 2 (with Android 5.0.1 as operating system). Each scene is composed by the virtual Smart-5 Six Comau model and a 3D representation of a fault. At the beginning of the scene, the robot is working normally, making some pre-defined tasks. At a certain random moment, a fault occurs and the robot may stop its movement or may continue its task, depending on the nature of the fault. The scenes are the following (see Figs. 4, 5, 6 and 7):

– Scene 1: this scene focuses on the fault on the velocity sensor. At the beginning the robot is acting normally; when the fault occurs, the virtual manipulator does not stop its movements because this type of error does not affect its motion but the corresponding 3D metaphors (the 3D arrows) stop to rotate and their color changes.
– Scene 2: this scene focuses on the fault of the actuation system. At the beginning the robot is acting normally; when the fault occurs, the virtual manipulator stops its movements and the engine of the blocked joint is highlighted.
– Scene 3: this scene focuses on the collision detection. In this scene there are two different robots, the 3D Smart-5 Six and a 3D AGV that is moving around the environment. At the beginning the 3D Smart-5 Six is acting normally; when it foresees the collision with the AGV, it pauses its movements, letting the AGV pass, and the sphere starts to blink for highlighting the risk of a collision. When the AGV has passed, the robotic arm resumes moving.
– Scene 4: it is the scene that focuses on the overloading problem. At the beginning the robot is acting normally: when it tries to raise a payload that weights more than the robot's limit, the manipulator suddenly stops and an anvil appears, superimposed on the payload.

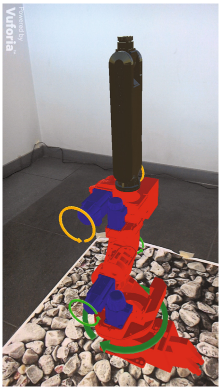

Fig. 4. Scene 1: the 3D arrows used to visualize the joints' angular velocity. (Color figure online)

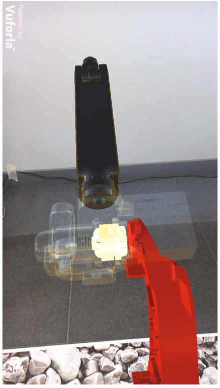

Fig. 5. Scene 2: when the robot stops its movements, the internal blocked engine is highlighted.

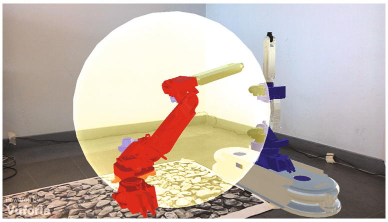

Fig. 6. Scene 3: when the robot foresees the collision with the AGV, it stops its movements changing the color of the sphere. (Color figure online)

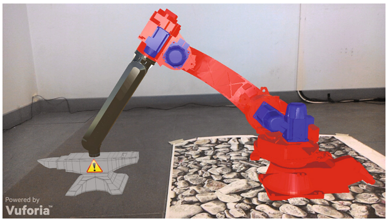

Fig. 7. Scene 4: when the robot tries to raise a payload that weights more than its intrinsic limits, a 3D anvil is superimposed on the payload

For each session, two scenes were visualized with the AR glasses and the other two with the mobile device: in order to get relevant results, the scenes were randomly selected. Users had to examine the whole scene in order to understand which was the nature of the fault. Users can freely move in the environment, watching the scene from different perspectives. A questionnaire has been created and proposed to the user after the test: prior to performing it, users were individually introduced, firstly with a short tutorial of the application and next with a description of the aim of the project. Testers were all volunteers and their participation were not remunerated. After the starting tutorial, they were placed in front of the target and they had to accomplish the whole test. During the test users had to pay attention to some factors in order to evaluate correctly the fluency, the usability and the utility of the system. After the test, users had to fill the questionnaire, composed by 24 questions.

