

Virtual Commissioning with Mixed Reality for Next-Generation Robot-Based Mechanical Component Testing

Michael Filipenko^a, Alexander Poeppel^a, Alwin Hoffmann^a, Wolfgang Reif^a, Andreas Monden^b, and Markus Sause^b

^aInstitute for Software & Systems Engineering, University of Augsburg, Germany

^aEmail: {filipenko, hoffmann, poeppel, reif}@isse.de, Phone: +49 821 598-2174

^bInstitute of Materials Resource Management, University of Augsburg, Germany

^bEmail: {andreas.monden, markus.sause}@mrm.uni-augsburg.de, Phone: +49 821 598-3435

Abstract

This paper presents a novel approach for robot-based mechanical component testing with a focus on how mixed reality can be used to improve this complex workflow. First we give a short overview of mechanical component testing and which benefits and challenges arise when using robots instead of classic testing actuators. We then explain in detail how mixed reality can be used during commissioning. Afterwards we give an overview of the software side of our approach by explaining the general architecture as well as showing various performance metrics. The main contributions in this paper are (1) a novel approach for combining mixed reality with robot-based mechanical component testing (2) a concept architecture for importing various data sources (like ROS oder Vicon data) into Unity and deployed to different mixed reality devices without requiring adaptation.

1 Introduction

The primary goal of component testing is to demonstrate the mechanical adequacy of individual components or assemblies, which are later joined to form a larger structure (e.g., an aircraft or car). The adaptation or rework of already installed components is very expensive and time-consuming, thus it is far more efficient to test individual components and identify faults prior to assembly. Furthermore, component testing is integral to a manufacturer's internal quality management and compliance with regulatory standards.

The field of component testing can be divided into destructive and non-destructive methods. While the latter comprises testing methods that are applied to determine material properties and detect material defects without compromising the integrity of the component (e.g., with optical or acoustic sensors), destructive testing evaluates mechanical properties by (partially) destroying the specimen, which is consequently rendered unusable after testing. Here we focus on (destructive) mechanical testing where the behavior of the component or its materials is studied during application of an external load.

In Section 3, we introduce our novel testing facility for mechanical component testing with two cooperating large payload industrial robots. Such a facility offers various advantages in comparison with classic electro-mechanical or hydraulic testing machines: (1) The robots can test the component flexibly in various positions and (2) even synchronized testing by both robots simultaneously is feasible. (3) The robots can apply axial and torsional load on the component simultaneously. (4) Force/torque vectors can be altered over the duration of the test, optionally depend-

ing on the response of the examined component. However, the manual setup of testing procedures is complex (cf. Section 3) and could benefit greatly from automated assistance.

Hence, we decided to use mixed reality [1] in order to assist manual setup. In contrast to entirely virtual commissioning, we term this form of assistance *Mixed Reality Commissioning* (cf. Section 4). This approach aims to accelerate the setup of testing procedures by aiding the test engineer on-site and thus is not reliant on continuously updating a simulation of the test bed to reflect the present state of its real-world counterpart.

Section 7 concludes this paper with a summary of the main contributions and an outlook on future work.

2 Related Work

Within the last years Augmented and Virtual Reality have advanced rapidly and even reached the consumer market. Moreover, much research has been conducted on the potential of this technology (especially with new hardware like the Microsoft HoloLens) to improve the area of robotics [2]. For example Blankemeyer et al. [3] have recently demonstrated how industrial robots can be programmed using Augmented Reality (AR). Using optical markers tracked by a HoloLens, virtual objects could be placed next to the robot. After recording the assembly task using AR the task could be executed with real objects. Rudorfer et al. [4] presented an approach in 2018 in which they used a HoloLens in combination with a UR5 to program pick and place operations. Even though their proof of concept turned out successful they mentioned problems such as insufficient accuracy during the operations.

Chacko and Kapila [5] showed another interface for programming a robot arm in 2019. Instead of a HoloLens they used a smartphone to augment the world around the robot and make it possible to send instructions to the robot for grip items a user had previously selected in AR. A similar approach was presented by Kapinus et al. [6] in which they used a tablet and its AR capabilities to teach a robot a sequence of pick and place tasks. The developed user interface was specifically designed to enable people without any prior knowledge about robots to program them.

