TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
35/2019

Novel Digital Twin Development Methodology for the Robot Cell Connectivity in a Smart Industry Environment

VLADIMIR KUTS

TALTECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Mechanical and Industrial Engineering
This dissertation was accepted for the defence of the degree 10/05/2019

Supervisor:	Assoc. Prof. Toivo Tähemaa School of Engineering Tallinn University of Technology Tallinn, Estonia
Co-supervisor:	Prof. Tauno Otto School of Engineering Tallinn University of Technology Tallinn, Estonia
Opponents:	Jeremy A. Marvel, Ph.D National Institute of Standards and Technology Engineering Laboratory, Intelligent Systems Division U.S. Department of Commerce
	Dr. Geza Haidegger MTA SZTAKI Institute for Computer Science and Automation Academy of Sciences

Defence of the thesis: 21/06/2019, Tallinn

Declaration:
I hereby declare that this doctoral thesis, my original investigation and achievement, submitted for a doctoral degree at the Tallinn University of Technology, has not been submitted for a doctoral or equivalent academic degree.

Vladimir Kuts

----
signature

European Union
European Regional
Development Fund

Investing
in your future

ISSN 2585-6898 (publication)
ISBN 978-9949-83-442-6 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-443-3 (PDF)

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
35/2019

Uudne digitaalsete kaksikute arendusmetoodika robottootmisrakkude sidustamiseks targa tööstuse keskkonnas

VLADIMIR KUTS

TALTECH KIRJASTUS

Preface

My current doctoral studies in the field of Mechanical Engineering was started in 2015 in the Department of Mechanical and Industrial Engineering, at the Tallinn University of Technology. My previous background included a BSc grade in Mechatronics and an MSc in Production Engineering, supplemented by work experience in various ICT and industrial companies. I have always been interested in Robotics and novel ICT technologies, so my interest in the field and previous background have helped me to start my studies as a PhD student under the supervision of Prof. Tauno Otto and Assoc. Prof. Toivo Tähemaa at the same time that I was working in an industrial company. The research included Industrial Robots and their usage in manufacturing for the purpose of inspection, measurement and quality control. During this research, I saw that Industrial Robot control and management could be done in a much easier way with the usage of various IT technologies and software tools. Knowing that, I quit work in the industrial sector, then joined TalTech University as a full-time researcher and lecturer. I started to work in the field of Virtual Reality tools development for the manufacturing industry with a focus on Robotic Cells. During my studies and work, I have given lectures in courses of digitalization as well as participated in and led projects. Moreover, one semester I have spent in CNR-STIIMA in Milano, Italy under the supervision of Dr. Marco Sacco and Prof. Walter Terkaj, where work on the synchronisation of virtual and real manufacturing cells was initiated. My PhD work combines knowledge in the robotics and visualization fields, and all the related projects were done in Estonia and Italy. To conclude, my doctoral study time was the most exciting experience in my work and study life, where I have been given the opportunity to do what is interesting for me and necessary for industry as well as meeting a lot of intelligent and interesting people from the related fields of my work.

Contents

List of Publications

The list of the author's publications, on which this thesis has been prepared, is as follows:

I Kuts, V.; Tähemaa, T.; Otto, T.; Sarkans, M.; Lend, H. (2016). Robot manipulator usage for measurement in production areas. Journal of the Machine Engineering, 16 (1), 57−67.

II Kuts, V.; Sarkans, M.; Otto, T.; Tähemaa, T. (2017). Collaborative work between human and industrial robot in manufacturing by advanced safety monitoring system. In: B. Katalinic (Ed.). DAAAM International (0996−1001). Vienna, Austria: Curran Proceedings.

III Kuts, V.; Modoni, G. E.; Terkaj, W.; Tähemaa, T.; Sacco, M.; Otto, T. (2017). Exploiting factory telemetry to support Virtual Reality simulation in robotics cell. Augmented Reality, Virtual Reality, and Computer Graphics, 1: 4th International Conference, AVR 2017, Ugento, Italy, June 12-15, 2017. L. Tommaso De Paolis, P. Bourdot, A. Mongelli (Ed.). Springer, 212−221. (Lecture Notes in Computer Science; 10324).

IV Kuts, V.; Otto, T.; Tähemaa, T.; Bukhari, K.; Pataraia, T. (2018). Adaptive industrial robots using machine vision. Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition: IMECE2018, November 9-15, 2018, Pittsburgh, PA, USA. ASME,

V Kuts, V.; Otto, T.; Tähemaa, T.; Bondarenko, Y. (2019). Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality; Journal of the Machine Engineering, 2019.

VI Kuts, V.; Modoni, Gianfranco E.; Otto, T.; Sacco, M.; Tähemaa, T.; Bondarenko, Y.; Wang, R. (2019). Synchronizing physical factory and its Digital Twin through an IIoT middleware: a case study. Proceedings of the Estonian Academy of Sciences, 2019.

Publications copies are enclosed in the appendices.

Author's Contribution to the Publications

The contribution to the papers in this thesis are:

I The author performed an investigation and comparison of industrial robot usage for measurement, performed the experimental part regarding a laser scan, as well as the main writing.

II The author performed an investigation and comparison of Industrial Robots offline as well as online programming and performed the main writing.

III The author participated in the design of the framework of the connectivity described use-case and performed the main writing.

IV The author participated in Machine learning in robotic cells using Digital Twin tools and performed the main writing.

V The author participated in the design of the case study, analysed the results and performed the main writing.

VI The author performed the use case model, the analysis of the results as well as the writing process.

List of Abbreviations

AGV	Autonomous Ground Vehicle
AI	Artificial Intelligence
AM	Autonomous Robots
API	Application Programming Interface
AR	Augmented Reality
BD	Big Data
CPS	Cyber-Physical System
DESI	Digital Economy and Society
DT	Digital Twin
ERF	European Regional Development Fund
FMS	Flexible Manufacturing Systems
FT	Factory Telemetry
GUI	Graphical User Interface
HRC	Human-Robot Collaboration
ICT	Information and Communications Technology
IDC	International Data Corporation
IFR	International Federation of Robotics
IIoT	Industrial Internet of Things
i4.0	Industry 4.0
IoT	Internet of Things
IR	Industrial Robot
LoD	Level of Details
PC	Personal Computer
ROS	Robot Operating System
RQ	Research Question
SmartIC	Smart Industry Centre
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UI	User Interface
VC	Virtual Commissioning
VF	Virtual Factory
VR	Virtual Reality
XR	Mixed Reality

1 Introduction

1.1 Background and Research Gaps

Created by European Commission Digital transformation monitor aim to promote and monitor the status of Digital Transformation in Europe [1]. Key technologies being monitored are 3D printing, Autonomous cars, Mobile and Mobility, Internet of Things (IoT), Artificial Intelligence (AI), Blockchain, Augmented (AR) and Virtual Reality (VR).

Estonia ranks 9-th out of the 28 Member States in the European Commission's Digital Economy and Society Index (DESI) 2017 [2]. Estonia is the leading in Europe in the online public services and scores above the EU average in digital skills and the average usage of the internet by citizens. However, in 2017 Estonia ranked only 20 out of 28 concerning the integration of digital technology into other sectors, its weakest score out of the five dimensions of DESI. Therefore, the critical challenge in Estonia is the digitisation of industries, which also concerns the digitalisation of manufacturing – Estonia is only developing itself towards Industry 4.0 (i4.0). However, developing fastly, due to small and innovative-minded population, with high skills and experience in information and communications technology (ICT) sector, Estonia has the potential to integrate ICT innovative solutions quickly into various fields, especially in manufacturing domain.

To support this initiative in 2017 by the European Regional Development Fund (ERF) were funded Smart Industry Centre (SmartIC) project, which focuses areas are shown in Figure 1.

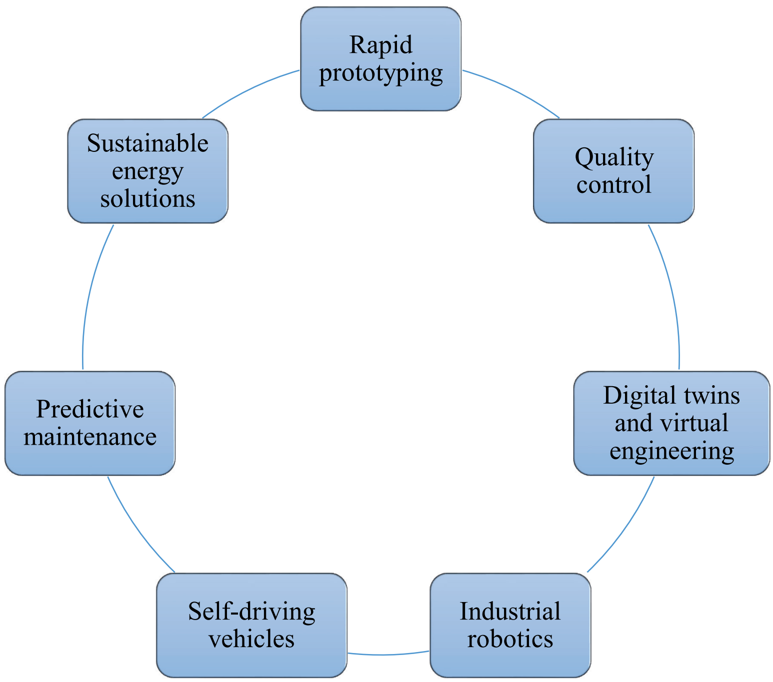


Figure 1 – Areas of focus of SmartIC

Visualised on pictures areas of SmartIC focuses mainly on next enabling technologies:

- Preparation, manufacturing and quality control of complex;
- Development of Digital Twins (DT) and VR and AR solutions in manufacturing;
- Industrial robotics;
- Self-driving vehicles in production logistics;
- Predictive maintenance and optimization smart manufacturing;
- Safe and sustainable energy solutions for smart industry

Those topics, especially Industrial Robotics and DT, VR solutions are crucial parts of the related dissertation.

First essential domain – Industrial robots are widely used manufacturing equipment in novel manufacturing. Industrial robotics is quickly becoming an important technology that is helping enterprises to innovate while improving business performance. It is an interdisciplinary technology that links mechanical engineering, electrical engineering and information technology. Different methods of programming and integration are used to add those to the manufacturing process flow to automate [3, 4]. Three methods of programming are mainly used for IR-s [5]:

- Offline method – every large manufacturer has its software tool for programming a robot. Every step can be simulated and performed, then downloaded to the robot controller.
- Manual method – usage of flex-pendant by the operator gives an intuitive and real-time feeling of what is happening with the machine.
- Online method – robot controller connected to the external control unit, which enables one to affect robot movement online, interrupting or re-writing the task performed by the robot.

The metric “robot density” has been adopted by the International Federation of Robotics (IFR) to track this tipping point in modern manufacturing. For reference, in 2018 there was a global average of 74 robotic units per 10000 workers, up from 66 units per 10000 workers in 2015. Europe alone has about 99 robotic units per 10000 workers [6]. In this report, Estonia is ranked 38th with 11 robotic units per 10000 workers [6]. The share of IR-s by performance evaluation has been analysed more thoroughly in [7].

Moreover, the human-robot collaboration (HRC) ISO standard [8] controls and regulates these different methods in a way for safe and reliable co-work between the operator and the machine [9] Automation supportive machines such as IR-s must work efficiently for the manufacturing process and safety for the human factor [10].

The IoT refers to the network of smart, interconnected devices and services that are capable of sensing or even listening to requests or needs, and are able to intelligently act on them, with the methods to collect, store and manage different data and the tools to define and train AI algorithms. The Industrial Internet of Things (IIoT) – a subset of the more massive IoT – focuses on the specialised requirements of industrial applications, such as manufacturing. IIoT systems are still not widespread across companies. It is estimated that European IoT is currently responsible for around 30% of connections worldwide. A recent study by Accenture also projected that within the next 8-10 years, the market for IoT solutions would be worth €80 billion, with the potential value for the EU28 that could reach nearly €1 trillion [11]. Based on this information and Literature Review in Chapter 2 were found shortcomings which are formulated as the first research gap.

Research gap 1: There is a number of main significant problems in the HRC concept:

- *In most cases, there is a lack of human safety control and fault monitoring in real-time systems;*
- *The downtime of industrial robots is high due to human interaction with it (here considering heavy machinery industrial robots – not co-bots). The reason is the low intelligence level of the safety system. A small disturbance causes a full halt (stop) of the system and rebooting is slow and ineffective;*
- *Robots do not adapt to external factors such as unexpected movements caused by third parties and do not make autonomous decisions between given tasks;*
- *The lack of interoperability: Interoperability is critical to maximising the value of IIoT. A widespread breakthrough for IIoT systems requires a fully connected global ecosystem, which is only possible through interoperability across systems and regions. Recent work by Manyika et al.* [12] *claimed that IoT value chains could increase their value added by 40% when different IoT systems are linked together.*

In addition, simulation software is widely used to plan IR cells and factories in general layouts as well as to plan and modify factory shop floors with the creation of a DT [13], which is a virtual representation of the real equipment of the factory, supported by management and configuration tools for optimization and simulation purposes, while at the same time receiving real-time data from the real factory.

The development of this digital representation of a real manufacturing cell can be integrated and used to support different system planning and executing operations within the factory layout, such as to simulate and optimise based on data received of the factory's overall performance. For example, the completed simulation can allow production workers and managers to perform changes in the factory manufacturing unit without any harm to production efficiency. In addition, representable of the manufacturing unit can plan change and build a full factory behind hardware such as Personal Computer (PC) without a need to interact with the real environment of the factory. Besides, VR tools can be enabled in the system to safely educate, collaborate with others, inspect and plan manufacturing systems on a 1 to 1 scale. Based on this information and a more deep investigation, which is described in Chapter 2, was found the second crucial gap in this area of research.

Research gap 2: There is a lack of DT design methods with enabling toolkit supporting the interaction of the real environment and enabling a full connectivity synchronisation framework between real and virtual environments, thus allowing for two-way connectivity between various assets in the two worlds, enabling full remote control over the manufacturing layout.

1.2 Research Problems and Framework

IR cells (one workplace equipped with an industrial robot) in modern manufacturing demand a various set of skills and the presence of a human operator near the robotic cell or operating room to re-program, upload software to the robot controller and make changes in the equipment set-up. Multifunctional and adaptive to external factors, decision methods made by the robot itself are not used to maintain operator and environmental safety, which leads to robot downtime for re-programming. Moreover, DT designing tools are mainly used only by the simulation's purposes in the virtual environment. Based on this and the previous chapter of background research and gaps, this dissertation's research problem is described as follows:

There is a lack of studies in design methods for DT, which allows for precise real-time synchronisation and connectivity between virtual and real environments, which could enable one to control, teach and manage IR cells from virtual environments remotely in real-time.

However, while DT concepts and tools have already been developed and used by larger manufacturers for factory layout design, updates and efficiency simulations, there are still challenges and questions of how to develop for the factories of the future, automate processes of digitalisation and why it is needed at all. As many manufacturers run experiments on the factory shop floor, they do not evaluate the need for novel technologies for just simulations and layout design. It is a challenge to promote and prove that DT can increase efficiency and reduce the downtime of the IR-s, because of the analysis and optimisation tools embedded in DT tools not used widely - it is one of the aims of the related study, more of which is described in Chapter 3. AR and VR technologies will directly influence a more significant number of European industries ranging from cultural and creative industries, manufacturing, logistics, defence, robotics and healthcare to education, training, entertainment, and the media, thus enabling new business opportunities. The International Data Corporation (IDC) estimates that the VR/AR market was worth about 24 billion euros in 2018 and will reach 183 billion euros by 2022 [14]. VR/AR may not be critical technologies in themselves, but as new computational interfaces, they may enable new applications and experiences in Engineering [15].