5.1 Results

The questionnaire is divided in four different sections: the first one is about the user's information, computer science knowledge and familiarity with augmented reality. The second section is divided in four sub-parts, one for each scene: the questions are relative to the clarity and the utility of the symbology of the assets used for the faults representation. Since scene 3 is slightly different from the others (the robot stops its movements not because of a fault but because it foresees a collision), the questions relative to it are marginally different from the ones used in scenes 1, 2 and 3. Depending on the typology of the question, three different modality of answers are presented in the questionnaire: double answer (Yes/No), linear scale from 1 (strongly disagree) to 5 (strongly agree) and multiple answer. In the multiple answer typology, users have to choose from the following fault list which one represents the most reasonable error:

– fault on joint position sensor;
– engine block;
– angular speed sensor fault;
– joint block;
– overload;
– fault on current sensor;
– collision detected;
– fault on the actuation system.

In the third section, some questions are proposed to compare the application usability with the devices used, the AR glasses and the smartphone. Finally, in the last section, there is a specific optional area where users can add their comments and feedbacks. The following images (Figs. 8, 9, 10, 11, 12 and 13) summarize the results of the test.

5.2 Results Analysis

Tests have been evaluated with a number of participants (10) too small to obtain results with statistical validity. Moreover, subjects of these tests were university students without any background in the robotic domain. Despite this, the proposed study can be suitable to lay the foundations for future developments. Testers have found some hardware-related problems: since the Moverio glasses have a very limited field-of-view (FOV) that is around 23°, subjects could not watch large objects entirely and they were forced to change their point of view. This is a well-known limitation of the see-through devices and only an improvement in the underlying technology could overcome this issue. For each scene, users both had to understand if the robot had a problem and they had to evaluate the intuitiveness of the symbology. Finally users had to understand the nature of the faults.

From Figs. (8, 9, 10, 11, 12 and 13), it is visible that in scene 1 the 70% of testers have reported that the robot has performed its routine without any problem (in fact a sensor fault does not affect the robot's motion). The symbology

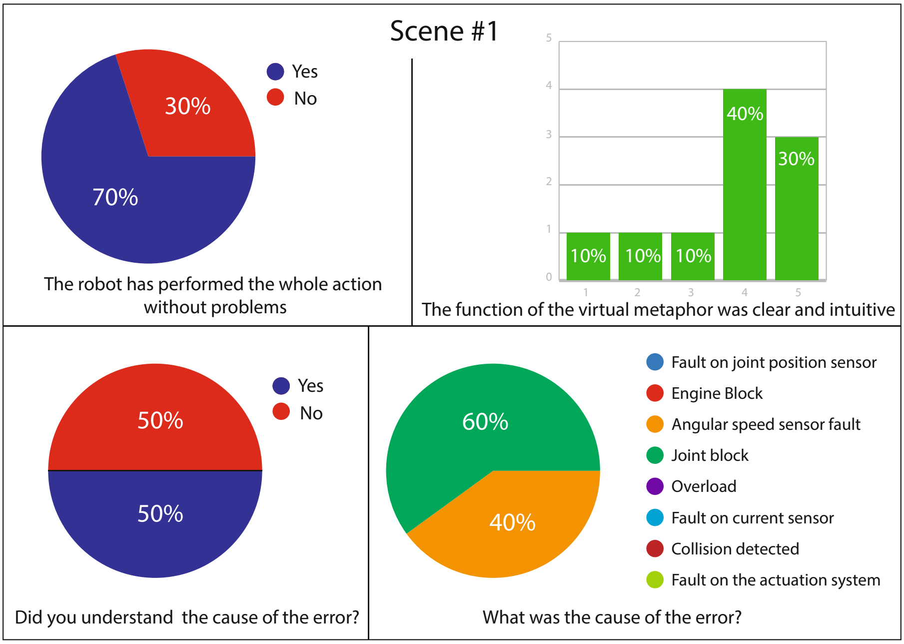

Fig. 8. Scene 1 results.