Muhammad et al. [7] presented a concept how AR could be used for robots supporting the Robot Operating System (ROS) to foster a shared human robot environment. By visualizing robot data like sensory queues or future actions such as path trajectories they aimed to improve and simplify the way humans interact with robots. Ghiringhelli et al. [8] used AR in a multi-robot environment to display the current state of each individual robot since swarm behavior is very difficult to debug once deployed on real robots. Their solution was to augment the video stream of a camera mounted in the room the robots where in. For the localization of each robot, LEDs were used which were mounted on each robot and were detectable by the camera.

As has been shown, many different approaches exist for the combination of Augmented Reality and (industrial) robotics. With our approach we want to show another possibility of how Augmented Reality can be used - namely in the field robot-based destructive component testing.

3 Robot-Based Component Testing

The robot-based testing facility (cf. Fig. 1) is built around a clamping surface spanning 7 m x 2.5 m for the flexible positioning of testing objects as well as a linear, electromechanical testing actuator from ZwickRoell. This testing actuator is used to apply a uniaxial load of up to 100 kN to the component under test.

In any type of component testing, available space on the test bed is a precious commodity since the setup of load tests is an arduous task and many components require multiple configurations on the test bed. Thus, placement of a new component must occur within the unoccupied space while still satisfying all requirements for testing the desired parameters. In addition, the physical movement of mechanical fixtures often involves heavy items that need fork-lifters or crane operations.

The use of two KUKA KR 1000 Titan robots, which are placed on opposite sides of the longer edges of the clamping surface, is part of the innovative approach of this research facility. Each robot is equipped with a 6D force/torque sensor on its flange and consequently can be employed as a flexible testing actuator which is able to apply forces and torques in all degrees of freedom.

Aside from the advantages of using robots for component testing, it also introduces additional complexity to the optimized positioning of the component under test. The reachability of the object must be considered when placing it. While a robot is more flexible overall than a linear actuator, it may not reach every position on the clamping sur-

face. Even positions the robot is able to reach might not be desirable for placing the object, since the maximum force cannot be achieved in every position e.g., when the robot arm is fully extended.

In order to predict the optimal positions, we developed a static force analysis model using the robot's Jacobian matrix in order to estimate the maximum applicable forces. The goal of statics calculations is to determine the relationship between the generalized forces exerted by the end-effector and the generalized torques applied to the joints with the manipulator at an equilibrium configuration which we can assume for component testing. Let τ denote the vector of joint torques and γ_e the vector of generalized end-effector forces with $\gamma_e = [f_e^T \mu_e^T]^T$. Here, f_e are the 3-dimensional force contributions and μ_e are the 3-dimensional torque contributions. According to [9], the relationship between the end-effector forces γ_e and the joint torques τ is given by the transpose of the geometric Jacobian J which is subject to the manipulator's joint configuration q :

$$\tau = J^T(q)\gamma_e$$

Hence, the maximum component testing force at the end-effector can be calculated using the geometric Jacobian J and the available joint torques. To calculate the available joint torques, the joint space dynamic model of the manipulator [9] is consulted:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau - J^T(q)\gamma_e$$

$B(q)$ is a 6×6 symmetric, positive-definite matrix representing the joint space inertia matrix. $C(q, \dot{q})$ is a 6×6 matrix such that $C(q, \dot{q})\dot{q}$ is the vector of Coriolis and centrifugal terms. $g(q)$ is the vector of gravity terms. As component testing is performed at very low velocities and accelerations, we can assume that both \dot{q} and \ddot{q} are equal to 0. Hence, we can simplify the joint space dynamic model and rewrite it as follows:

$$J^T(q)\gamma_e = \tau - g(q)$$

In order to compute the maximum generalized end-effector forces γ_e^{max} for a given joint configuration q and given maximum joint torques τ^{max} , we utilize the following equation:

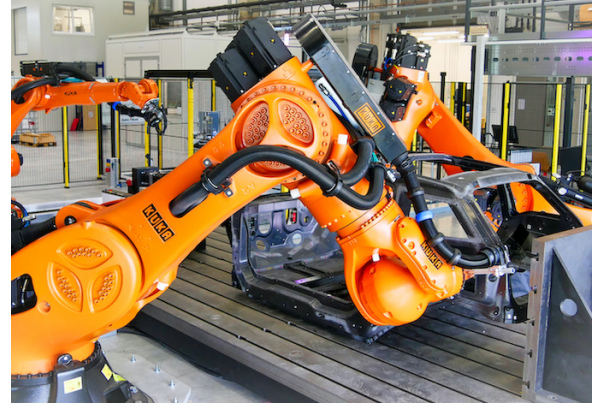
$$\gamma_e^{max} = (J(q)^T)^{-1} (\tau^{max} - g(q))$$

Depending on the specific robot pose, the gravity term either increases or decreases the maximum end-effector forces (e. g., when pushing downwards, the robot's weight will increase the end-effector forces). We use this equation to estimate the highest reachable end-effector forces. The dynamics are calculated using an internal library function from the KUKA.Load software. As a result, we are able to extend the forces for component testing well beyond 10 kN which correspond to the payload class of the robot.