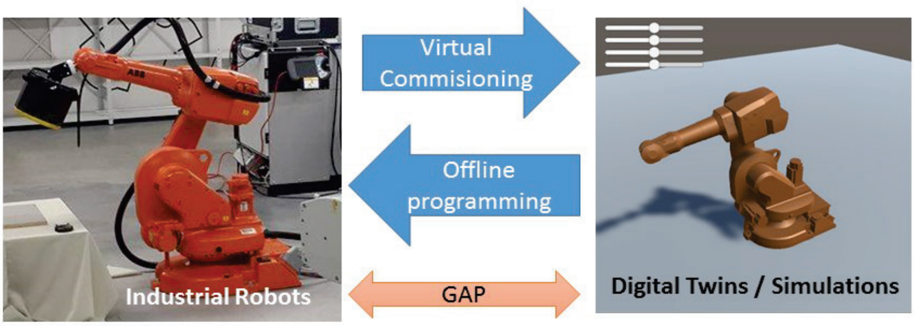


Figure 2 – Industrial robots and DT relations – the gap in precise online synchronisation

The main problem is visualised in Figure 2, which shows the present IR and DT concept relations. Even if the data from real industrial machines can be used to feed simulations and simulation tools used as offline programming tools for the robots, there is still a gap related to the lack of design methods for the management of IR cells. In the scope of digitalisation and paradigm of i4.0, IR should not just feed each other with data, but enable precise online synchronisation between real and virtual worlds online.

1.3 Objectives and Research Questions

This thesis aims to develop and test novel DT design methods to create concepts and toolkit, which enables full control and synchronisation between real and virtual worlds, leading to an increase of efficiency of the robotic cell via the reduction of downtime while re-programming it. The focuses to achieve it are next:

- To test various IR-s programming methods and provide a concept for collaborative, adaptive and safe for people IR cell.
- To research factory telemetry and provide an architecture and framework for Virtual and real factory connectivity.
- To provide a concept of design methods and a tool for the design of fully synchronised with real IR cell DT in provided use-case.

The dissertation intends to provide researchers, IR-system integrators and manufacturing units, design methods and toolkit for creating a safe and precise DT of their IR cell unit in a fast and quality focused way. Moreover, it facilitates how various assets, both virtual and real, can be monitored using collected data integrated and combined via a middle-layer platform to create a fully synchronised DT with its original source equipment, enabling enlargement of equipment functionality and reduction of downtime while re-programming. Therefore, the next research questions (RQ) are answered in this work:

RQ1: What are the options in evaluating the optimal and timely effective way to program an industrial robot? Which is a flexible way to do it?
RQ2 How can a concept of a safe human-robot collaboration environment be made and tested safely?
RQ3: What factors must be considered while designing an "ideal" DT, and which tools and methods shall be used?
RQ4: How to enable precise online synchronisation and connectivity through IIoT tools to make an industrial robot cell fully re-programmable in real-time via DT?

The research questions are answered in the following six articles:

Article 1 is a study that had input for RQ1. In this article, industrial robot programming was performed through different methods
Article 2 is a study that responded to RQ1 and RQ2, and made a comparison between different programming methods as well as introduced a safe HRC concept for further tests.
Article 3 is a study that responded to RQ3. In this article, methods of connectivity between DT and a real industrial robotic cell were evaluated and which factors must be considered in DT design were answered.
Article 4 supported answers to RQ2 and RQ3 combining the adaption of the industrial robotic cell with the help of DT.
Article 5 responded to RQ4 by means of the development of a fully synchronised use-case and middle layer for DT.
Article 6 is a study that supported Article 5 in the development of an IIoT platform and integrating it to the middle layer described in Article 5.

1.4 Research Process and Framework of the Dissertation

Research done during the related study combines different methodologies and known methods. IR programming and robot safety in combination with the DT concept with precise synchronisation are critical topics of the current study. A literature review and development of concepts for case studies and their execution are the main techniques in this experiment-based research. The developed concept of exploiting the factory telemetry approach includes research on IR cell programming methods and different connectivity methods of Industrial IIoT developments. The developed approach enables an accurate synchronisation between the IR cell and its DT. A framework was created for connectivity, to allow for the exploitation of factory telemetry, DT, IR online programming and to use the developed toolkit for future research and use-cases in an environment of Smart Industry. The research was performed from 2015 to 2019 in the Department of Mechanical and Industrial Engineering at the Tallinn University of Technology. During February-May in 2017, research was supported by an internship in the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing of the National Council of Research in Milan, Italy. The results of the PhD dissertation have been presented at several international conferences and in journals. The main platforms for publishing were the International ASME-IMECE (The American Society of Mechanical Engineers, Mechanical Engineering Congress & Exposition), DAAAM International (Danube Adria Association for Automation & Manufacturing), Springer, and the Journal of Machine Engineering sponsored by CIRP (The International Academy for Production Engineering). Figure 3 shows the structure of the chapters in the related dissertation.

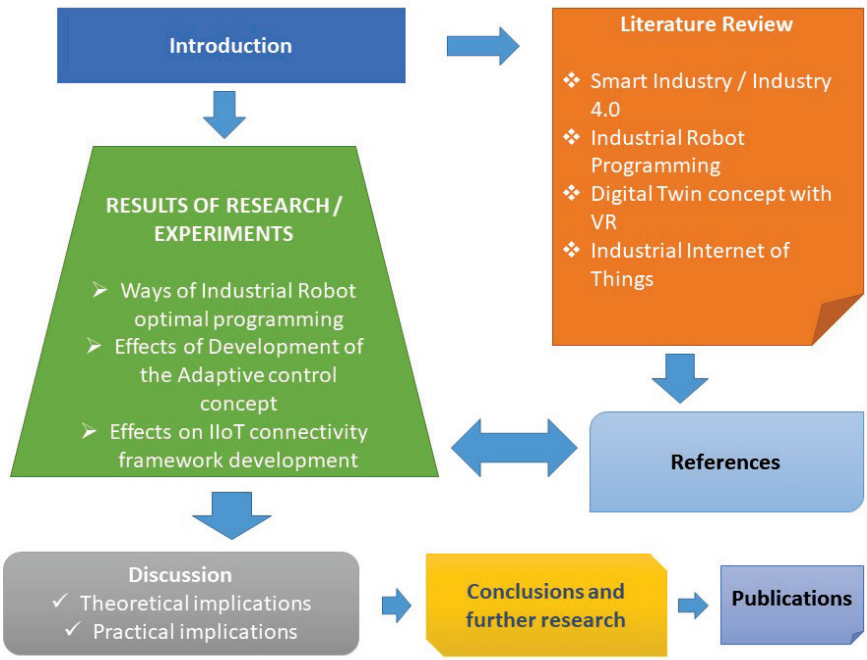


Figure 3 – The framework of the dissertation

2 LITERATURE REVIEW

The literature review combines background information and descriptions of the research area before going to the research results section. The review section helps the reader to understand the importance of the completed research in the dissertation and introduce to it.

2.1 Smart Industry/Industry 4.0

For the last four years, the actual topic in the modern manufacturing world has been the paradigm of i4.0, which means the fourth industrial revolution. One part of the triggers of this concept were general social, economic and political changes, which demanded changes in sides of the next application [16]:

- Time of development periods – innovations and updates should be integrated faster;
- Product customisation – more flexible and modular product choice;
- Flexible Manufacturing Systems (FMS) – demand for flexibility and adaption on the production line;
- Decentralisation of manufacturing control algorithms;
- Ecological aspect – reduction of resource waste.

Another side triggered way to the i4.0 revolution was a technological push with an increased level of automation, digitalisation and new networking technologies as well as the need for miniaturisation on the factory shop floor [16], which means creating more flexible and compact for on-demand production of the products.

The concept of i4.0 originally came from Germany and combines different architectures and frameworks from mainly two domains – industrial and manufacturing technologies merged with digitalisation tools. This is a combination of the above-named cyber-physical systems (CPS) [17] and stands on the nine main pillars (see Fig. 4).

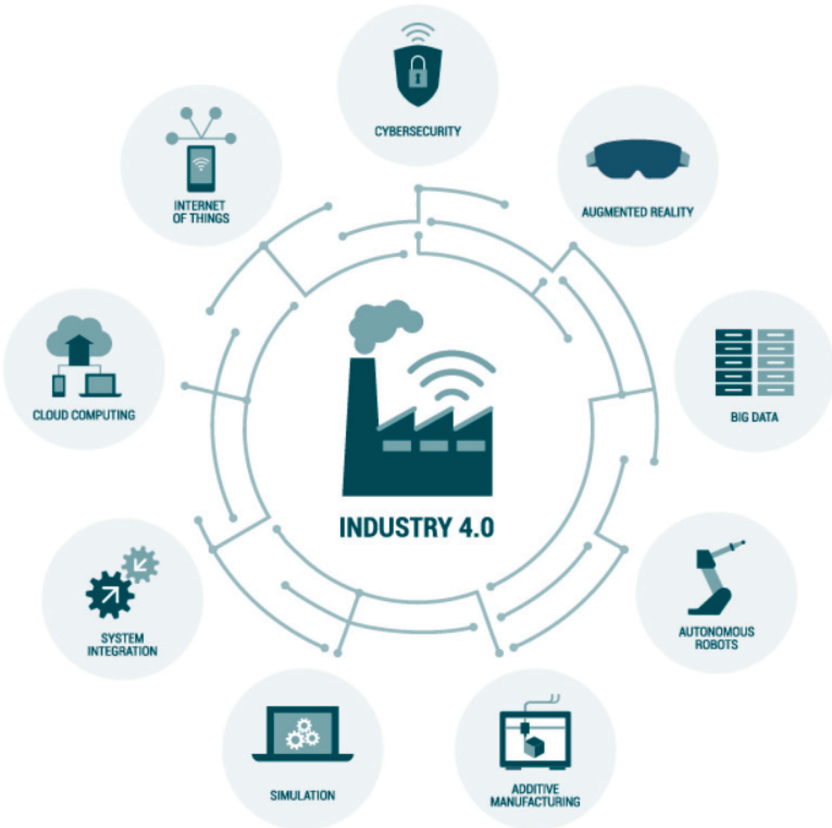


Figure 4 – Industry 4.0 visual schematic [17]

As it was already mentioned – there are nine main pillars on which i4.0 stands [17], [18]. As seen in Figure 4, they are as follows:

- IoT in the scope of i4.0 is transferred to IIoT, which means the connection between every asset of virtual and real equipment or tool of the factory;
- Cybersecurity – when it comes to automation and digitalisation, an important part is the protection of data and connectivity, so only those who are supposed use it can use it;
- Autonomous Robots (AR) are essential when it comes to fast troubleshooting, the training of operators with digital manuals on site and guides for workers that show what to do or where to go;
- Big Data (BD) – as it is vital to send data from cameras, sensors, lasers, etc., there must be expertise in how to manipulate this data and not get lost in its volume. AI is crucial in the support of data analysed in this domain;
- Autonomous Robots (AR) – for logistics and assembly usage to automate processes where mainly people are involved;
- Additive Manufacturing – more customizable production of smaller batches of products with the usage of 3D printers;

- Simulation – DT of manufacturing layout done in the virtual world help to optimise and analyse updates for the factory before integrating them;
- System Integration – many parts are not connected, while the integration of different software and hardware assets into one create a single integrated system, in which every part of manufacturing can be tracked and controlled;
- Cloud Computing – the usage of nonphysical storage to reduce hardware usage. All calculations are analysed, and optimisation can be done from non-physical databases and storage facilities.

Considering all the related topics mentioned above and visualised before, all research mainly done nowadays on robotics, IoT, data analysis, human-robot interaction, etc., is research somehow related to I4.0 as all those domains are covered in this paradigm.

However, researchers are also working now on an overall i4.0 concept classification, optimisation and related fields investigation. They are also creating conceptual frameworks, scenarios and researching the meaning of smart manufacturing [19]. Furthermore, there is the standardisation of new aspects of production, where importance is also on safety, both physical and cyber. As creation and development should be done in one direction – standards of creation are crucial so that incompatible parts are not created [20]. Moreover, compliance and a hierarchical, semantical model must be considered as everything should be connected with IIoT protocols, data flow from different assets should be classified as well as feedback from components should be received and categorised [21] in order to facilitate smart cooperation on site.

Finally, i4.0 is a very rapidly developing domain, where ICT, manufacturing and the human aspect for collaboration with the machines are combined. Systematic research and reviews are frequently released, and there is still a lot to investigate in this field [22].

A related dissertation mainly combines the described domains as IoT, Autonomous Robots and Simulations enlarged with VR technologies.

2.2 Industrial Robot Programming

In this dissertation, the primary use case used is IR cells programming through new design methods due to the previous experience of the author. The way to control IR from VR [23] and a simulation environment investigation of different types of programming methods were carried out to show the reader the importance of developed during this work approach.

Standard IR programming is performed in three different ways [24], [25], [26]:

- Manual
- Offline
- Online

This section describes the various methods in detail, approaches in research towards each and points out their advantages and disadvantages. The outcome of the comparison is shown in a table in Section 2.2.4 showing the best approach for IR programming based on a literature review.

2.2.1 Manual Programming Method

Manual programming, which can also be called the robot guiding method of an IR, is performed from a so-called flexpendant or joystick, from which commands to the robot controller are sent on operator demand. The robot can be controlled in real-time, and the robot "learns" the steps of the path on command from the operator. This process is the most straightforward way of learning for a person. To simplify programming, different aid equipment and algorithms are integrated into this process as motion capture systems [25].

2.2.2 Offline Programming Method

The offline programming method is a usage of software tools, which are dedicated to the exact brand and type of robot. The tool is very powerful for creating an IR cell simulation where pre-definition of the precise path is done. However, the tools are usually not flexible and the learning curve is tough for new operators. Different path planning and joints optimisation work is being done in research to simplify operator programming work and create auto path generation with the help of assisting software packages [26], [27], [28]. Moreover, generic software packages also exist, which support different brands of IR offline programming.

2.2.3 Online Programming Method

The online method is the generation of an IR path during its working process. It gives a possibility to adapt to the surrounding environment and re-write code online, while connected with a robot directly. Researchers use additional software packages such as MATLAB and Robot Operation System (ROS) in order to make rapidly deployable new systems integration onsite [29] as well as to make the generation of motions for faster work easier for operators [24]. However, this approach is hard for new personnel working on IR equipment as they need skills in programming, electronics, mechanical engineering and a knowledge of sensors, machine vision and learning. This demands the creation of an architecture of usage to make the online method programming more intuitive [30].

2.2.4 Industrial Robots Programming Methods Comparison

Table 1 represents a comparison between the different IR programming methods. Analysing it shows that the online programming method has more advantages compared with offline and manual mainly because of the time consumption involved. The offline method is also complex and not adaptive to online signals for surrounding sensors. However, with all its flexibility, the online method demands specific skills or classified design methods of this type of programming. Thus, this is part of a related dissertation.

Table 1 – Comparison between Industrial Robot programming methods [24], [26], [27], [28], [30].

