Towards Human-Robot Collaboration: An Industry 4.0 VR Platform with Clouds Under the Hood

Bálint György Nagy[1], János Dóka[1,2], Sándor Rácz[3], Géza Szabó[3],
István Pelle[1], János Czentye[2], László Toka[1], Balázs Sonkoly[1,2]
[1]*MTA-BME Network Softwarization Research Group,* [2]*Budapest Univ. of Technology and Economics,* [3]*Ericsson Research*

***Abstract*—Safe and efficient Human-Robot Collaboration (HRC) is an essential feature of future Industry 4.0 production systems which requires sophisticated collision avoidance mechanisms with intense computation need. Digital twins provide a novel way to test the impact of different control decisions in a simulated virtual environment even in parallel. In addition, Virtual/Augmented Reality (VR/AR) applications can revolutionize future industry environments. Each component requires extreme computational power which can be provided by cloud platforms but at the cost of higher delay and jitter. Moreover, clouds bring a versatile set of novel techniques easing the life of both developers and operators. Can these applications be realized and operated on today's systems? In this demonstration, we give answers to this question via real experiments.**

I. Introduction

Future Industry 4.0 production systems are expected to be flexible, scalable and easily reconfigurable in order to support universal manufacturing processes, product-independency and demand driven capacity scaling. Human-Robot Collaboration (HRC) and collaborative robots are key enablers of these systems combining human's craftmanship and cognitive skills with strength and precision of robots [6]. Unfortunately, there is a trade-off between safety and productivity, and today's standards can easily lead to degradation in efficiency. The movement of the robots directly impacts the productivity, therefore, the sophisticated control of robot trajectories is an essential function which can result in increased productivity while meeting safety requirements. In order to ensure safety, *collision avoidance* is a core function which detects potential collision while executing a given trajectory and calculates detours. Instead of moving physical robots, *digital twins* provide a novel way to test the impact of different control decisions in a simulated virtual environment. By these means, several control options can be calculated and tested in parallel with future consequences. In addition, Augmented/Virtual Reality (AR/VR) bring a novel set of applications which can revolutionize future factories. These emerging technologies enable novel use-cases including enhanced ways of collaboration, efficient training processes and improved productivity. A central element of AR/VR applications is the engine managing the 3D virtual space. Different *3D engines* are available with different capabilities, such as Unity, Unreal, or ApertusVR. However, from Industry 4.0 aspects, these options have several limitations. Gazebo [3] is an open-source physical simulation framework supporting multiple engines. For example, ODE (Open Dynamics Engine) is a high performance library for simulating rigid body dynamics which is supported by the Robot Operating System (ROS) [5]. These capabilities make it a potential component of future Industry 4.0 applications.

Both collision avoidance and the 3D engine may require extremely large computational capacity and the results should be provided within strict latency bounds. Cloud platforms can provide the required computational power together with novel techniques, such as cloud native programming and serverless architectures, but at the cost of higher delay and jitter. Can these applications be realized and operated on today's systems? Can industrial robots and VR experience tolerate the additional delays? In this demo, we give answers to these questions.

978-1-7281-2700-2/19/$31.00 2019 © IEEE

II. Main ideas

On the one hand, the computation load generated by Gazebo significantly depends on the events in the virtual space. For example, if an object is moved, several computations are triggered calculating the collisions and interference of constituent objects. The current version of ODE supports multi-threading which can be exploited during the physical simulations if the interactions among different parts are minimal. If we dynamically adjust the allocated CPU cores under e.g. the container running Gazebo depending on the current demand, the operation cost can be reduced with better resource utilization. On the other hand, collision detection and detour trajectory calculations comprise a lot of functions which can be run in parallel. Well designed cloud native programs can gain the benefits provided by public/private Function as a Service (FaaS) platforms. A set of functions is to be turned into stateless and synchronized via other solutions (e.g. making use of data stores). This will introduce additional delay and jitter which should be analyzed to evaluate the feasibility of this approach in an industry environment.

III. Our platform

We have integrated Gazebo with VR applications implemented in the Unity framework. By these means, the traditional Unity framework is extended with a high performance VR engine meeting the requirements of Industry 4.0 applications. In addition, we provide a HRC component responsible for collision avoidance controlling UR10 robotic arms in a virtual space (borrowed from the "Agile Robotics for Industrial Automation Competition"). Our software stack is built on