Apart from the reachability of the object's position, the path of the robot to the contact point and the path taken whilst applying the test forces must also be collision free. Planning this in advance for every new test setup is fairly time consuming and typically not within the expertise of a



(a) CAD overview of the test stand



(b) Real test facility with a BMW i3 chassis

Figure 1 The facility for mechanical component testing is comprised of two KUKA KR 1000 titan robots, an electromechanical testing actuator from ZwickRoell and a clamping surface (7 m x 2.5 m) for mounting testing components. The third robot is intended to facilitate further inspection methods (e.g., 3D optical measurement systems).

test engineer for whom the robot is merely supposed to be a tool.

4 Approach

In our context, the term *Mixed Reality Commissioning* describes the combination of a regular, real world commissioning process with virtual elements for quick and intuitive verification of feasibility. This is achieved by using an optical see-through head-mounted display like the HoloLens from Microsoft to visualize information (often referred to as holograms) in the user's field of view. This information can range from simple text-based output over pictures or videos to complex CAD or sensor data and can both be fixed in space or follow the movements of the user. Such holograms can be used to optimize the movements of a robot, to simplify the teach-in or to visualize planned trajectories [2].

Our approach focuses on simplifying the commissioning process for new component tests and can be split into three levels of assistance whereby each level is based on the preceding levels. The first and simplest level merely visualizes additional information about the robot, the component used for testing and other objects in the test bed. Displaying current joint and workspace limits of the robot or some type of heat map representing the maximum applicable force could greatly assist the operator in finding a suitable robot position. Regarding the tested component, further details like optimal contact points and force vectors could be helpful for operators. Optionally, information on the current component and suggested mounting points may be displayed. The second level is the usage of holograms to plan the setup of a new test scenario. Instead of moving real objects their virtual copies may be used. This accelerates the setup itself because moving holograms around to test several positions is much faster than having to move their real counterparts or repeatedly changing a setup in a fixed simulation.

The third level of support focuses on planning the robot trajectory using mixed reality. In combination with positioning the components beforehand the operator could check the trajectories in the real world for possible collisions. Therefore no time intensive adaptations of the robot simulation have to be performed to ensure collision free path planning. The user can achieve this quickly in the real world without the need of measuring each possible obstacle for input into the simulation. By adding a possibility for the operator to constrain the robot movement to a given space with virtual walls within the proposed system, the operator can easily mitigate possible collisions with real-world objects not present in the simulation.

5 Preliminary User Experience Design Study

In order to design and evaluate new concepts for improving the workflow using Mixed Reality Commissioning, we decided to implement our approach from section Section 4 as a game initially. Since usability and user experience have a huge impact on how much people enjoy playing a game we identified the implementation of a game as a suitable initial step, in order to create a user interface which is (1) easy to understand, (2) simple to use and (3) efficient.

By simulating the circumstances inside a game and abstracting from technical barriers (e.g. suitable mounting fixtures or specific material properties) we can focus on the previously mentioned aspects. It also allows us to create prototypes more rapidly and present our results to a broader audience.

5.1 Technologies

HoloLens (1st gen) was chosen as the target hardware, due to its current advantageous capabilities (self-sufficient, ease of use and setup, spatial understanding, good integration inside Unity). For development Unity 2019.2.13f1

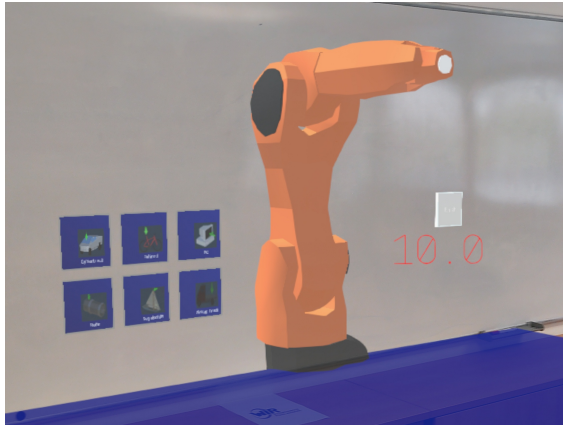


Figure 2 Game overview: Six blue buttons on the left side to select a specimen. Next to it the robot and the clamping surface used for testing. A timer for the user in red (showing 10.0) and above it a white button to confirm placement.