	Manual	Offline	Online
Advantages	• Ease of usage • No need for additional software • Suitable for making simple programs • When finished, the program is ready for usage	• Dedicated environment • Precise path planning • Programming is faster than manual • The quality and flexibility of the robot programs is higher (modularity) • Enables one to simulate before real implementation (axis configuration, collision detection) • Robot downtime is reduced - while offline programming is in progress, a robot can still work with the old program.	• Flexibility in equipment choice • Adaptive by the support of sensors • The path can be overridden after the safety check on the DT during the work process
Disadvantages	• Long programming time • The precision of path-planning is low • Need for real robot for programming • The robot cannot work at the same time • Possible collisions during programming.	• Learning and using the software is difficult • Needs its own tools for each robot type • Commercial software license must be obtained before programming • Needed when the programs get more complex.	• Skills for environment set-up • Special software and equipment needed when the programs get more complex.
Time for Programming	In comparison with the offline and online methods, it is the most time-consuming. For example – one minute of robot routine program is one hour of operator programming work.	The offline method is approximately two times faster than manual programming because of the ease of the tools and access to simulation playback.	The online programming method's time depends on the equipment used and additional perception tools. Mainly, time-consumption is between offline and manual programming methods (slower than Offline but faster than Manual – in the case of a complex program).

The standards methods described and compared in the above table are also being enlarged with various interfaces according to Villani et al. [5]. There it is stated that in addition to basic Offline, Online and various approaches of manual programming, Multi-Modal interfaces are used. Those methods are meant to be people-friendly input modes such as speech, gesture, eye tracking, facial expression and haptics, which allow developed interfaces to recognise all of it in a way to effectively control the robots. Villani et al. also introduce VR and AR interfaces and the literature of related research, but they still state that it involves ongoing and future research, and that there is no final solution yet.

In this dissertation, based on the author's previous knowledge and literature in the field of IR programming methods, experiments with new IR cell design methods were introduced and performed with a combination of online programming approach algorithms and frameworks described above, merged with VR development technologies.

2.3 Digital Twin Concept with VR

2.3.1 Digital Twin

The concept of DT is a part of the digitalisation and simulation pillar of the i4.0 paradigm, which was introduced in Section 2.1 of this dissertation [31], [32], [33], [34]. Authors from different fields and universities have been researching DT for years. NASA first used virtual replicas in 1957. However, the primary contribution to the definition of DT in the scope of the i4.0 paradigm was made by Grieves [35] in 2014, which states that it is a one-to-one scale, precise virtual replica of an existing real-world asset. In the manufacturing world, DT is used for a replica of a product, manufacturing process and system, which combines both. The main difference from a regular simulation is that DT is fed with real data and often connected to real sensors or exploiting historical data [13], [36]. Moreover, a digital representation of the factory support reconfiguration and management processes through simulation, analysis of simulation results and optimisation processes based on data received are carried out. However, a simulation needs preparation, which consists of different factors in order to create a valid model for further implementation to the real manufacturing flow [37].

The model does not necessarily mean a 3D visual model, as importance is in process flow and the data behind it. Different tools are used to create this kind of model as can be seen in the flow modulation in Figure 4 and 3D visualisation in Figure 5. In the figures, process flow simulations of IR and CNC bench are visualised in two different ways. In Figure 5, it is shown only on the process level and in Figure 6, on the visual system level simulation, which can be controlled via process level input. Both of these examples are commonly used DTs.

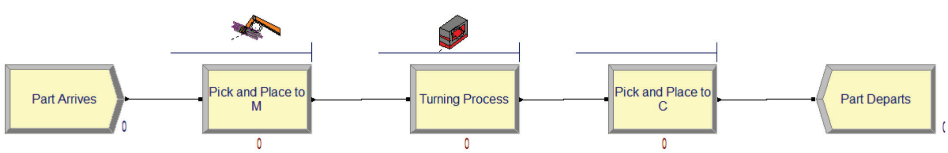


Figure 5 – Process flow simulation layout of IR and CNC bench example [38]

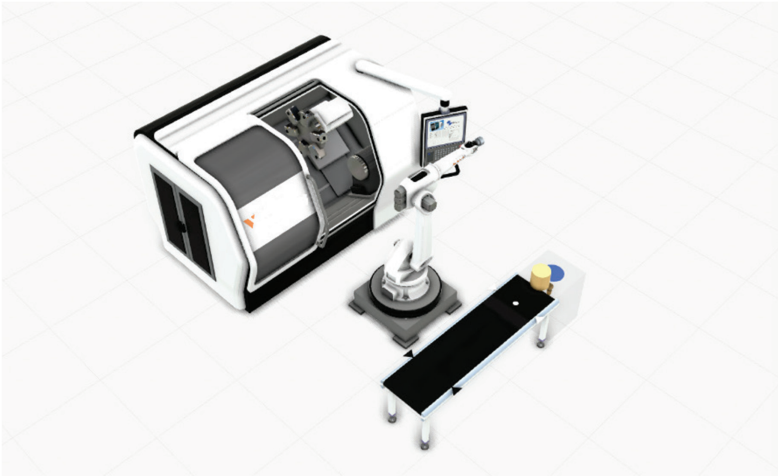


Figure 6 – Capture of the process visualisation process example with conveyor, IR and CNC machine [38]

Overall, without the matter of usage method, the flow for effective DT creation is the same in all virtual replica design tools. Table 2 shows a summary of these in order to make an accurate simulation.

Table 2 – The way to effective DT [34], [35], [37]

Method	Description
Precision / Level of Details (LoD)	DT precision requirements, setting the level of details for specific DT tasks in order to make it realistic and efficient for execution. Both – visualisation and process flow precision.
Data Acquisition and Validation	The import and analysis of data received from monitoring system sensors to enable correct data flow from source equipment should be activated and used in the system.
Data Model	Data Model and format merged by data received from different sources and assets of the system. Both in real-time and historical data.
Synchronisation	Data flow and type should be the same between the virtual and real worlds; the system should be connected via a telemetry middle layer platform.

DT is already a useful tool for the layout design and visual upgrade of a factory before real implementation [39], [40]. However, the author of this dissertation states in Chapters 3.2 and 3.3 of a related study, that DT can also be exploited in another way, enabling full online control over the manufacturing systems for management and re-configuration of it remotely. Moreover, adding VR tools for remote interference online via DT simulation, adds a safety layer and more possibilities for hazardous environments and remote work overall. This is one of the main objectives of the dissertation, enabling precise online synchronisation between DT and its original source in the Industrial Virtual and Augmented Reality Laboratory (IVAR lab). An example of real manufacturing DT in the means of simulations can be seen in Figure7.

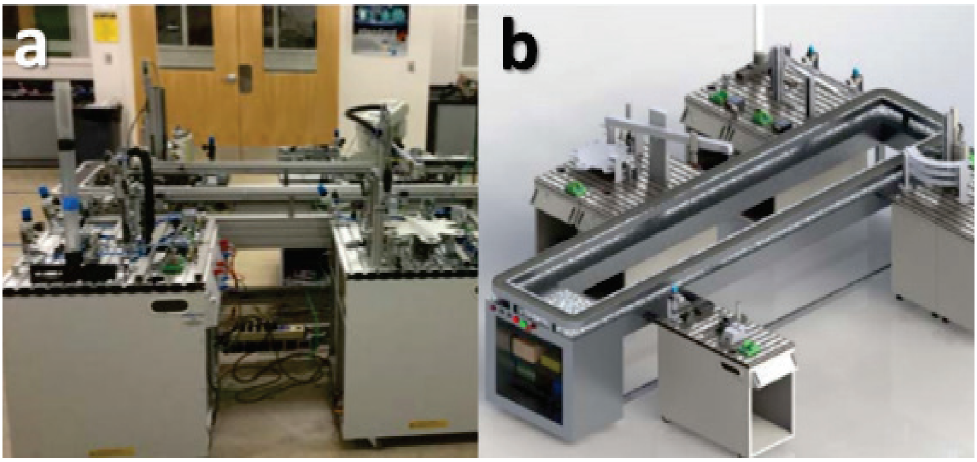


Figure 7 – DT example on Festo FMS system – a) real b) digital [41]

As can be seen, every asset of the real equipment, furniture and all other surrounding objects are able to be transferred to the digital representation in very good quality, which makes research more immersive and precise in the means of simulation. The author of this research is creating a DT toolkit, which also allows for the controlling of a simulated environment.

2.3.2 Virtual Reality

VR is thought to be glasses for entering digital worlds in order to feel a presence somewhere out of reality. However, this statement is misguiding. VR is a digital world which is created with the help of software and hardware. It is not necessary to use a headset to enter. Every computer game that has been done in the past and which will be done in the future is a VR environment. A headset, however, adds presence to the simulation in this world and helps one to feel the surroundings of digital representations in an immersive and more precise way [42].

VR headset technologies have existed since 1965 and were created by Ivan Sutherland [43]. Research has been done by different institutions and authors since, but the main so-called boom and development came at the same time as the i4.0 revolution, as headsets such as Oculus and HTC Vive were developed for a consumer public. However, the expected revenue was not achieved, as the price together with VR computers was too high for the regular consumer. However, times are changing and in 3 years, this market is expected to grow up to 1200% according to Statista studies [14]. Every few months new VR headsets, haptic devices or new experiences in different domains are announced. The primary market for VR is the gaming and entertainment industry, but it is not limited to it.

Listed below are some popular domains in the field of research and start-up industries:

- Entertainment – gaming and VR movies are quite popular nowadays, and the amount of investment and sales in this field are increasing due to the high availability of headsets on the market [42], [44].

- Industrial usage – VR is used by many well-known brands in order to show production simulations and even robot offline programming tools from the inside, simulating presence. For DT simulations, there is more precise investigation, changing the layout and monitoring data flow from the inside as well as robotic arms control via telepresence simulations. Moreover, design in mechanical and civil engineering is one of the developed areas in the field of immersive experiences [45], [46], [47], [48], [49], [50], [51]. The current thesis is focused on using VR as an industrial solution.
- Education, research and training – usage in schools with a narrator behind simpler headsets such as Samsung Gear and Oculus GO with a visualisation of some processes from different domains such as chemistry, mathematics, geology, etc. Moreover, training environments for complicated machines and cases are being exploited in the industrial usage cases described above [52], [53], [54]. Research in various fields is being performed as well, and this dissertation is an example [55].
- Medicine – simulations of anatomy and medical processes. It is used for an introduction to the profession and for some more precise studies in university clinics. However, precision is not enough to rely only on this education tool. Immersive technologies used for rehabilitation have been introduced now in clinics and research centres [56], [57], [58], [59].
- Marketing – companies, universities and countries use VR sets for the promotion of some of their unique values. Moreover, modern tourism is exploiting immersive technologies for popularising places and points of interests [60], [61].
- Military – different haptic devices are mainly being pushed forward as well as Omni platform development for more precise combat situation simulations using different software tools such as VBS3, etc. Usage mainly for training and simulations [62].
- Heritage and art – 3D environment for cultural and historical engagement with the immersive environment as well as unlimited possibilities for the artist in sound design and classical visual art [63], [64], [65], [66].

This is not a final list but only examples of possible usages. The main task is to show that VR is not limited only to entertainment but it is also useful in various fields (see Fig. 8).

Figure 8 – VR usage in various industries [67], [68], [69], [70], [71], [72]

VR headset usage is an important tool for person precision simulation in different environments. Moreover, the main part of immersive experiences are being developed in Gaming Engines such as Unity3D [73] and UnrealEngine [74]. These allow additional functionalities such as a multiplayer mode, which is a multi-user connection to one environment, not limited to a regional location.

The author of this dissertation uses VR as an immersive human presence simulation and collaborative remote control tool of completed simulations. Because VR gives a person a feeling as if you really are in the middle of an actual industrial shop floor with all of the manufacturing equipment, it is the most realistic way to understand real production processes. One can see and check details that are challenging to recognise when desktop visualisation is used. VR can be considered to be an effective presence simulator and immersive co-working tool with the usage of human avatars.

2.3.3 Robot Operating System

Robot Operation System (ROS) is open source software with a primary platform on Linux-based operation systems (ROS for Windows is also available now) for any manipulation with a different type of robots [75], [76]. It consists of a large number of libraries and packages for manipulation with IR and mobile robots as well as with sensors, although it is not limited to those. Moreover, it is integrated with visualisation software packages such as Rviz and Gazebo for more precise simulations of robot environments. ROS can be considered online programming tools which grant control over the robot online.

ROS nowadays is an essential tool for the prototyping of various self-driving algorithms for Unmanned Aerial Vehicles (UAV), Autonomous Ground Vehicles (AGV) and Unmanned Ground Vehicles (UGV) as ROS has a library of various types of sensors, motor encoders and can calculate kinematics by itself. After all, it comes down to the optimisation of connected devices and creating new packages [77], [78]. Moreover, sensor IoT systems ROS can be used as a driver carrier for different types of devices, and as this is open-source software, the number of developers and amount of support is large worldwide. It can be stated that ROS can be used for all robotic and electronic equipment which has a port or connection to the computer based on the Linux operating system.

This is why many developers use Raspberry PI for prototyping, as it already has Linux embedded into the motherboard.

Unfortunately, ROS usage cannot be called a real-time method because of delays in connectivity. However, a huge community and library make these tools universal for experiments on IR, which is a use case for the related dissertation where ROS is used as a test for the safety system simulation on an ABB IR cell example together with simulated sensors.

2.4 Industrial Internet of Things

2.4.1 IoT Concept – Connectivity Layers

The idea of IoT is to connect everything. According to the concept, every device in the system is connected to the network and communicates with others when it is demanded by the program of the device. Devices may have their own network addresses such as an IP and communicate with each other triggering various algorithms. There are different visions of how it should look like, but the main idea is the same [79].

Different platforms have been introduced during recent years as a connectivity layer, and different enterprises offer their enabling solutions: local, remote and wireless. In Table 3, one of the examples of an IoT content table can be seen.

Table 3 – IoT content table based on Al-Fuqaha et al. [80]

IoT Elements		**Samples**
Identification	**Naming**	EPC, uCode
	Addressing	IPv4, IPv6
Sensing		Smart Sensors, Wearable sensing devices, Embedded sensors, Actuators, RFID tag
Communication		RFID, NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-Wave, WiFi, WiFiDirect, , LTE-A
Computation	**Hardware**	SmartThings, Arduino, Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, BeagleBone, Cubieboard, Smart Phones
	Software	OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, Hadoop, etc.)
Service		Identity-related (shipping), Information Aggregation (smart grid), Collaborative-Aware (smart home), Ubiquitous (smart city)
Semantic		RDF, OWL, EXI

Various protocols and systems are exploited in order to adjust cross-connectivity on the factory level, and the main approaches are to move into a semantic subscription agent-based model for better connectivity of the middle layer. According to the author, this dissertation is exploited using mainly an MQTT protocol based on the architecture and achievements done by related research on ontology and semantic models of Smart House and virtual factory solutions [81], [82], [83], [84], [85], [86].

2.4.2 IIoT - Applications

The Industrial Internet of Things is one of the top 10 IoT applications where this domain is together with Smart Home, Wearables, Smart City, Smart Grids, Connected vehicles, Connected Health, Smart Retail, Smart Supply Chain and Smart farming domains [87], [88] as also shown by market share in Figure 9. IIoT is a part of the i4.0 paradigm and is based on fitting IoT to industrial needs based on manufacturing environment requirements related to safety and reliability.