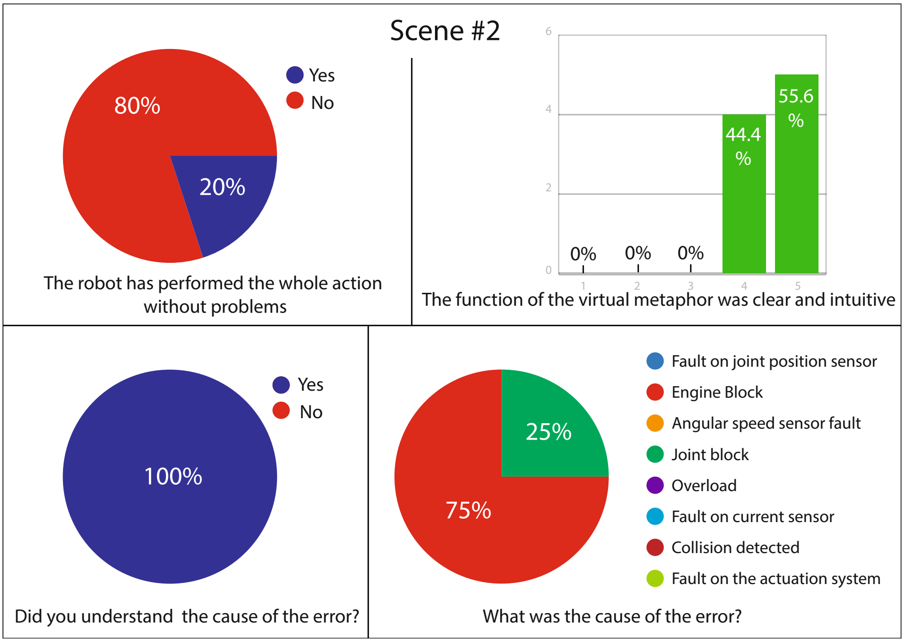

Fig. 9. Scene 2 results.