[10] was used with the integrated Vuforia plugin [11] (Version 8.6.7) for marker tracking. The Mixed Reality Toolkit [12] (Version 2.2.0) was used to leverage basic interaction functionalities with the HoloLens.

5.2 Current Gameplay

To introduce a new user to the necessary gestures to interact with the holograms and to make sure the operator is wearing the device as intended, a quick tutorial is shown when starting the application. It consists of three steps:

- Aligning the display correctly by showing content, that is only visible in its entirety once the user is wearing the device correctly. An Air-Tap gesture is used for confirmation of correct positioning. This gesture will be used later on as well to select specimen.
- Using the Air-Drag gesture of the HoloLens to move a cube into a specified area. This gesture is used to move specimen across the clamping surface later on.
- Using Two-Hand manipulation from the HoloLens to rotate a cube by 45 degrees. This gesture is required to rotate specimen.

After completing the tutorial the user has to scan a marker in order to align the holographic scene properly with the real world. Once the alignment is complete, the user sees a simplified version of the testing facility shown in 1a.

Six holographic buttons for selection of the specimen are located statically in close proximity to the clamping surface as shown in figure 2. By performing an Air-Tap on the corresponding icon the operator can select a single specimen.

The selected specimen will subsequently appear on the clamping surface at a randomly selected position. The selected position is also dependent on the difficulty of the game, as the difficulty increases if the user is required to rotate the object in order to achieve sufficient force exerted by the robot and decreases if the object is placed in the correct orientation by the system.

The operator then has the ability to move a specimen around using the gestures previously practiced in the tutorial. During interaction the specimen is highlighted with a color on a scale from red to green (cf. figure 3) denoting the force the robot can achieve relative to the force required for testing the specimen. As previously mentioned in section 3 the force the robot is capable of applying is dependent on its current position and consequently the placement of the specimen under test.

Once the user is satisfied with the result a button needs to be pressed to submit it. Afterwards an animation of the robot reaching for the specimen and crushing it is shown and a highscore is displayed.

5.3 Real World Transfer

While the workflow and the interactions described in section 5.2 are perfectly adequate for the game, they need to be adapted to real-world commissioning processes described in section 4.

To harness the benefits of providing a live feedback system during the commissioning process a few adaptations must be made.

One issue that comes to mind is the lack of accuracy in a holographic system such as (in our case) the HoloLens. Although it can keep holograms stationary with a high degree of accuracy whilst using a marker, there still remains an inaccuracy of a few millimeters, which is not acceptable for our use-case, since a slight misalignment of the KR-1000 could lead to significant damage. Furthermore, there is no guarantee that markers will constantly remain visible, in which case the alignment error may exceed a few millimeters.

A possibility to address this lack of accuracy is to merely move the robot to staging position within a few centimeters of the final position with help of the AR headset. Continuing from this staging position, the worker can then refine it further towards the desired goal position using standard control mechanisms.

By making it possible to move the positioning process into the real world the commissioning will be simplified. This is mainly caused by removing the need to have an accurate 3D-simulation, in which all obstacles in the real world need to be modeled. Instead the worker can place the hologram at a suitable position, inspect the resulting robot trajectory (which is displayed as a hologram as well) and verify it in situ. This ensures that possible collisions can be detected even before the robot starts moving. As the trajectories can be generated automatically on the device without the worker being required to verify the robot trajectories within a dedicated simulation environment, programming the robot is greatly facilitated. As a matter of fact operators can focus more on their actual tasks and do not have to expend time on programming each individual robot action. Another aspect that requires adaptation is the display of additional data relevant to the operator. In addition to the color coding, it may be useful to display relative or absolute values of the applicable force in order to provide the testing expert with accurate feedback on the performance of the system in the desired configuration. As is always