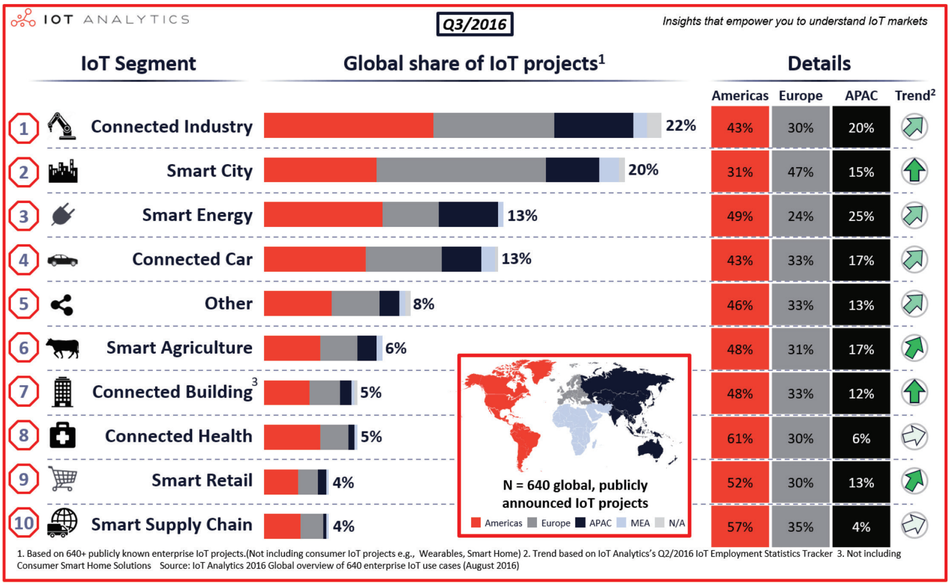


Figure 9 – IoT top fields [88]

However, this dissertation is focused on industrial solutions; the author is willing to show the main applications in this area, which are listed in Table 4.

Table 4 – IIoT platforms applications [89], [90]

Application	Description
Production Line Remote and Local Monitoring / Predictive Maintenance	Additional and embedded sensors usage for equipment in order to monitor and track data on different factors such as power consumption, speed and products produced. Based on this data, an evaluation can be done on how effective working time is and the speed of the machine as well as what can be improved. Moreover, this data is used for evaluation, gathering statistics and comparing it with the maintenance schedule. After a few maintenance procedures, the IoT system analysis tool can predict the time for the next maintenance session.
Unified Factory Telemetry / Smart Ambience, Power	Data stream and flow from different connected assets of the factory are converted into one format in the middle layer. Data traffic depends on connected assets, as for cameras and LIDARs the data is huge. Its management is very crucial in IoT systems. This is where the concept of a BD handle came from. The middle layer can control sequences for power savings and temperature adjustments; for example, making power consumption of the factory more effective, while controlling idle states.
Failure and Safety Control / Remote Maintenance	Based on data received from IoT sensors, the ability to have an overview of failure and safety sides of the factory, making autonomous decisions on the re-launching of equipment or tracking logging remotely. Moreover, digital tools connected via telemetry give one the ability to control the machine remotely, not being on site.
Centralised Control Cloud-based Analysis	Data storage in local or network cloud storage. Access to data worldwide and the ability to compute and analyse data autonomously. All data gathered in one place, making decision making more centralised.
Workforce Tracking / Logistics	Cameras and other area mapping technologies used for logistical routes control of personnel/ mobile robots, as using sensors for working time calculation of the workers in order to analyse behaviour, optimal routes and delays in the manufacturing process.

Related to the dissertation, a crucial part is telemetry and connectivity between virtual and real-world assets of the factory. Exploiting the above described MQTT protocol, the author created an IIoT middle layer architecture and toolkit based on what was introduced in this section: DT, VR and IoT technologies.

3 RESULTS OF RESEARCH

In this section, the main results from the six articles are underlined and evaluated, answering the research questions raised in Chapter 1.

Firstly, articles I and II study the approaches to IR programming. After that, articles III-IV extend the study with the development of adaptive and synchronised DT architecture. Finally, articles V-VI extend the research with the development of a fully synchronised model of DT exploiting IIoT combined with the IR online programming method.

Taking into consideration the related dissertation research problem, which was stated in Chapter 1 – the first two articles (I and II) are an introduction into IR's most common programming methods. Articles III-VI have achieved a safe and effective environment for controlling a robot from immersive DTs.

3.1 Ways of Industrial Robot Optimal Programming

3.1.1 Robotized 3D Scanning

At the beginning of the research, there was a need to find some simple tasks for the robot arm and perform experiments on various applications of robotics. As quality inspection is crucial in novel manufacturing automation processes, the 3D scanning application was chosen for the tests of IR programming approaches in order to analyse the complexity of the process nowadays.

Experiments with 3D scanners were performed in order to find the right methodology for handling the different types of objects depending on size, structure, and complexity for the inspection procedure in the manufacturing processes. IR with an attached 3D laser scanner was used in order to evaluate the test objects (see Figure 10).

Figure 10 – Test object (left) and exemplary single scan result (right) (Article I)

Different shape and type objects were chosen, and a new non-standard routine for IR was elaborated. Two methods for task performance were considered – manual and offline. The offline method demands more time for programming due to the risk of having dark zones in the scanned output model. To inspect the test objects, 3D scanning required precise coverage and manipulation of the robot arm with an inspection tool. Therefore, the manual programming approach was chosen and the results were evaluated according to the time, the complexity of the program (number of steps) and output program duration as seen in Table 5. During the programming, speed and scan, as well as the zone were tuned, depending on the angle and complexity of the test objects' structure.

Table 5 – Evaluation of the IR 3D scanning program (Article I)

Inspected Object	Steps in Programming Code	Time for Programming (in minutes)	Program Duration (in seconds)
Object 1 (Frequency converter)	61	60	1080
Object 2 (3D printed blade)	31	10	120
Object 3 (rubber sphere)	64	60	300

Based on the results and returning from evaluation Table 1 in Chapter 2.2.4 of this study, it can be stated that the manual method is not practical from a time perspective. Every step of the program demanded a visual inspection by the robot operator and tuning from a Joystick is not precise from the coordinates' point of view. Moreover, the testing of the program took a full cycle of scan routine. However, coordinates can be tuned manually from a joystick after or in offline programming too. The offline method is also not suitable for this case, as any valid 3D CAD models were available and a precise routine could not be performed considering all the areas covered.

Robotised scanning research showed that other ways of programming are needed for such kinds of applications, which are adaptive and safer for the operator and can be performed remotely.

3.1.2 Collaborative and Safe Work Environment Using Robot Operation System

ROS is used [91] for various experiments on IRs. As the offline and manual methods were not sufficient for the application described in Article I, different ways of enhancing the methods of operator collaboration and control with IR were considered. The online programming tool ROS was used, which was previously described in Chapter 2.3.3, to create a safe environment simulation for an operator with the ability for a robot to adapt to its surroundings using additional perception sensors. In this experiment, the 2D LIDAR was used.

Before the experiment could be executed, methodology and difference between offline and online programming should be compared. As shown in Figure 11, the offline method is a straightforward execution based on information from the controller, which gets data from additional perception sensors and performs the task. When something interferes with it, the robot does not adjust itself, but stops so the code cannot be re-written while executing the pre-defined routine task.

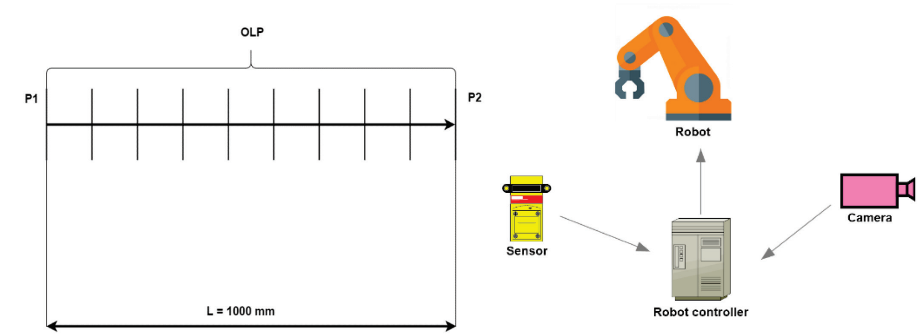


Figure 11 – Visual offline programming method introduction (Article II)

While the offline programming task has a pre-defined trajectory, an online tool can involve more milestones during the route of the robot's path. As seen in Figure 12, deviations are possible in every step of the program, which can have its own security level. In details, when sonar or an LIDAR sensor detects an operator or object in the path of the robotic arm, an online tool such as ROS tries to change trajectory to avoid it and if this is not possible, stops the arm while the object or person has not left the danger zone.

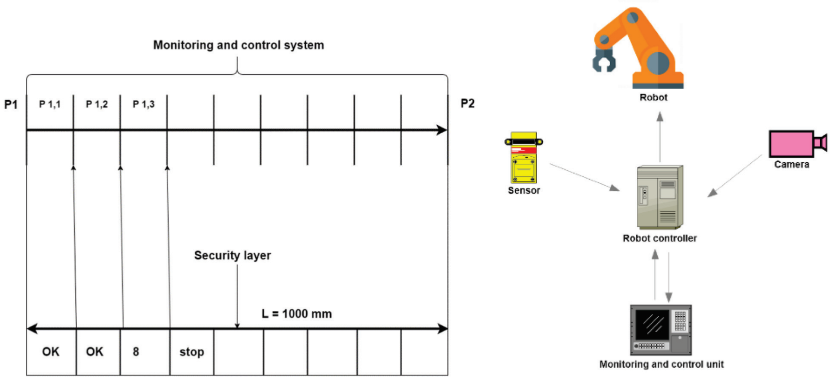


Figure 12 – Proposed online programming method (Article II)

An experiment in a visual ROS environment was designed, where a virtual robot was connected with a real sensor. The solution was validated for research purposes; however, it is not ready for industrial purposes due to a few reasons:

- The unfriendly interface of the ROS – not a valid solution for end customers
- The learning curve of the ROS
- Delay in communication making communication more difficult

Based on the experiments performed as part of this research in Articles I and II with a standard manual and online programming tools, it was stated that there is a need for design tools which allow for an accurate synchronisation and adaption of industrial equipment to the surroundings and which are modular and user-friendly for the users. This sub-chapter answered RQ1, and it was decided to use the online method as a base for further developments.

3.2 Effects of Development of the Adaptive Control Concept

3.2.1 DT Connectivity Architecture Development

In Chapter 2.3.1, the concept and methods of creating an effective DT were described. However, the integration of Factory Telemetry (FT) to the connectivity model of the DT and its real representation, demands one to enable factors and the architecture of the connectivity model, enabling dual-way communication with each other. To remotely control and observe the real equipment of factories through their digital replicas, the data flow of specific control and data gathering commands should be modelled as a visual flow and schematic architecture.

As seen in Figure 13, all data from both the virtual and real-world sensors transfer all data to the system controller, which distributes the necessary data to the virtual and real representations of the system. At the same time, essential data such as kinematics, coordinates and state are transferred from the real machines to the same controller. Thus, the virtual system imports and makes visual equipment work. However, if the virtual robot is interfered with by the operator from a desktop application or a VR avatar, the data, in the same way, will be sent to the middle-layer integrating data imported by real machines. This closed-loop architecture enables a synchronisation of the two representations of the same factories.

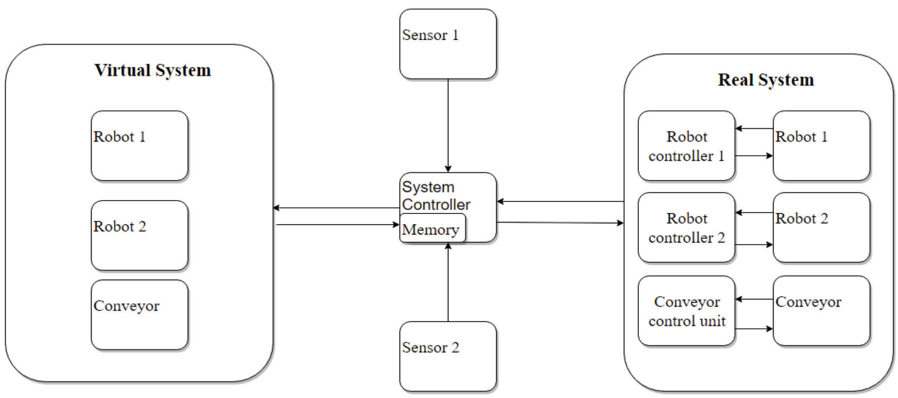


Figure 13 – Virtual and real-world synchronisation model (Article III)

There are two ways to enable and handle data synchronisation between two factory representations:

1. Real-time – a binary protocol is used for data transfer.
2. Web service oriented – the ability to store data and use it for historical tests and visualisation as well.

Both methods are described more thoroughly in Article III. However, the hybrid model of both solutions is the method of approach for the online monitoring and control of DT. Middle-layer creation with an accessible Application Programming Interface (API) is crucial to enable temporal time-stamped data storage and connectivity to ports for assets in order for them to use it to analyse issues that happened in the past and system performance overall.

The main requirements for API development are as follows:

- Plug-and-play connectivity
- Data storage capability

- Possibility to enable data analysis with external tools
- Connectability of agents such as manufacturing equipment, virtual equipment and various sensors in order to receive and send data.

Figure 14 shows the data connection model between real and virtual representations of the system via a middle-layer platform, both web-based and in real-time:

- Data flow 1 from and to both virtual and real systems in real-time.
- Real system data storing 2 in the historical data warehouse of the middle layer can be run via data transfer channel 3 as a simulation in the Virtual system and used as a template for re-programming the real one.

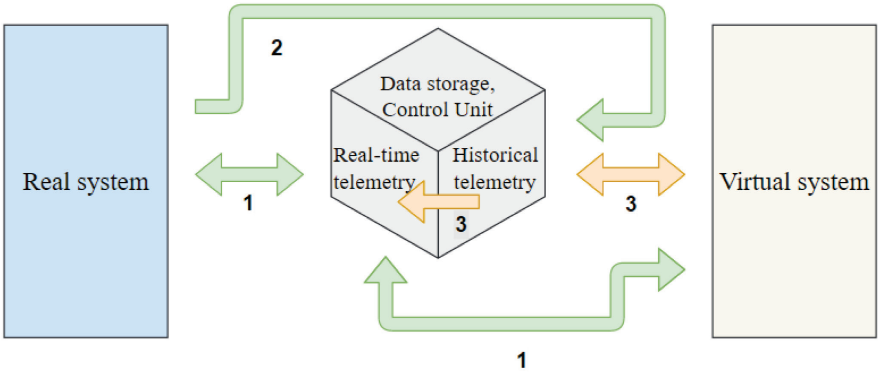


Figure 14 – Data flows between the virtual and real systems (Article III)

As stated before, the middle layer of the DT synchronised model should use two different aspects for telemetry data exchange and communication – a TCP protocol for real-time, fast traffic exchange and a Web-service based protocol for data storage options. It is worth mentioning that there are no fully real-time solutions due to latency, which is always higher than 0, so these solutions can be called online connection protocols. Different options such as a cloud-based IoT solution should be considered as well as new communication methods for a fast, secure and reliable data exchange.

This approach partly answers RQ3 raised in this dissertation on connectivity architecture for DT synchronisation with its real representation in a dual-way. Based on this architecture, experiments were performed to find a method for effective and precise data exchange between virtual and real DT middle layer assets to enable and create a toolkit for research and industrial purposes.