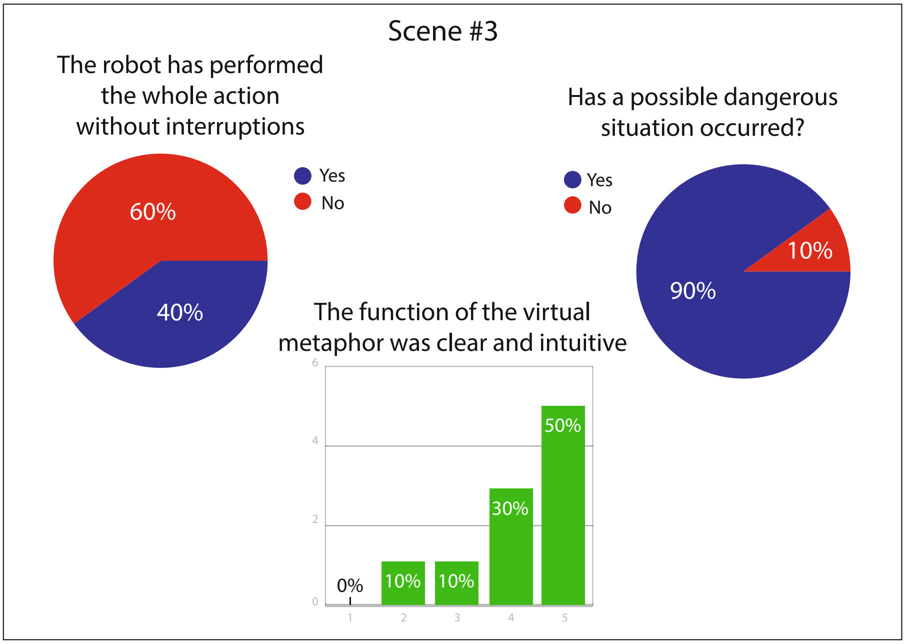

Fig. 10. Scene 3 results

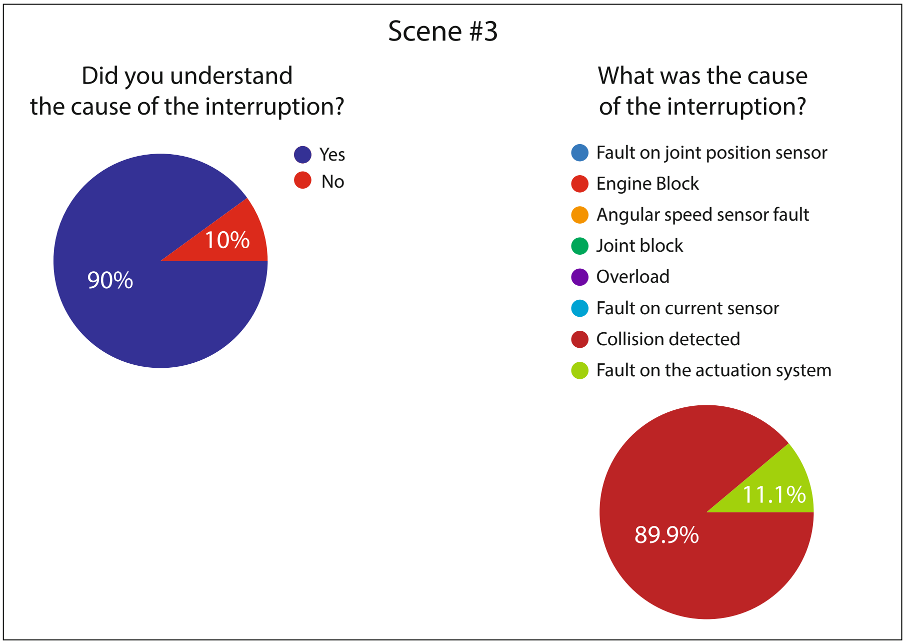

Fig. 11. Scene 3 results

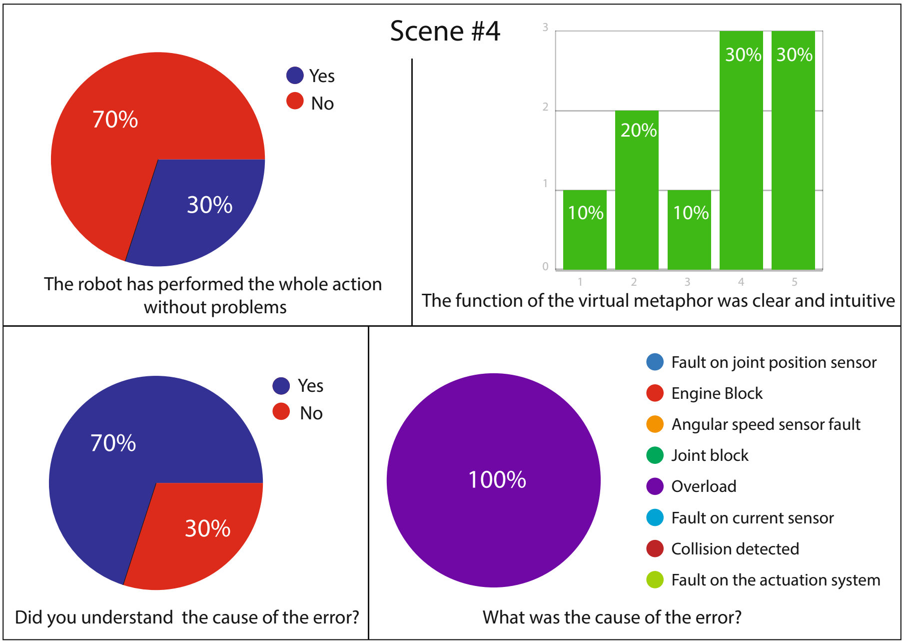

Fig. 12. Scene 4 results

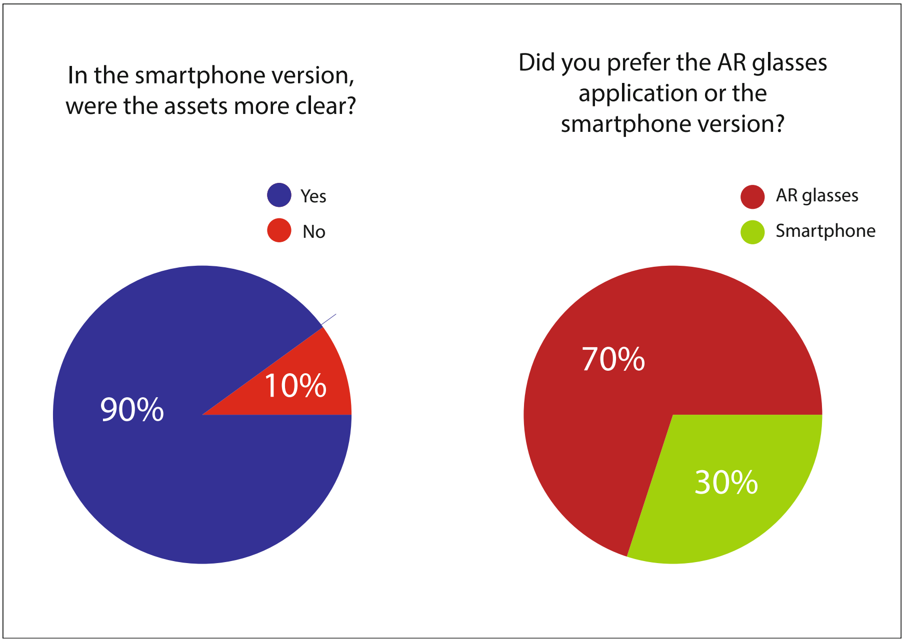

Fig. 13. The comparison between the AR glasses and the smartphone.

responses are also important, because 70% of the testers agreed that the virtual arrows have been clear and intuitive in representing speed, but only few users understood the real problem. In fact, it has been confused with the joint block error. It is important to know that this type of error is particular, and for this reason it is very difficult for users without some specific knowledge in robotics to understand it. In the second scene 80% of the testers have recognized that the robot had a problem, and 100% of them agreed or strongly agreed that the symbology was very clear. Furthermore, among the users who understood there was a fault, 25% said that the robot had a joint block and 75% chose engine block. Despite the correct answer was engine block (75%), the latter can be regarded to cause a joint block: hence the remaining 25% of testers understood the nature of the problem but not the real cause. In the third scene, despite only 60% of the users have noticed that the robot stopped its movements when the AGV was passing, 90% of the subjects have understood that a possible dangerous situation was occuring. In fact, 80% of testers have found the sphere symbology intuitive or very intuitive; furthermore, 89.9% of them understood the right cause of the problem. In the fourth scene, although testers have indicated that the symbology should be more intuitive and clear, 70% of them have understood that the fault was due to overloading problem. Finally, subjects have indicated that they would have preferred the smartphone version, since it would have been more comfortable and assets would have been clearer and more understandable.