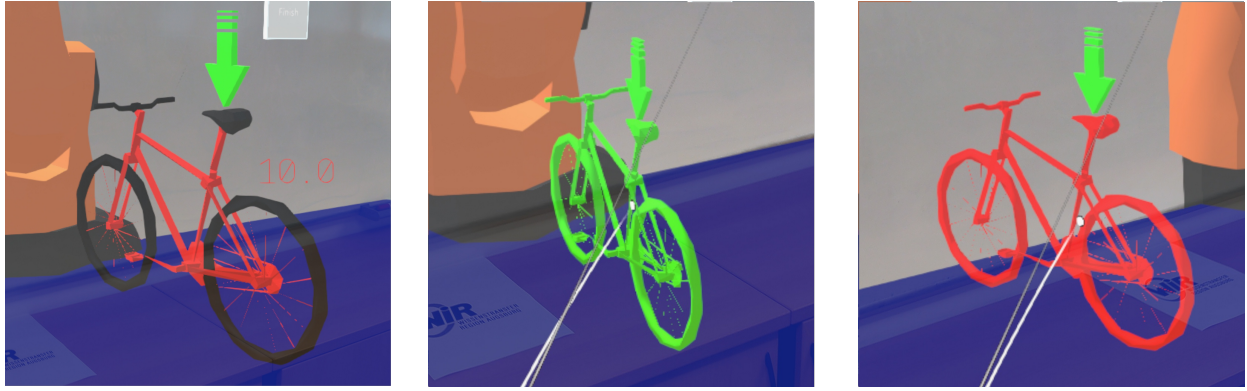


Figure 3 Different colors are a simple way to indicate whether enough force can be applied (green) by the robot or if the position is not suited for testing the specimen (red). In contrast: The specimen while not interacting with it (left picture).

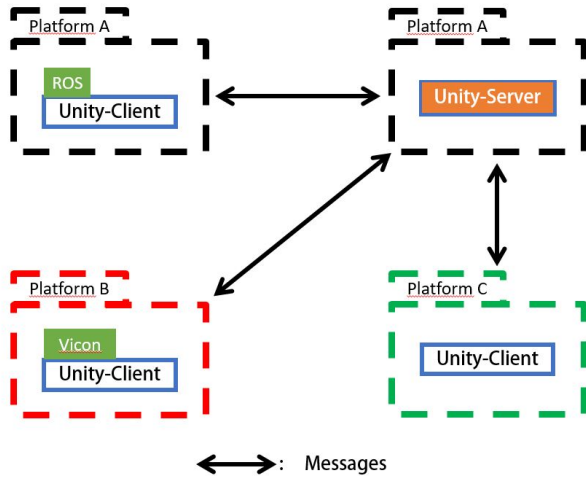


Figure 4 The overall architecture and its deployment to various systems

the case in the design of user interfaces, a balance must be struck between the wealth of information that can be displayed and the operator’s capability to ingest said information. Particularly in such an expert-focused use-case, a high degree of configurability for the individual operator is key.

6 Architectural Overview

In this section we describe the concepts of the underlying software architecture, which is under continuous development at our research institute. Currently, there exist a number of projects, which have successfully combined ROS or ROS2 with the HoloLens [13, 14] or other Augmented Reality Devices [15]. Their goal is to bring classic approaches to program industrial robots via ROS using Rviz and Gazebo into the real world by transferring them into the Mixed Reality space. One of the bigger issues we encountered was the missing native support for the Universal

Windows Platform by ROS. Therefore, many components must be ported to the HoloLens to make it work, which some of the projects mentioned above did.

Our approach does not focus on porting ROS or ROS2 to a specific platform but rather using and expanding Unity’s networking capabilities and its wide range of supported platforms to our advantage.

Figure 4 gives a high level overview of our approach. It consists of various Unity clients connected with one Unity server. Each client represents either some sort of interaction device (like the HoloLens) or a datasource (e.g. for ROS). By building upon Unity’s platform independent network layer, messages can be exchanged between multiple Unity clients even though the specific deployment platforms are different. We also enabled the separation of each Unity client or server into a separate project, which decreases the overall project size and improves maintainability. This makes it possible for us to utilize the popular ROS# project [16] within Unity and stream the relevant data to a device which neither supports ROS nor ROS# natively (like the HoloLens). This decreases the load on the respective mobile device, allowing the focus of its computing power on the presentation of data and evaluation of user input.