3.2.2 Robots Adaptive Process Control Methodology

The research problem of the related dissertation was initiated by robot programming methods evaluation for safety, collaborative and adaptivity factors. Coming back to the domain of the online programming method as a flexible evaluation of how-to achieve a level of intelligence for IR using existing tools and developed in Article III's DT connectivity architecture.

Three main components for the concept of intelligent IRs are the following:

1. Object recognition – IR should understand the shape, colour, dimensions and structure of the object, depending on the task.
2. Object localisation – coordinates in the Cartesian coordinate system and a geometrical depth of the object by the aid of mapping and stereo sensors such as LIDAR and cameras in this experiment example.

3. Path planning – according to the data received from the sensor, robot arm kinematic movement adjustment algorithm activation is done according to the needs and executable task related to the test object.

Additional steps are demanded in order to execute the input task for this exact scenario, where the robot should sort objects according to their structure and colour when attached to a gripper (Article IV). These additional and critical steps are visualised in Figure 15 and are as follows:

1. Motion execution – according to the track generated by path planning algorithms.
2. Object placement – the result of an input task after which the robot moves to the standby position, or executes the next task generated by input sensors or the operator.

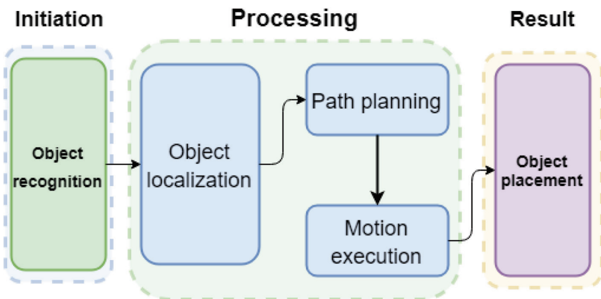


Figure 15 – Steps for solving the smart path planning of a robot (Article IV)

The setup of this simplified implementation of the smart robot planning method for palletising tasks is tested in the DT of the real robot cell before implementation in real life to minimise robot usage for experiments so it can execute other tasks not losing time for re-programming and setup changes. An exemplary setup is visualised in Figure 16 and consists of a six-axis IR, two LIDARs (L1, L2 in the figure) and a single stereo camera (Cam in the figure).

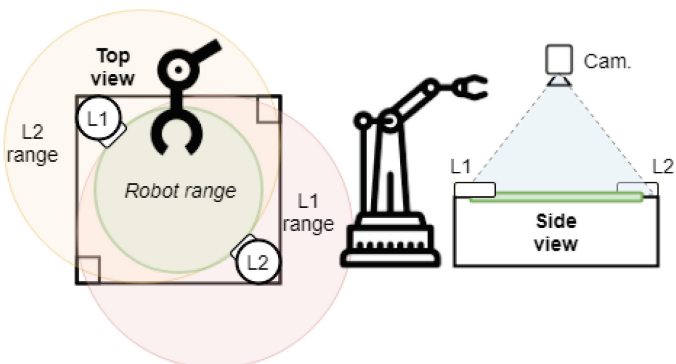


Figure 16 – Sorting robot setup with additional perception tools (Article IV)

Additionally, the software was chosen because of it DT visualisation with the possibility of a VR presence in ROS is described above in Chapter 2.2.3 and Matlab toolboxes. ROS open source libraries such as sensor and stereo camera calibration toolkits as well as localisation and mapping toolkits are connected to the Matlab and by this means, gathers input data from the external sensors such as LIDARs and camera, VR, UI and IR. The Matlab toolbox for mathematical inverse kinematics calculation, signal processing and object identification task is used. The architecture of the system is visualised in Figure 17 and described in detail in Article IV.

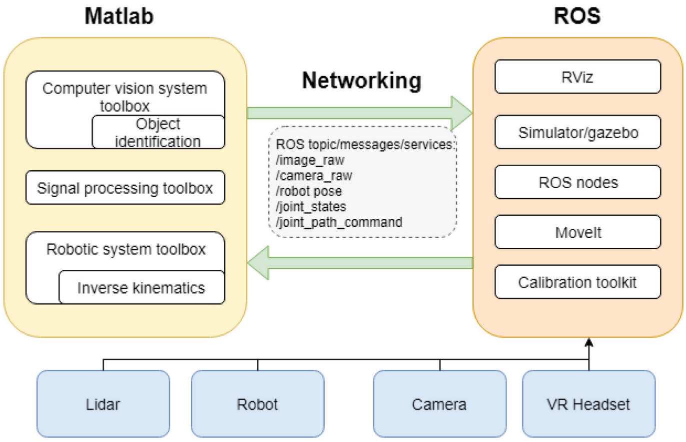


Figure 17 – ROS system setup (Article IV)

However, for the exact experiment a Matlab toolbox was used – it is too slow for real-time and online applications. In the future experiments described in the next chapters, a kinematics module of gaming engine Unity3D is used. Even for Matlab, after the experimental architecture was excluded for the future; its architecture flow is used the same way; only the software tools have been changed.

A related system proposal, with the help of DT visualisation connected to a real machine, intended to help with VR usage for motion execution and to give the ability for unlimited use and test cases for the researcher, without interfering with an actual IR cell, was made. However, enabling re-programming and self-learning algorithms for future usage on real machines, as all data is transferred through one middle-layer, was introduced in Chapter 3.2.1. This chapter fully answers RQ2 and RQ3 of a related study on the creation of a safe, reliable and open for an experiment design tool for human-robot collaboration.

3.3 Effects on IIoT Connectivity Framework Development

3.3.1 Dual-way Synchronisation between a Real Robot and Its Virtual Counterpart

The simulation environment for the experiments based on the architecture and concept developed as described in Chapter 3.2 was created in a game engine platform because it is free to use for research and it has a large functionalities list, regarding physics and VR tools. The aim of the simulation environment, or DT of this experiment, is to be able to, by exploiting FT architecture, synchronise real and virtual worlds using the example of an IR.

For the re-creation of a real IR, a cell was created with the aid of 3D modelling software, producing a virtual representation of the laboratory with all its assets (see Figure 18). The manual of creation of a precise digital model is described in Article V in all aspects of the technical side. It is essential to mention, that process is time-consuming and complex, because of the aim to create a precise replica of real existing objects.

However, if first time creation of environment can take months, next upgrades and modifications are quick, due to the modular architecture of the environment. A short list of most important aspects what should be considered while digitalising manufacturing equipment, when both are created using their own 3D models based on drawings or importing from manufacturing websites or other libraries follows:

- The rigging of the model should be performed, which means defining the central pivot points for movable objects (see Figure 19). Usual 3D models from IR vendors are with an incorrect central point.
- Game engine rendering is performed in real time – the models' geometry should be simplified for smoother visualisation and performance of the simulation, the however main body should be unmodified to keep scale and collision mesh correct.
- Proportions should be re-checked before importing to the simulation environment as it is 1 to 1 scale DT.
- Kinematic model adjustment for the equipment. One robot example – determination of the endpoints for joint kinematics configurations for calculations of axis movements (see Figure 20), for precise and intuitive path planning control of the joints.

Without consideration of these crucial points, the precision of a simulation will be lost and efficiency will lack the correct data.

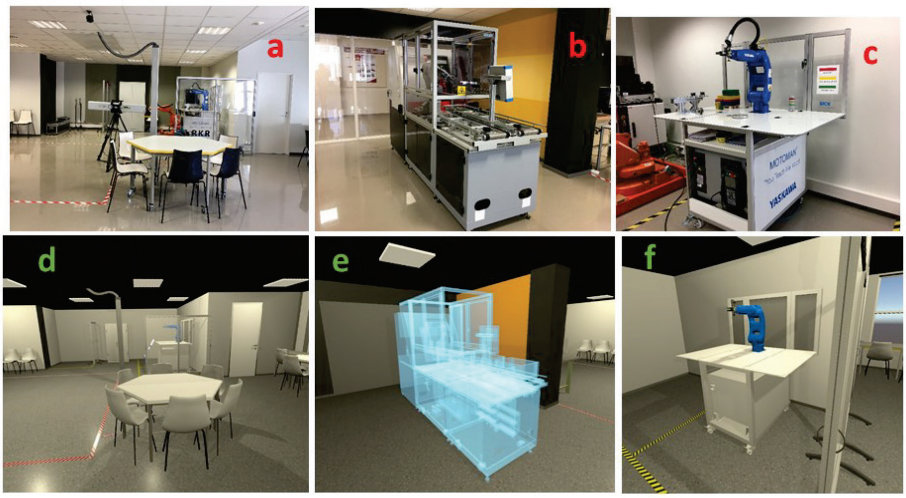


Figure 18 – Digitalized laboratory – IVAR DT (a, b, c – real; d, e, f – replica) [92]

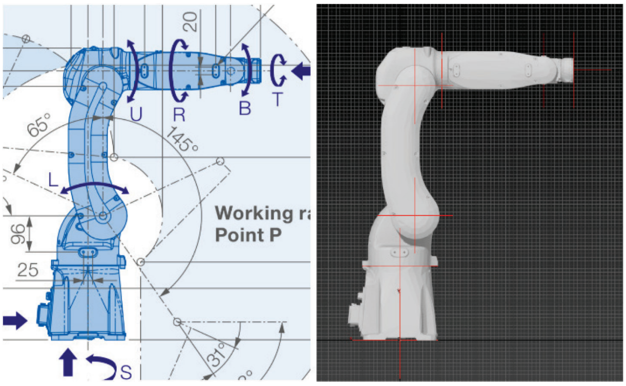


Figure 19– Robot model rigging (Article V)

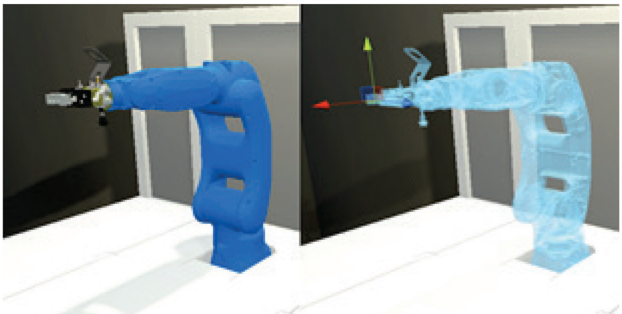


Figure 20 – DT robot (left) and its digital programming clone with IK endpoint target (right) (Article V)

To create an immersive experience in DT, the VR toolkit of a game engine was used in order to manipulate a virtual robot from a safe and remote place. A prototype of UI was created in order to be able to control each joint of the robot with a precision of 0.01 degree and with the possibility of bringing the robot back to its initial position (see Figure 21). It is worth mentioning that each joint has precisely the same maximum joint angle limit as its real source. Commands from virtual UI send commands to the real IR controller, which executes the task based on the input. However, if the operator in a real factory environment starts a manual programming of the robot, it will override the virtual code for safety reasons, and its virtual replica will follow those commands, visualising the movements of the IR. Moreover, the safety aspect is considered as the real IR has a safety system, which on detection of a person or object near the machine, as described in Chapter 3.1.2, stops the robot and cuts off the connection between the two worlds until the object or person is removed from the danger zone.

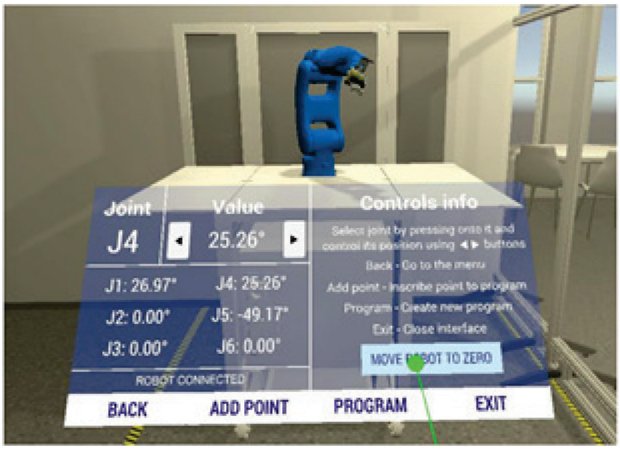


Figure 21 – Digital Twin control UI (Article V)

A crucial aspect of the creation of a DT environment is the connectivity layer between the virtual and real worlds. As described in Chapter 3.2.1, middle layers should save and synchronise data between the two world assets. After the creation of the experimental environment, experiments on the architecture of real life tests were performed based on an introduced DT scheme, which can be seen in Figure 22.

The main idea is to synchronise data in the middle layer, which is labelled on the schematic as DigitalTwinController. Its location in this experiment is locally based in the computer, which ran a simulation application. Input from the user in a virtual environment is sent directly to the middle layer, where data of the joint movement task of the routine which should be performed is exported to the digital IR controller labelled as VirtualRobotController and IR controller, which in this case is MotomanGP8Controller.

Next, for both the virtual and real robot controllers, the activities are similar:

- Virtual IR controller calculates endpoint or rotates joints depending on the input task and visualises it after calculating this in a virtual environment.
- Real IR controller sends commands to the IR via a Local Network; the Ethernet in this case.
- Feedback about the execution and success/failure of the given command is sent by both robots. With real – the status data of motors in joints. For virtual, collision detection data and coordinates data in a simulation environment. The collision system is described in more detail below.

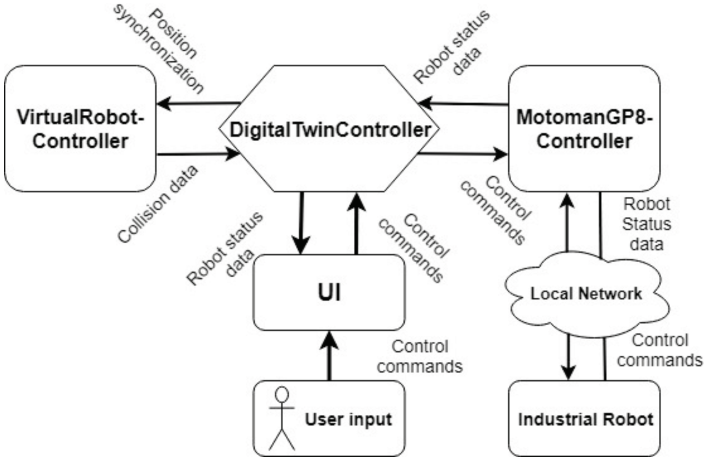


Figure 22 – Digital Twin system scheme (Article V)

The experiment was performed successfully as proof of the concept of the proposed connectivity architecture between a real factory and a virtual representation presented in this dissertation.

The environment created is a basis for unlimited research options with manufacturing equipment in the virtual world. The re-programming of real machines and safe teleoperation of robots in dangerous environments using VR tools are only a few of the advantages of the system. Controlling a robot with virtual sensors was achieved – the transparent safety wall/layer in the digital environment was created around physical objects. When an operator who is manually controlling a robot reaches a plane that is too close, the virtual layer will collide with the robot and stop it from making the next dangerous movements, creating an additional safety layer for real-world machines. With it, workspace awareness was improved by using the developed DT design tool.

RQ4 was responded to by the development of a fully synchronised robot cell DT with a working middle layer of telemetry introduced in Article III.

3.3.2 Digital Twin integration with IIoT

The creation of a fully synchronised DT with manufacturing environment is a feasible tool for monitoring and a fast re-programming of a machine while being away from the shop floor. However, what is most important is the implementation of the connectivity protocol to the middle-layer, as the telemetry data transfer speed should be as real-time as possible to be a more precise control tool.