5.3 Additional Test

It has been decided to evaluate the same system with a real robot. The aim is to verify if the visualization of 3D assets is affected by the virtual robot or not. To do this, a humanoid physical robot was used, because it was not possible to use a proper industrial robot. The real robot is the open source 3D printed life-size humanoid robot InMoov [23] although it is not an industrial manipulator, its arm can be seen as an arm robot composed by different joints, each of them controlled by an electric motor. Thus, it is indeed suitable for testing the effectiveness on the 3D assets.

Both the same assets and the same Android devices have been used for this additional test. The main difference with the previous test is that the robot is already blocked because an error has occurred. Ten new testers have been found and again they were not robot technician expert. Subjects had to identify the nature of the faults, visualizing the 3D metaphors superimposed on the real robot. As in the previous session, two scenes were visualized with the AR glasses and the remainder with the mobile device. Figure 14 shows the InMoov robot with a 3D metphor correctly aligned.

The results of this additional test confirm in part the hypothesis introduced at the beginning of this section: in fact, in scene 2 and 4 more than the 80% of the testers understood the nature of the faults, confirming that using a virtual representation of the virtual robot does not affect the effectiveness of the 3D metaphors. Also in scene 1, users faced the same problems found with the virtual version. On the other hand, results show some unexpected results for scene 3: in