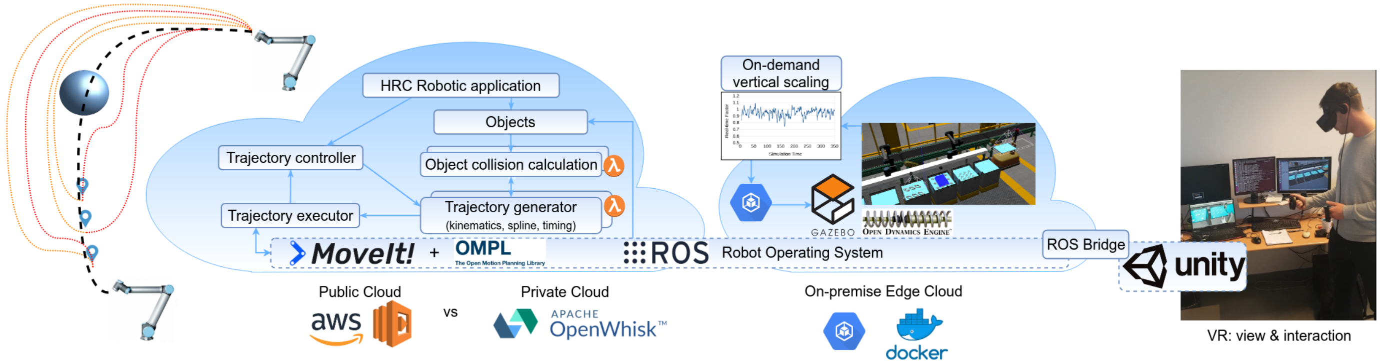

Figure 1: Our cloud-based Virtual Reality platform for Human-Robot Collaboration

MoveIt [4] (running on top of ROS) and OMPL [8] providing the functions to calculate and manipulate robot trajectories. More specifically, available software modules were adjusted to turn them into cloud native, and novel components were implemented to realize the missing pieces of the application and to coordinate the cloud based operations. The integrated system is deployed on different cloud platforms exploiting vertical and horizontal scaling features and the feasibility of the approach is evaluated. The high level architecture of our VR platform is shown in Fig. 1.

The left part implements a robotic application controlling a complex production environment containing two robotic arms, conveyor belts and other movable objects. Here, we focus on the control of the robotic arm. The top level application commands the arm to move to a given position via a trajectory controller which invokes a trajectory generator to calculate an appropriate one. Different constraints and goals can be given impacting the final path which is implemented by the cooperation of the trajectory executor and the low level controllers of the robot (running in ROS). The collaboration with human workers or other dynamic events could require to revise the current trajectory. If an object appears on the way of the robot arm, a detour should be calculated and executed. Furthermore, rapidly changing environments need to have several detour options to be calculated in advance and made available. In our system, when a potential collision is detected, the calculation of several detours is initiated from different starting points (indicated by POI symbols in Fig. 1) on the currently executed trajectory. From a given starting point, multiple routes with varying characteristics are calculated and potential collision is checked by a dedicated object collision calculation function. The latter calculation is carried out for all segments of the given path and colliding routes are rejected. The performance of the overall robotic cell is significantly affected by the quality of the detours and the response time of the platform, i.e., when the detours become available. (The robot needs the result before reaching the corresponding starting point.) We argue that several functions of the robot control can be operated on (edge) cloud platforms in parallel if the latency requirements can be met. Here, the latency stems from the networking delay (the software components are geographically distributed) and from the operational overhead of the cloud platforms which can be significant. For example, in [7] we analyzed the delay characteristics of different operations of the AWS Lambda framework. Amazon's AWS Lambda [2] and Apache OpenWhisk [1] are investigated as public and private cloud solutions, respectively, with various deployment options.

The center part (of Fig. 1) operates Gazebo and the ODE simulation engine in a Docker container. The real-time factor, which directly characterizes the quality of the users' experience, is measured on-the-fly and if it falls bellow a threshold, additional CPU cores are allocated to the container. In contrast, reaching an upper threshold results in releasing a given fraction of CPU cores.

The right part corresponds to our Unity-based demo environment where users can interact with the system, e.g., objects can be moved triggering detour calculation and modified movement of the robotic arms.

During the demo, we showcase how trajectory control and Gazebo's 3D engine can exploit the features of available cloud platforms, such as on demand vertical scaling and inherent horizontal scaling provided by FaaS platforms; and demonstrate the impact of delays caused by different deployment options. Learn more:
https://sb.tmit.bme.hu/mediawiki/index.php/HRC.

Acknowledgement

This work was supported by National Research, Development and Innovation Office (NKFIH) under the research and development project in Hungarian-Korean cooperation (2018-2.1.17-TÉT-KR-2018-00012).

References

[1] (2019) Apache OpenWhisk. https://openwhisk.apache.org.
[2] (2019) AWS Lambda. https://aws.amazon.com/lambda.
[3] (2019) Gazebo. http://gazebosim.org.
[4] (2019) MoveIt: Moving robots into the future. https://moveit.ros.org.
[5] (2019) Robot Operating System. http://www.ros.org.
[6] K. Dröder *et al.*, "A machine learning-enhanced digital twin approach for human-robot-collaboration," *Procedia CIRP*, vol. 76, pp. 187–192, 2018.
[7] I. Pelle, J. Czentye, J. Dóka, and B. Sonkoly, "Towards Latency Sensitive Cloud Native Applications: A Performance Study on AWS," in *IEEE International Conference on Cloud Computing, CLOUD 2019*, 2019.
[8] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012.