The IIoT protocol combination with a created synchronised DT model should increase the dynamic and make the created environment more flexible and modular for new connectivity assets. MQTT was chosen to be the base prototype of connectivity because it is an out-of–the-box solution. However, the game engine demanded additional programming and import for IIoT related tools for the experiment as is more precisely described in Article VI. Cloud-based storage was installed to the server in Germany in order to have a longer lag between assets connected – this replaced the local DigitalTwinController server described in Chapter 3.3.1. The event-based and subscription-agent model was exploited as a new part of the middle layer, which means that every piece of equipment or sensor is an agent and sends data to cloud storage.

To this data an interested asset can be subscribed, which means in case of the experiment done in this dissertation, that virtual robot is being subscribed for updates from real and real from virtual. This approach simplifies adding new equipment and gives one the ability to send control commands from various platforms – VR, desktop and mobile. During this research, robot control from a desktop application from Italy was tested.

The synchronisation process through the IIoT protocol middleware is visualised in Figure 23, where the RobotController controls the machine or robot; in this case, through a specific API. The DigitalTwinController manages a DT model and agent-subscribe middleware pushes data to cloud storage. Communicators handle data from the real and virtual components in order to filter out only the necessary data format values.

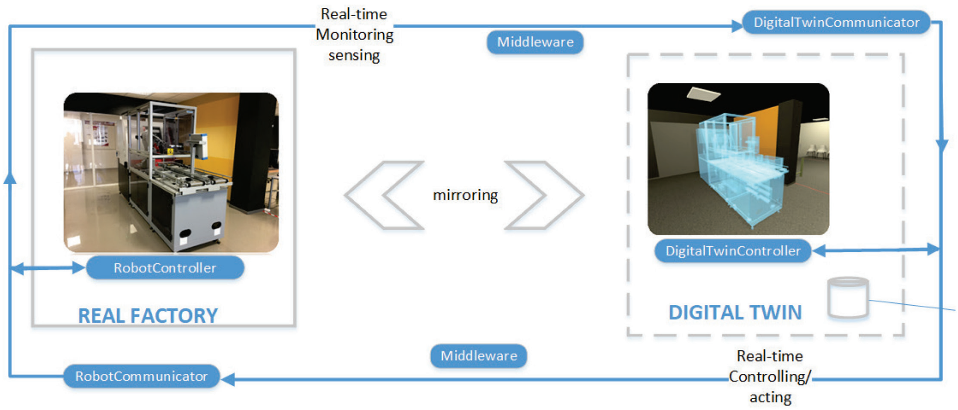


Figure 23 – Synchronization process through the proposed software infrastructure (Article VI)

The support of VR tools designed by a DT toolkit was chosen in order to have a multi-platform environment and game engine, because of the integrated physics engine. Moreover, game engines have embedded collaborative modes such as a multiplayer mode, where different people around the world can collaborate in one environment allowing researchers or operators to work on one problem as well as discuss research topics related to the specific machines (see Figure 24).

This feature is crucial in the concept of smart factories as remote maintenance and educational processes can be performed in a DT environment, but with interaction with synchronised ones. Moreover, it allows for the monitoring and performing of a complex task from remote locations.

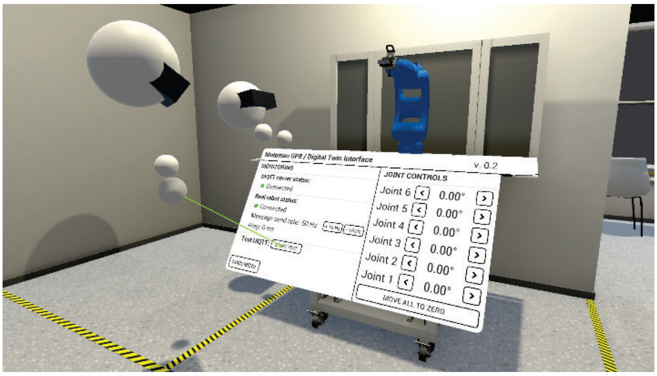


Figure 24 – Researchers' avatars controlling the robot in VR (Article VI)

Coming back to the traffic speed, optimisation was tested by using different use cases adjusting the rate values by 10 to 100 Hz with different frame rates. The Quality of Server (QoS) of the MQTT protocol, which determines how reliably messages are delivered, was also considered and adjusted from 0 to 2, where higher values require more computation time for the processing of sent or received messages. The experiment was performed using two different locations via a server in Germany:

1. Physical robot control in Estonia from the DT in the same location with the robot showed that QoS 1 and 2 are more optimal and the lowest ping was near 35 Hz when it was 75 ms (See Figure 25).

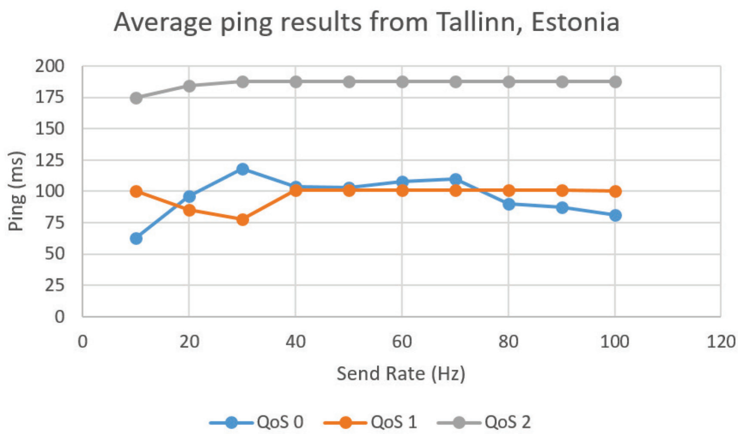


Figure 25 - Experiment results Tallinn, Estonia (Article VI)

2. Physical robot control in Estonia from the DT in Bari, Italy showed the same results regarding QoS level and the rate showing 100 ms as the average speed was 20-25 Hz (See Figure 26).

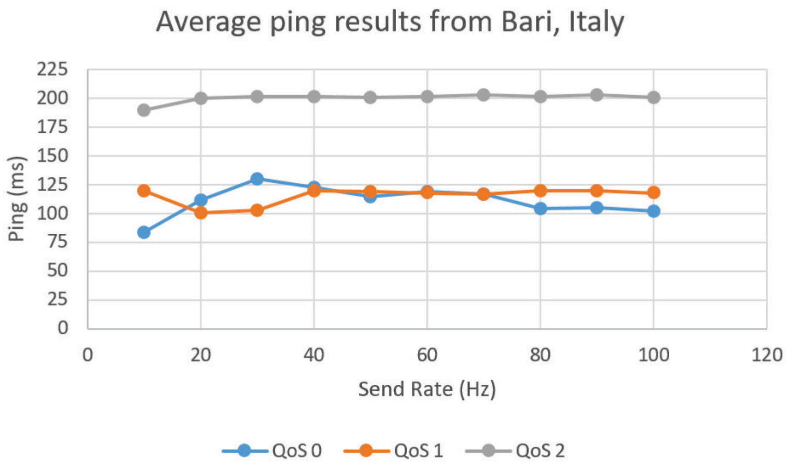


Figure 26. Experiment results Bari, Italy (Article VI)

As a result, it can be stated that the send rate did not affect the time of DT synchronisation as much as the QoS parameter did due to a longer message processing speed. However, the communication speed was not high enough to rely on as real-time; it is only a matter of optimisation and experiments with different protocols and ways of networking in order to increase communication speed between real and virtual worlds.

With this experiment, answering RQ4 was finalised and as a result, a fully synchronised with a real manufacturing environment DT was developed. Moreover, its modular UI and environment enable the adding of various environments and equipment as well as connecting those through IIoT protocols allowing for the designing of a precise and immersive simulation and virtual control systems. The impact of this research for academics and enterprise is described in detail in Chapter 4 of this study.

4 DISCUSSION

This chapter concludes the main outcomes provided in Articles I-IV in two categories:

1. The dissertation and experiments' effect and value for research investigations and theoretical developments which can be conducted in laboratories.
2. The dissertation and experiments' effect and value for manufacturing companies and the practical usage of the results achieved for the everyday work process.

4.1 Theoretical Implications

The related dissertation has a few theoretical implications. These are mainly related to methodologies developed for the conduct of further research in the field and in laboratories. The primary research outcome approaches and theoretical investigations are described in a way so as to be repeatable and extendable by other investigators in the field.

Articles I and II are an introduction and as an outcome, show and compare the manner of IR programming methodologies. Based on Article I, the methodology for using a laser scanner integrated with a robotic manipulator was introduced and the optimal way for programming of the chosen type of objects in a more precise way was also introduced. However, it was not found to be an effective use of time or a safe way for IR programming. Article II's studies investigated collaborative, safe working environments which can react to surrounding movements and the re-allocation of objects. These studies can be used by researchers for an inspection methodology development base and the development of ROS-based collaborative environment simulations.

Articles III and IV study architecture creation methodology for DT connectivity with its real source equipment and ways of creating an adaptive, self-learned IR with the usage of VR and AR. The architecture of creation of fully synchronised DT was introduced in Article III, which can be used as a base for future research and the addition of modules to every component, while the overall architecture is universal for a different type of manufacturing or lab equipment. Moreover, Article IV introduces ways and the design of an experiment on adaptive IR using the means of localisation and path-planning for a sorting task. A concept of learning via the digital environment was introduced, exploiting the architecture of factory telemetry introduced in Article III.

The main theoretical impact is done by Articles V and VI, where the main experimental methodology introduced in Articles I-IV was conducted, and a fully synchronised DT developed on the example of an IR was created. The theoretical basis is in the modular framework created and open source code, which can be used as a basis for various kinds of experiments with factory telemetry, collaborative experiments on safety systems and equipment performance evaluation for the purpose of monitoring and optimisation. Article VI also proves modularity by exchanging the local telemetry middle layer by MQTT protocol and adding multiplayer options to the source environment for online collaboration. The precision of the system is defined by researchers giving the possibility of adjusting every model or process of the immersive simulation; for example, adding gears to robot joints in order to investigate related to the work and adjust them to find more optimal solutions. Moreover, the universal basis of UI reduces time spent on re-programming and the integration of research experiments into Smart Industry environment lab setups.

4.2 Practical Implications

In addition to the value for researchers, the developed methodology and environment in the related dissertation also have beneficial approaches for manufacturing environments.

The main value of Articles I-II is in the evaluation of different programming and inspection hardware setup evaluation comparisons. Based on this, its safer IR cell can be integrated and a correct quality control method camera can be chosen. Articles III and IV's main value lies in its system integrators to be a schematic for developed solutions.

The central practical value of the related research is in Articles V and VI, in the modular and immersive environment created. Articles V-VI's practical outcomes are mainly as follows:

- Remote teleoperation of IR online, which eliminates the human presence factor in hazardous and dangerous environments as well as avoiding contact with manufacturing machines.
- Support for collaboration online between various "client" environments. This gives the possibility of collaborating worldwide by means of a multiplayer mode, allowing the teaching, educating and maintaining of the machines — there is no need to bring a manufacturer specialist on site for maintenance or the re-configuration method. DT also supports training in the simulation on real machines, while those machines perform their tasks.
- Additional safety layers. Simulation triggered sensors can stop the real machine making it infeasible to install physical sensors to prevent wall/table hitting. A collision detection mechanism embedded into game engine physics can prevent dangerous interference with the surroundings.
- Data visualisation and monitoring from the VR environment give the possibility of analysing it while not being onsite. In addition, historical data can be used for statistical improvements and evaluation of failures in the past, while the real machines are doing their duties.
- DT gives a possibility for the re-programme machine not being disconnected from the process. While the DT connection is cut off, new changes to the line can be transferred and integrated into the new robot routine loop after the connection is established again.

It is important to mention that these advantages have not been finalised as every aspect of the manufacturing process can be simulated on a realistic model. It is then embedded on real machines' work process with quick re-programming through the IIoT platform. New equipment is also easy to add by means of the plug and play connectivity methods.

5 CONCLUSIONS AND FURTHER RESEARCH

The primary goal of this dissertation, which was aimed at the development of a DT methodology base on the IR cell use case, was accomplished with an experimental environment output available for re-creation for other or collaborative researchers.

To separate the conclusion points discussed and summed up separately in Chapters 3 and 4 more precisely on the theoretical and practical impact levels – the following achievements are outlined as the dissertation's main contributions:

- Various IRs' programming methods were studied, tested and experimented on within the shop floor of the laboratory or in accurate simulations connected with real cells in order to evaluate safer and more collaborative ways to access heavy industrial machines for people.
- Work on factory telemetry architecture creation was accomplished considering the modern ways of connectivity models in manufacturing and simulation data input.
- Proof of concept on factory telemetry connectivity between the manufacturing floor and its DT or IR cell example was performed. Thus, a description and methodology of design methods for fully synchronised DT were created.

The result of this dissertation consists of the game engine used for the design of manufacturing layout and implementation of an IIoT middle layer which can support a full-way synchronisation between real machinery and its virtual counterpart. The environment support modularity created for new equipment and multi-platform usage such as a VR headset, desktop and mobile applications for monitoring real equipment from DT as well as interfering with it. Moreover, the multiplayer option as a collaboration tool was used and tested between Italian and Estonian laboratories successfully.

Future Work

Future developments of the DT methodology created should be enlarged with the following functionality but not limited to it:

- Equipment library for the final customer – can also be exported from existing Simulation software tools.
- UI adapted not only for robotic equipment.
- Offline programming method integration to the design environment and research on hybrid programming methods combining offline and online interference methodologies in fully synchronised DT.
- AI and machine learning algorithms used for the education of manufacturing hardware – for example, on sorting a robotic task, educating a robot on sorting virtual objects, then transferring the virtual knowledge gained to the real machine.
- Future developments have to address the optimisation of connectivity systems with a purpose to increase traffic transfer speeds. As part of it, novel telemetry methods as 5G should be tested.
- Cybersecurity experiments on traffic transfers.
- The precision of details inside the equipment models for the troubleshooting of broken parts or experiments on optimisation.

References

[1] European Comission, “Digital Transformation Monitor,” [Online]. Available: https://ec.europa.eu/growth/tools-databases/dem/monitor/. [Accessed 5 2 2019].

[2] European Comission, "Europe's Digital Progress Report," 2017.

[3] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger and O. Madsen, “Robot skills for manufacturing: From concept to industrial deployment,” *Robotics and Computer-Integrated Manufacturing ,* pp. 282-291, 2016.

[4] A. Gwiazda, W. Banas, A. Sekala, K. Foit, P. Hryniewicz and G. Kost, “Modular industrial robots as the tool of process automation in robotized manufacturing cells,” in *IOP Conference Series: Materials Science and Engineering*, 2015.

[5] V. Villani, F. Pini, F. Leali, C. Secchi and C. Fantuzzi, “Survey on Human-Robot Interaction for Robot Programming in Industrial Applications,” in *IFAC PapersOnLine*, 2018.

[6] IFR, “Robot density rises globally,” International Federation of Robotics, 2018.

[7] T. Kangru, J. Riives, T. Otto, M. Pohlak and K. Mahmood, “Intelligent Decision Making Approach for Performance Evaluation of a robot-based Manufacturing Cell,” in *Proceedings of the ASME 2018 International Mechanical Engineering Confress and Exposition: IMECE2018* , Pittsburgh, 2018.

[8] ISO, *ISO/TS 15066:2016,* International Organization for Standardization, 2016.