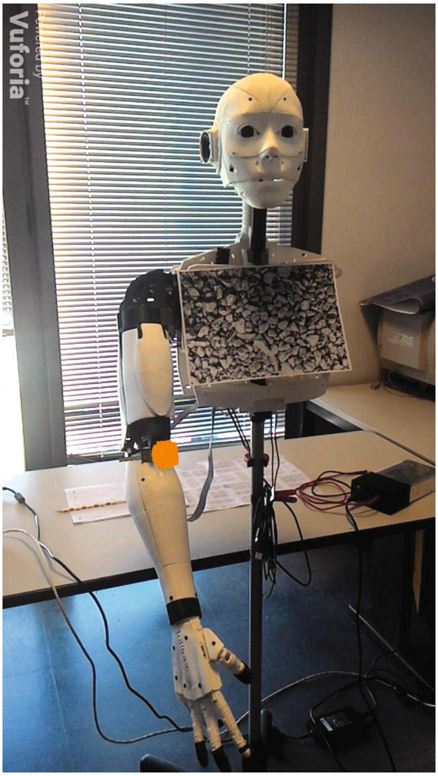

Fig. 14. The 3D metaphor, that represents a fault on the actuation system, correctly aligned on the real robot.

the previous collision scene, 90% of the subjects have been able to recognize the correct problem; in the additional test, only 50% of the testers could understand the true nature of the problem. As the robot was already blocked, the 3D collision sphere was already blinking, creating disturbance in the scene. Thus, it is possible to deduce that since there was the absence of an initial correct phase, in which the robot can complete its task without interruptions, users could not compare the initial phase with the fault phase and, consequently, they could not understand the nature of the faults.

Another important result is that using a physical robot users preferred to use the AR glasses despite the 3D assets were the same used for the previous version. Indeed, in the case of the virtual robot, users had difficulties in visualizing all the 3D assets at the same time using the Moverio glasses, because of the well-known limitation of this kind of devices. On the other hand, in the real robot scenario, subjects have been able to focus only on the 3D metaphors, which are much less demanding as the FOV is concerned: thus, in this case they have preferred the Moverio glasses.

6 Conclusions

In this paper a new AR fault visualization system has been proposed. In a human-robot collaboration context, human operators work side-by-side with industrial manipulators, hence they have to trust them, especially when faults occur, creating unpredictable scenarios. Thus, the new HRAC environment has been proposed: in this scenario, not only workers carry out tasks close to industrial manipulators but they can visualize faults, by means of 3D metaphors, directly on the true location of the errors. The effectiveness of the proposed 3D assets has been tested: results show that the use of a virtual robot model or of a real one does not affect the clarity of the visualization. Issues were found relative to the representation of the angular speed value: most of the visualization tools for robotic systems, such as RVIZ [25] or OpenHRP [26], represent rotations using 3D arrows [27] and test result seem to hint at possible issues in the application of this typology of interface in an AR environment. One possible explanation might be that some metaphors are suitable only for a completely virtual world, while others can be used also in an AR scenario.

The two experiments differed from the quantity and the dimensions of the 3D assets represented in the scene: this difference has indeed influenced the choice of the most suitable device. In fact, in the case of the virtual model, the limited FOV of the Moverio glasses results into a worse user experience; on the other hand, in the real robot scenario, since the quantity and the dimensions of 3D metaphors to be visualized were less demanding, the Moverio's FOV was sufficient to guarantee a reasonable experience.

Further experiments will be taken to better investigate which metaphors are suitable for representing faults in an augmented reality system, also using some see-through devices, such as the Microsoft Hololens, with a greater FOV.

Acknowledgements. This work is co-funded by the regional project HuManS (Human Centered Manufacturing Systems).

References

1. Bauer, A., Wollherr, D., Buss, M.: Human-robot collaboration: a survey. Int. J. Humanoid Robot. **5**(01), 47–66 (2008)
2. Morato, C., Kaipa, K.N., Zhao, B., Gupta, S.K.: Toward safe human robot collaboration by using multiple kinects based real-time human tracking. J. Comput. Inf. Sci. Eng. **14**(1), 011006 (2014)
3. Coovert, M.D., Lee, T., Shindev, I., Sun, Y.: Spatial augmented reality as a method for a mobile robot to communicate intended movement. Comput. Hum. Behav. **34**, 241–248 (2014)
4. Chen, J., Patton, R.J.: Robust Model-based Fault Diagnosis for Dynamic Systems, vol. 3. Springer, Boston (2012). https://doi.org/10.1007/978-1-4615-5149-2
5. Miseikis, J., Knobelreiter, P., Brijacak, I., Yahyanejad, S., Glette, K., Elle, O.J., Torresen, J.: Robot Localisation and 3D Position Estimation Using a Free-Moving Camera and Cascaded Convolutional Neural Networks (2018). arXiv preprint arXiv:1801.02025