6.1 Preliminary Evaluation

To test the performance of our implementation, as well as its functionality, this system has previously been used in a student thesis to stream Vicon data (a popular and precise motion capturing system) to a HoloLens. We also performed quantitative analyses, in which the average round trip time from client to server was measured under a certain throughput.

The test bench consisted of two computers that were connected via LAN using a network switch. The client sent 50 messages per frame to the server, which returned them to the client. Each of these messages had a size of about 750 bytes and variable content to better approximate real-world conditions. To ensure transmission quality we chose a channel of the Unity Networking Layer, which takes care

of a reliable delivery (to the best of our knowledge this is a reliable UDP connection). The average measured round-trip time we encountered was 15ms. This is mainly due to the fact that we are bound by the Unity update cycle (60 times per second on our client), since we are using the Networking Layer of Unity, which means we can only retrieve messages from the server during processing of the next frame.

7 Conclusion & Outlook

We presented a novel approach for improving robot-based component testing by integrating mixed reality. The greatest advantages are: (1) Removing the need to maintain a separate simulation of the real-world test bench by enabling the integration of simulated specimen into the real world. (2) Different levels of support, depending on the type of assistance the test engineer requires. (3) Faster iterations when setting up new component tests or testing multiple robot motions.

Our future work includes completing the implementation of all three levels of support and evaluating our presented approach once all levels are integrated. Furthermore, we are currently evaluating the static force analysis model in order to safely operate the testing facility and its robots.

8 Acknowledgements

This paper presents, in part, results of the research project "WiR Augsburg – Wissenstransfer Region Augsburg", which is funded by the Federal Ministry of Education and research (BMBF) within the program "Innovative Hochschule".

9 Literature

- [1] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," in *Telemanipulator and telepresence technologies*, vol. 2351, 1995.
- [2] M. Filipenko, A. Angerer, A. Hoffmann, and W. Reif, "Opportunities and Limitations of Mixed Reality Holograms in Industrial Robotics," in *Workshop on "Factory of the Future", 2019 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2019.
- [3] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pre-gizer, and A. Raatz, "Intuitive robot programming using augmented reality," in *Procedia CIRP*, vol. 76. Elsevier B.V., jan 2018, pp. 155–160.
- [4] M. Rudorfer, J. Guhl, P. Hoffmann, and J. Kruger, "Holo Pick'n'Place," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2018-September. Institute of Electrical and Electronics Engineers Inc., oct 2018, pp. 1219–1222.
- [5] S. M. Chacko and V. Kapila, "Augmented Reality as a Medium for Human-Robot Collaborative Tasks," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, oct 2019, pp. 1–8. [Online]. Available: <https://ieeexplore.ieee.org/document/8956466/>
- [6] M. Kapinus, V. Beran, Z. Materna, and D. Bam-busek, "Spatially Situated End-User Robot Programming in Augmented Reality," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2019*. Institute of Electrical and Electronics Engineers Inc., oct 2019.
- [7] F. Muhammad, A. Hassan, A. Cleaver, and J. Sinapov, "Creating a Shared Reality with Robots," in *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2019-March. IEEE Computer Society, mar 2019, pp. 614–615.
- [8] F. Ghiringhelli, J. Guzzi, G. A. Di Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Interactive Augmented Reality for understanding and analyzing multi-robot systems," in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., oct 2014, pp. 1195–1201.
- [9] B. Siciliano, L. Sciacicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [10] "Unity real-time development platform | 3d, 2d vr & ar engine," <https://unity.com/>, (Accessed on 09/24/2020).
- [11] "Getting started with vuforia engine in unity," <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>, (Accessed on 09/24/2020).
- [12] "Welcome to mrtk | mixed reality toolkit documentation," <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/WelcomeToMRTK.html>, (Accessed on 09/24/2020).
- [13] "ARViz: Augmented Reality Visualizer for ROS2 – ROSIN." [Online]. Available: <https://rosin-project.eu/ftp/arviz>
- [14] "Rvis2AR - Visualization platform for AR / VR devices – ROSIN." [Online]. Available: <https://rosin-project.eu/ftp/rvis2ar>
- [15] "ROS2AR – ROSIN FTP with AR smart glass using Android." [Online]. Available: <https://rosin-project.eu/ftp/ros2ar>
- [16] "GitHub - siemens/ros-sharp: ROS# is a set of open source software libraries and tools in C# for communicating with ROS from .NET applications, in particular Unity3D." [Online]. Available: <https://github.com/siemens/ros-sharp>