[9] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier and P. Fraisse, “Collaborative manufacturing with physical human–robot interaction,” *Robotics and Computer-Integrated Manufacturing ,* 2016.

[10] ISO, *ISO/TR 20218-1:2018,* International Organization for Standardization, 2018.

[11] A.T.Kearney, “The Internet of Things: A new Path to European Prosperity,” A.T.Kearney, 2016.

[12] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin and D. Aharon, “Unlocking the potential of the Internet of Things,” McKinsey Digital, 2015.

[13] S. Boschert and R. Rosen, “Digital Twin—The Simulation Aspect.,” in *In Hehenberger P., Bradley D. (eds) Mechatronic Futures.*, 2016.

[14] Statista, “Projected size of the augmented and virtual reality market 2016-2022,” Statista, 2019a.

[15] Statista, “Projected breakdown of global augmented and mixed reality market 2022,” Statista, 2019b.

[16] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld and M. Hoffmann, “Industrie 4.0,” *Business & InformationSystems Engineering,* pp. 239-242, 2014.

[17] i-SCOOP, “Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0,” 6 January 2019. [Online]. Available: https://www.i-scoop.eu/industry-4-0/#Industry_40_is_not_just_about_factories_anymore.

[18] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel and M. Harnisch, “Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries,” The Boston Consulting Group, 2015.

[19] P. Zheng, H. Wang, Z. Sang, R. Y. Zhong, C. Liu, K. Mubarok, S. Yu and X. Xu, “Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Frontiers of Mechanical Engineering,* pp. 137-150, 2018.

[20] S. Weyer, M. Schmitt, M. Ohmer and D. Gorecky, “Towards Industry 4.0 - Standardization as the crucial challenge for highly modular, multi-vendor production systems,” in *15th IFAC Symposium on Information Control Problems inManufacturing: INCOM 2015*, 2015.

[21] S. Wang, J. Wan, D. Zhang, D. Li and C. Zhang, “Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination.,” *Computer Networks,* pp. 158-168, 2016.

[22] Y. Lu, “Industry 4.0: A survey on technologies, applications and open research issues,” *Journal of Industrial Information Integration,* pp. 1-10, 2017.

[23] E. Manou, G.-C. Vosniakos and E. Matsas, “Understanding industrial robot programming by aid of a virtual reality environment,” *International Journal of Mechanical Engineering Education,* pp. 1-21, 2018.

[24] K. Kaltsoukalas, S. Makris and G. Chryssolouris, “On generating the motion of industrial robot manipulators,” *Robotics and Computer-Integrated Manufacturing,* pp. 65-71, 2015.

[25] D. J. Fölscher and K. Kruger, “Saving Time on Robot Programming: Programming by Demonstration using Stereoscopic Motion Capturing,” in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, Stellenbosch, South Africa, 2016.

[26] D. Massa, M. Callegari and C. Cristalli, “Manual guidance for industrial robot programming,” *Industrial Robot: An International Journal,* pp. 457-465, 2015.

[27] J. Zhang, Y. Wang and R. Xiong, “Industrial Robot Programming by Demonstration,” in *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2016.

[28] Z. Pan, J. Polden, N. Larkin, S. Van Duin and J. Norrish, “Recent Progress on Programming Methods for Industrial Robots,” in *Robotics and Computer-Integrated Manufacturing*, 2012.

[29] E. Dean-Leon, K. Ramirez-Amaro, F. Bergner, I. Dianov, P. Lanillos and G. Cheng, “Robotic Technologies for Fast Deployment of Industrial Robot System,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, 2016.

[30] E. M. Orendt and D. Henrich, “An Architecture for Intuitive Programming and Robust Execution of Industrial Robot Programs,” in *In: Schüppstuhl T., Tracht K., Franke J. (eds) Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter.*, Berlin, Heidelberg , 2018.

[31] A. Mussomeli, B. Meeker, S. Shepley and D. Schatsky, “Signals for Strategists: Expecting digital twins,” Deloitte Insights, 2018.

[32] Fraunhofer, “Hannover Messe Preview 2017 - Industrie 4.0: Virtual twin controls production,” Fraunhofer-Gesellschaf, 2017.

[33] A. Parrott and L. Warshaw, “Industry 4.0 and the digital twin: Manufacturing meets its match,” Deloitte.

[34] T. H.-J. Uhlemann, C. Lehmann and R. Steinhilper, “The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0,” in *The 24th CIRP Conference on Life Cycle Engineering*, 2017.

[35] M. W. Grieves, “Digital Twin: Manufacturing Excellence through Virtual Factory Replication,” 2014.

[36] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” *The International Journal of Advanced Manufacturing Technology,* pp. 3563-3576, 2018.

[37] W. Terkaj, T. Tolio and M. Urgo, "A virtual factory approach for in situ simulation to support production and maintenance planning," *CIRP Annals - Manufacturing Technology,* pp. 451-454, 2015.

[38] K. Mahmood, T. Karaulova, T. Otto and E. Shevtsenko, “Performance Analysis of a Flexible Manufacturing System (FMS),” in *The 50th CIRP Conference on Manufacturing Systems* , 2017.

[39] R. Rosen, G. v. Wichert, G. Lo and K. D. Bettenhausen, “About The Importance of Autonomy and Digital Twins for the Future of Manufacturing,” *IFAC-PapersOnLine,* pp. 567-572, 2015.

[40] F. Tao and M. Zhang, “About The Importance of Autonomy and Digital Twins for the Future of Manufacturing,” *IEEE Access - SPECIAL SECTION ON KEY TECHNOLOGIES FOR SMART FACTORY OF INDUSTRY 4.0,* pp. 20418-20427, 2017.

[41] D. Ngo, D. A. Guerra-Zabiaga, G. González-Badillo and R. Vatankhah Barenji, “Towards a Digital Twin for Cloud Manufacturing: Case Study,” in *ASME International Mechanical Engineering Congress and Exposition*, Pittsburgh, 2018.

[42] C. Anthes, R. J. García-Hernández, M. Wiedemann and D. Kranzlmüller, "State of the art of virtual reality technology," in *2016 IEEE Aerospace Conference*, 2016.

[43] C. D. Watkins and S. R. Marenka, “CHAPTER 2 - FROM WHENCE VIRTUAL REALITY—A BRIEF AND INCOMPLETE HISTORY,” in *Virtual Reality Excursions with Programs in C*, ACADEMIC PRESS, 1994, pp. 43-48.

[44] J. Bates, “Virtual Reality, Art, and Entertainment,” *Presence: Teleoperators and Virtual Environments,* pp. 133-138, 1992.

[45] L. P. Berg and J. M. Vance, “Industry use of virtual reality in product design and manufacturing: a survey,” *Virtual Reality,* pp. 1-17, 2017.

[46] J. I. Lipton, A. J. Fay and D. Rus, “Baxter’s Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturin,” *IEEE ROBOTICS AND AUTOMATION LETTERS,* pp. 179-186, 2018.

[47] D. W. Mizell, “Virtual Reality and Augmented Reality In Aircraft Design and Manufacturin,” in *1996 NAE Symposium on Frontiers of Engineering*, 1996.

[48] E. Matsas and G.-C. Vosniakos, “Design of a virtual reality training system for human–robot collaboration in manufacturing task,” *International Journal on Interactive Design and Manufacturing (IJIDeM),* pp. 139-153, 2017.

[49] S. Choi, K. Jung and Sang Do Noh, “Virtual reality applications in manufacturing industries: Past research, present findings, and future directions,” *Concurrent Engineering,* pp. 40-63, 2015.

[50] F. Covaciu, A. Pisla, G. Carbone, F. Puskas, C. Vaida and D. Pisla, "VR INTERFACE FOR COOPERATIVE ROBOTS APPLIED IN DYNAMIC ENVIRONMENTS," in *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, 2018.

[51] A. Kose, E. Petlenkov, A. Tepljakov and K. Vassiljeva, "Virtual Reality Meets Intelligence in Large Scale Architecture," in *De Paolis L., Bourdot P., Mongelli A. (eds) Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2017. Lecture Notes in Computer Science*, 2017.

[52] M. Ott and L. Freina, "A LITERATURE REVIEW ON IMMERSIVE VIRTUAL REALITY IN EDUCATION: STATE OF THE ART AND PERSPECTIVES," in *Conference proceedings of »eLearning and Software for Education« (eLSE)*, 2015.

[53] J. Fogarty, J. McCormick and S. El-Tawil, "Improving Student Understanding of Complex Spatial Arrangements with Virtual Reality," *Journal of Professional Issues in Engineering Education and Practice,* 2018.

[54] K. W. Lau and P. Y. Lee, "The use of virtual reality for creating unusual environmental stimulation to motivate students to explore creative ideas," *Interactive Learning Environments,* pp. 3-18, 2015.

[55] J. Lwowski, M. Joordens, A. Majundar, P. Benavidez, J. J. Prevost and M. Jamshidi, "The Utilization of Virtual Reality as a System of Systems Research Tool," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, Paris, 2018.

[56] S. M. N. Glegg and D. E. Levac, "Barriers, Facilitators and Interventions to Support Virtual Reality Implementation in Rehabilitation: A Scoping Review," *PM&R,* pp. 1237-1251, 2018.

[57] T. Joda, G. O. Gallucci, D. Wismeijer and N. U. Zitzmann, "Augmented and virtual reality in dental medicine: A systematic review," *Computers in Biology and Medicine,* 2019.

[58] J. M. Rosen, H. Soltanian, R. J. Redett and D. R. Laub, "Evolution of virtual reality [Medicine]," *IEEE Engineering in Medicine and Biology Magazine,* pp. 16-22, 1996.

[59] A. Rizzo and G. J. Kim, "A SWOT Analysis of the Field of Virtual Reality Rehabilitation and Therapy," *Teleoperators and Virtual Environments,* pp. 119-146, 2005.

[60] Y. C. Huang, K. F. Backman, S. J. Backman and L. L. Chang, "Exploring the Implications of Virtual Reality Technology in Tourism Marketing: An Integrated Research Framework," *International Journal of Tourism Research, Int. J. Tourism Res,* pp. 116-128, 2016.

[61] S. Barnes, "Understanding Virtual Reality in Marketing: Nature, Implications and Potential," *SSRN,* 2016.

[62] A. Lele, "Virtual reality and its military utility," *Journal of Ambient Intelligence and Humanized Computing,* pp. 17-26, 2013.

[63] A. Gaitatzes, D. Christopoulos and M. Roussou, "Reviving the past: Cultural Heritage meets Virtual Reality," in *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, Glyfada, Greece, 2001.

[64] J. P. Guerra, M. M. Pinto and C. Beato, "VIRTUAL REALITY - SHOWS A NEW VISION FOR TOURISM AND HERITAGE," *European Scientific Journal,* pp. 49-54, 2015.

[65] A. Kose, A. Tepljakov, S. Astapov, D. Draheim, E. Petlenkov and K. Vassiljeva, "Towards a Synesthesia Laboratory: Real-time Localization and Visualization of a Sound Source for Virtual Reality Applications," *JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS,* pp. 112-120, 2018.

[66] S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson and R. Nordahl, "Virtual Reality Musical Instruments: State of the Art, Design Principles, and Future Directions," *Computer Music Journal,* pp. 22-40, 2016.

[67] Wareable, "Wareable," 1 10 2018. [Online]. Available: https://www.wareable.com/.

[68] IMDA, "Info-communications Media Development Authority," 1 10 2018. [Online]. Available: https://www.imda.gov.sg/.

[69] Road to VR, "Road to VR," 1 10 2018. [Online]. Available: https://www.roadtovr.com/.

[70] J. Hanegraaf, "JOHANHANEGRAAF architectural technology," 1 10 2018. [Online]. Available: https://www.johanhanegraaf.nl/.

[71] Arch Virtual, "Arch Virtual," 1 10 2018. [Online]. Available: https://archvirtual.com/.

[72] IMDB, "IMDB," 1 10 2018. [Online]. Available: https://www.imdb.com/title/tt1677720/.

[73] Unity3D, "Unity," 1 3 2019. [Online]. Available: https://unity.com/.

[74] Epic Games, "Unreal Engine," 1 3 2019. [Online]. Available: https://www.unrealengine.com/en-US/.

[75] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[76] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos and M. Carreras, "ROSPlan: PlanningintheRobotOperatingSystem," in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[77] M. Kamel, T. Statsny, K. Alexis and R. SIegwart, "Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System," in *In: Koubaa A. (eds) Robot Operating System (ROS). Studies in Computational Intelligence*, 2017.

[78] R. A. Brooks, "A Robust Layered Control Syste For A Mobile Robot," *IEEE JOURNAL OF ROBOTICS AND AUTOMATION,* p. 14, 1986.

[79] L. Atzori, A. Lera and G. Morabito, "The Internet of Things: A survey," *Computer Networks,* pp. 2787-2805, 2010.

[80] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE COMMUNICATION SURVEYS & TUTORIALS,* pp. 2347-2376, 2015.

[81] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang and J. Rodriguez, "Massive Internet of Things for Industrial Applications: Addressing Wireless IIoT Connectivity Challenges and Ecosystem Fragmentation," in *IEEE Industrial Electronics Magazine*, 2017.

[82] J. Wan, S. Tang, Z. Shu, D. Li, S. Wang, M. Imran and A. V. Vasilakos, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE SENSORS JOURNAL,* pp. 7373-7380, 2016.

[83] G. E. Modoni, M. Sacco and W. Terkaj, "A telemetry-driven approach to simulate data-intensive manufacturing processes," *Procedia CIRP,* pp. 281-285, 2016.

[84] G. E. Modoni, A. Trombetta, M. Veniero, M. Sacco and D. Mourtzis, "An event-driven integrative framework enabling information notification among manufacturing resources," *International Journal of Computer Integrated Manufacturing,* pp. 241-252, 2019.

[85] G. E. Modoni, M. Doukas, W. Terkaj, M. Sacco and D. Mourtzis, "Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models," *International Journal of Computer Integrated Manufacturing,* pp. 1043-1059, 2017.

[86] G. E. Modoni, M. Sacco and W. Terkaj, "A Semantic Framework For Graph-Based Enterprise Search," *Applied Computer Science,* pp. 66-74, 2014.

[87] DATAFLAIR TEAM, "IoT Applications | Top 10 Uses of Internet of Things," 1 10 2018. [Online]. Available: https://data-flair.training/blogs/iot-applications/.

[88] J. Bartje, "The top 10 IoT application areas – based on real IoT project," IoT Analytics - Market Insight for the Internet of Things, 2016.

[89] Kaa , "Kaa," 1 1 2019. [Online]. Available: https://www.kaaproject.org/industrial-automation/.

[90] S. Schneider, "Chapter 3 THE INDUSTRIAL INTERNET OF THINGS (IIoT) Applications and Taxonomy," in *Internet of Things and Data Analytics Handbook*, John Wiley & Sons, Inc, 2016, pp. 41-81.

[91] P. Tsarouchi, S. Makris, G. Michalos, A.-S. Matthaiakis, X. Chatzigeorgiou, A. Athanasatos, M. Stefos, P. Aivaliotis and G. Chryssolouris, "ROS based coordination of human robot cooperative assembly task-An industrial case study," *Procedia CIRP,* pp. 254-259, 2015.

[92] TalTech, "IVAR lab," 2 January 2019. [Online]. Available: http://ivar.ttu.ee/.