6. Nikolaidis, S., Lasota, P., Rossano, G., Martinez, C., Fuhlbrigge, T., Shah, J.: Human-robot collaboration in manufacturing: quantitative evaluation of predictable, convergent joint action. In: 2013 44th International Symposium on Robotics (ISR), pp. 1–6. IEEE (2013)
7. Chadalavada, R.T., Andreasson, H., Krug, R., Lilienthal, A.J.: That's on my mind! robot to human intention communication through on-board projection on shared floor space. In: 2015 European Conference on Mobile Robots (ECMR), pp. 1–6. IEEE (2015)
8. Matsumaru, T.: Mobile robot with preliminary-announcement and display function of forthcoming motion using projection equipment. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006, ROMAN 2006, pp. 443–450. IEEE (2006)
9. Akan, B., Çürüklü, B.: Augmented reality meets industry: Interactive robot programming. In: Proceedings of SIGRAD 2010: Content Aggregation and Visualization, 25–26 November 2010, Västerås, Sweden, no. 052, pp. 55–58. Linköping University Electronic Press (2010)
10. Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., Chryssolouris, G.: Augmented reality (AR) applications for supporting human-robot interactive cooperation. Procedia CIRP **41**, 370–375 (2016)
11. Mateo, C., Brunete, A., Gambao, E., Hernando, M.: Hammer: an android based application for end-user industrial robot programming. In: 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6. IEEE (2014)
12. Fantuzzi, C., Secchi, C., Visioli, A.: On the fault detection and isolation of industrial robot manipulators. IFAC Proc. Vol. **36**(17), 399–404 (2003)
13. Singh, V. D., Banga, V. K.: Overloading failures in robot manipulators. In: International Conference on Trends in Electrical, Electronics and Power Engineering (ICTEEP'2012)/Planetary Scientific Research Centre, pp. 15–16 (2012)
14. Michieletto, S., Chessa, N., Menegatti, E.: Learning how to approach industrial robot tasks from natural demonstrations. In: 2013 IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO), pp. 255–260. IEEE (2013)
15. Meta 2 ar headset. Accessed 2018
16. https://www.microsoft.com/it-it/hololens
17. Eski, I., Erkaya, S., Savas, S., Yildirim, S.: Fault detection on robot manipulators using artificial neural networks. Robot. Comput.-Integr. Manuf. **27**(1), 115–123 (2011)
18. Vemuri, A.T., Polycarpou, M.M.: Neural-network-based robust fault diagnosis in robotic systems. IEEE Trans. Neural Netw. **8**(6), 1410–1420 (1997)
19. Ghrieb, A.O., Kourd, Y., Guersi, N.: Supervision of industrial manipulators using ANFIS system. In: 2017 17th International Conference on Control, Automation and Systems (ICCAS), pp. 161–166. IEEE, October 2017
20. http://www.ros.org/
21. https://github.com/siemens/ros-sharp/wiki
22. https://www.epson.it/products/see-through-mobile-viewer/moverio-bt-200
23. http://inmoov.fr/
24. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE Trans. Inf. Syst. **77**, 1321–1329 (1994)
25. https://github.com/ros-visualization/rviz
26. https://fkanehiro.github.io/openhrp3-doc/en/

27. Hulin, T., Hertkorn, K., Preusche, C.: Interactive features for robot viewers. In: Su, C.-Y., Rakheja, S., Liu, H. (eds.) ICIRA 2012. LNCS (LNAI), vol. 7508, pp. 181–193. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33503-7_19
28. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the AFIPS, pp. 757–764. ACM, San Francisco (1968)
29. De Pace, F., Manuri, F., Sanna, A.: Augmented reality in industry 4.0. Am. J. Comput. Sci. Inf. Technol. **6**(1), 1–7 (2018)
30. Sanna, A., Manuri, F.: A survey on applications of augmented reality. Adv. Comput. Sci. Int. J. **5**(1), 18–27 (2016)
31. https://github.com/siemens/ros-sharp