Acknowledgements

This research was supported by Project AR16077 Smart Industry Centre (SmartIC) No. 2014-2020.4.01.16-0183, supported by the EU Regional Development Fund and also supported by VERT18042 Virtual Learning Factory Toolkit (VLFT) and 3-5/VERT18066 Transforming Educational Programmes For Future Industry 4.0 Capabilities (TEFFIC), funded by Erasmus. Finally, by TalTech ASTRA program "TalTech development program 2016-2022" Doctoral School of Energy and Geotechnology (2014-2020.4.01.16-0032), funded by EU Regional Development Fund.

I express my warmest thanks to my friends, colleagues and especially family – my wife, child, parents and grandparents for supporting me in the journey of achieving this milestone in my life.

Separately, I wish to express my sincere gratitude to my supervisors: Professor Tauno Otto and Associate Professor Toivo Tähemaa, who believed in my novel ideas in the field of Mechanical Engineering and always kept me motivated and enthusiastic about the things that I have been doing. Thanks to them, my inspiration was never lost and my best gratitude to them is to keep doing what I am doing and to achieve their level one day.

Abstract

Novel Digital Twin Development Methodology for the Robot Cell Connectivity in a Smart Industry Environment

The robotisation of manufacturing is essential for automating the processes in a smart factory. However, the solutions are mainly robust and not entirely safe for the operator and factory workers. The methods of programming need skill development, and the most precise manual approach demands a long time for re-programming of the machines. Offline and online methods, however, can make a robot programmable from a remote location, but not effective for re-programming and adaptation in the surrounding environment. In the scope of Industry 4.0, there is a demand for new ways for manufacturing lines to digitalise as well as for a safe and easy to use the machinery control from a remote location.

The primary objective of the current study was to develop an immersive simulation based on the principles of Industry 4.0 and to exploit Digital Twins concept not only in the means of simulation but to create a DT creation methodology of the dual-way synchronisation of real world industrial equipment with its digital replica based on the example of the Industrial Robotic cell in order to be able to manage and control the factory from the simulation in real-time.

A substantial part of the study analyses the developments and approaches in the field of Smart Industry and combines research in novel Industrial Robot programming, the DT concept, Virtual Reality (VR) technologies, the Robot Operation System (ROS) and the Industrial Internet of Things (IIoT).

The experimental part of this research starts with an introduction of the experiments on industrial programming methods for a robot with an attached scanner. During the investigation of safe and collaborative programming methods for the operator, it was realised that there is a lack of methodologies for smart simulations by means of a dual-way simulation. Thus, the first milestone of the dissertation was the creation of an architecture methodology for dual-way synchronisation of the DT with its source, and safe and effective ways for the adaptive industrial robot algorithms to be introduced to part of the framework by connecting them with physical and virtual sensors.

Concluding experiments created an immersive environment based on the example of the FMS and robotics research laboratory, enabling the teleoperation of industrial robots from a VR simulation. Moreover, an operator in real life could cut off the connection and take over control. In addition to a modular and universal user interface, a collision system in DT was introduced, which prevented the real robot in movement form from unintentional colliding with unexpected surrounding objects/people.

A crucial part of this dissertation was the elaboration of novel connectivity methods. Based on the architecture developed, IIoT technologies were developed allowing technological devices from any platform such as VR headsets, desktops and mobile applications to connect to the created environment. Moreover, a multiplayer and collaborative connection was created by the connectivity of the laboratories in Italy (STIIMA-CNR) and Estonia (TalTech).

The main goal of the study was achieved. A base open source solution for fully syncronised Digital Twin environments development was created. Fully modular within the means of connectivity and equipment, the novel methodology of the design of Digital Twins was introduced.

Lühikokkuvõte

Uudne digitaalsete kaksikute arendusmetoodika robottootmisrakkude sidustamiseks targa tööstuse keskkonnas

Tootmise robotiseerimine on targa tehase protsesside automatiseerimise oluline osa, kuid enamik kasutusel olevatest lahendustest on lihtsad ning pole operaatorile ja töötajatele täiesti ohutud. Programmeerimismeetodid vajavad oskuste arendamist ja täpseim käsitsi programmeerimise metoodika nõuab tehnoloogiaseadmete ümberprogrammeerimiseks pikka aega. Offlain- ja onlainmeetodid võimaldavad roboteid programmeerida distantsilt, kuid pole ümberprogrammeerimise ja ümbritseva töökeskkonnaga kohastuvuse seisukohast efektiivsed. Tööstus 4.0 raamistikust lähtudes on vajadus uute meetodite järele, kuidas tootmisliine digitaliseerida ning neid siis ohutult ja lihtsalt teisest asukohast kaugjuhtida.

Selle doktoritöö põhieesmärk oli arendada tehisreaalsuses simulatsioonikeskkonda Tööstus 4.0 põhimõtetest lähtuvalt, kasutades digitaalsete kaksikute kontseptsiooni mitte ainult simulatsioonivahendite, vaid kahesuunaliselt sünkroniseeritavate digitaalsete kaksikute loomise metoodika arendamiseks, mis võimaldaks tööstusrobotite tootmisraku näitel hallata ja juhtida tehast simulatsioonikeskkonnast reaalajas.

Doktoritöö põhiosas on analüüsitud targa tootmise arenguid ja lähenemisviise ning esitatud kokkuvõtvalt uuringud uudse tööstusrobotite programmeerimise, digitaalsete kaksikute kontseptsiooni, virtuaalreaalsuse tehnoloogiate, robootika operatsioonisüsteemi ROS ja tööstusliku nutistu osas.

Doktoritöö eksperimentaalses osas on katsetatud roboti tööstuslike programmeerimismeetodite kasutamist skaneerimisrobotil. Programmeerimismeetodite analüüsil selgus, et nutistu simulatsioonipõhiseks programmeerimiseks puuduvad simulatsioonide kahepoolse sünkroniseerimise metoodikad. Väitekirja põhiraskuseks oli robotiseeritud töökoha digitaalse kaksiku füüsiliste ja virtuaalsete andurite kahesuunaline sünkroniseerimine, ning seeläbi adaptiivsete tööstusrobotide tehisreaalsusest ohutu ja efektiivse programmeerimise algoritmide metoodika väljatöötamine .

Eksperimentide sooritamiseks loodi ümbritsev keskkond paindtootmise ja robootika uurimislabori näitel, mis võimaldab tööstusrobotide teleoperatsiooni VR-simulatsioonist. Reaalne operaator võis ühenduse katkestada ja juhtimise käsitsi programmeerimismeetodil üle võtta. Lisaks modulaarsele universaalsele kasutajaliidesele rakendati digitaalses kaksikus ka kokkupõrke hoiatussüsteemi, mis takistab reaalse roboti liikumisel soovimatut kokkupõrget ümbritsevate objektidega/inimestega.

Väitekirja oluliseks osaks oli uute ühenduvusmeetodite väljatöötamine. Uudsel arhitektuuril põhinevad IIoT tehnoloogiad võimaldasid ühendada tehnoloogiaseadme mistahes platvormilt VR-peakomplekti, laua- ja mobiilirakendustena loodud keskkonnaga. Veelgi enam, tehisreaalsuses töötav ühendus loodi ka Itaalia STIIMA-CNR ja Eesti TalTech laborite vahel.

Uuringu peamine eesmärk saavutati: loodi baaslahendus avatud lähtekoodiga täielikult sünkroniseeritud digitaalsete kaksikute keskkondade arendamiseks. Loodud sai ühendusvõimaluste ja seadmete osas modulaarne digitaalsete kaksikute projekteerimise metoodika.

Appendix

Publication I

Kuts, V.; Tähemaa, T.; Otto, T.; Sarkans, M.; Lend, H. (2016). Robot manipulator usage for measurement in production areas. Journal of the Machine Engineering, 16 (1), 57−67.

Journal of Machine Engineering, Vol. 16, No. 1, 2016

Received: *25 January 2016*/ Accepted: *01 February 2016*/ Published online: *10 March 2016*

*industrial robot, production engineering,
3D scanning, measuring,
inspection, manipulator*

Vladimir KUTS[1*]
Toivo TAHEMAA[1]
Tauno OTTO[1]
Martins SARKANS[1]
Henri LEND[1]

ROBOT MANIPULATOR USAGE FOR MEASUREMENT IN PRODUCTION AREAS

Measuring and inspection works in production process using laser scanners attached to the robot manipulators are more and more used in different ways and purposes. The aim of this article is to study and analyze the usage of 3D measuring devices attached to the robot manipulator according to determine the range and the purpose of such industrial measurements. Different scanning devices are applied in production areas with varying purposes, such as 3D scanning of the new product, in order to inspect and find defects. Attached to the robot manipulator, scanners or laser measuring devices are able to scan every object in range of a robot, which makes the measuring process very flexible. Several tests with 3D scanners were performed to find out the optimal conditions and configuration for production purpose. As a result, knowledge base suitable for different production areas was developed. Thus, it would be feasible to have a common database to be implemented in different production areas.

1. INTRODUCTION

The possibility of fast and precise measurements by using measurement tools attached to the robot manipulator has raised up a wide interest in fields of production and quality control. There are many ways to use those devices in different purposes. It is possible to find weaknesses in structure, inspect detail's geometry, make a digital copy of its geometry and put it into an assembly in the drawing made in CAD software.

However, previous researches [1],[2],[6] have not properly determined the field and range of the usage of 3D robot scanning. There is clear need for a methodology of such a robotic measurement systems to evaluate their suitability according to object size and geometrical complexity.

---

[1] Tallinn University of Technology, Department of Machinery, Tallinn, Estonia
[*] E-mail: vladimir.kuts@ttu.ee

2. HARDWARE OVERVIEW AND METHODOLOGY

This section will give an overview of different types of manipulators and measuring devices attached to them.

2.1. ROBOT MANIPULATORS

The introduction of industrial robots is an accelerating trend of the recent years. Between 2005 and 2008 the number of new robots built was approximately 110,000 per year. From 2011 onwards, it exceeded the verge of 150,000 robots, and in 2014 the number was 229,000 [14]. Major exploiters of robots are automotive and electronics industries.

There are many different manipulator manufactures in the world. The most well-known brands are Kuka, ABB, Motoman, Fanuc, Comau, Panasonic, Nachi, Adept, Kawasaki, OTC, Hirata, ESAB, Hyundai, Honda, Deneb, Lincoln, etc. Experimental part given in this paper was done by using ABB IRB 1600 robot manipulator with 100 N lifting abilities.

Industrial robots have a wide field of usage depending on the possibility to automate the production technology. For a long time period the development of robotics was motivated by mass production only. Those appropriate tasks like welding, painting, assembly, parts feeding, packaging e.c.t) were in the focus. Today, the cost and ease of deployment of robots have made the usage available to small-scale production and small businesses as well. This forces robot and software makers to develop applications that meets also needs of SME [9] production.

As it was mentioned before – robot manipulators can be used in different areas: welding (arc welding, spot welding, friction welding), product packaging and palletizing, assembly process, finishing, gluing, painting, machining (wood, plastic, aluminum), cutting (laser cutting, plasma cutting), surface scanning for the purpose of measuring (laser scan), or inspecting. This list is not exhaustive and that shows how widely industrial robots are used in the development of the manufacturing processes.

Some authors of this article have experienced in the field of industrial robotics in arc welding (MIG/MAG) uptake [7]. Its implementation, deployment and appropriateness of the SME environment have been covered in previous articles [4],[5]. As was researched [10], in case of parts with reflective surfaces the selection of proper 3D scanning orientations can significantly reduce outlier extensity. Thus scan path planning knowledge has value in case of using industrial robot manipulators, where repeating paths is configurable.

The purpose of this article is to further analyze novel technological processes where industrial robots can be implemented to increase productivity. Those novel fields are 3D measurement tasks, inspection works and quality control. The emphasis is to develop methodology suitable for SME-s having a need for flexible and rapid measuring-inspection solutions. During the research a knowledge base of different products (profiles, materials, colors) was developed in order to be measured/inspected by the industrial robot solution.

2.2. ROBOT-AIDED 3D SCANNING

Nowadays optical 3D scanners are gaining more usage in production as non-contact quality assurance tools with high capability. Non-contact type scanners (optical scanners) have some advantages over contact type scanners (CMM). Because of high accuracy and capability, they are mostly suitable in quality control systems where large datasets of geometry must be observed. These are suitable for both - small (i.e. measuring the wear of cutting inserts with high resolution) and large objects (i.e. castings or welding constructions) [6],[7],[8],[9]. Non-contact scanners are also preferred as they can be used in automated scanning processes with a robot manipulator. For example, in the automotive industry for automated online quality control to check geometrical errors occurring during manufacturing process [13].

Considering the working principle, mainly two different types of scanners are used [11]: 3D laser scanners and Structured Light Scanners. 3D laser scanners may use laser triangulation, time of flight or phase shift method. Laser triangulation is accomplished by projecting a laser line or point onto an object. Then a sensor is capturing laser beam reflections at a known distance from the laser source.

Laser triangulation is most used because it has the following advantages over other methods:

- can be used in various indoor lightings,
- can be used to scan parts of any material,
- provides excellent measuring resolution.

A disadvantage of the laser triangulation method is laser beam is not eye safe.

The other common scanning method is structural light or white light scanning. This technique utilizes 2D light pattern (zebra stripes) which are projected and moved on the object surface. At the same time two CCD cameras are recording the pattern and through triangulation the complex surface parameters are calculated. The advantages of white light scanning are following:

- provides good accuracy,
- fast measurement,
- eye safe.

Disadvantages of the white light method are:

- sensitive to ambient light,
- cannot be used to scan shiny surfaces,
- some trouble to scan very detailed parts with many ribs and sharp features.

In the experimental part we used two types of scanners. The first model was ATOS II 400 optical 3D scanner from GOM (Fig. 1). ATOS utilizes structural light method and its measuring speed is 1.4 million points in 7 seconds [12],[13]. Sphere spacing error of ATOS system is 0.026 mm according to 3D scanner standard VDI 2634. Scanning resolution is 0.17 mm. ATOS uses uncoded markers which have been glued on the object surface and used for merging different scanning images.

The second model was Nikon MMDx100, together with the K600 Touch Probe System and ABB IRB robot called also as Nikon K-Robot Automated Scanning System (Fig. 2). Nikon MMDx100 scanner work principle is based on laser triangulation and its scanning accuracy is 10 μm. Overall accuracy for the Nikon system is depending on the measuring zone size. For the zone II the accuracy is 90 μm + 25·L μm [15].

Fig. 1. 3D scanning by ATOS II 400

Fig. 2. Scanning by Nikon K-Robot scanning system

During the measurement procedure the object should not been moved with respect to K600 tracking head. Nikon K-Robot is optimized for production environments that require full-time part inspection in changing environments. With the K-Robot Automation software fully automated scanning tasks can be done with optimal speed.

2.3. METHODOLOGY OF MEASURING

The measurement setup is similar to [3], but instead of turntable a tracking head was used. Measurement system consists of many different parts: PC with software, measuring 3D head, tracking head, manipulator and controller. For measuring the sensors attached onto the measuring head must be seen by remote stand, whereas three of them should be seen at the same time. Otherwise, the software will not recognize the measured product. Also, there are two different laser scopes going from 3D head – laser line and laser point, they must be as close as possible to each other while the object is measured.

In the measurement process it needs to be confirmed that all dark/shadowed areas are covered. It means that for every measured object the manipulator movement driving program must be unique. Every manipulator producer has their own curve design software. ABB, which robot is being used by Department of Machinery in Tallinn University of Technology, has RobotStudio software. For the robot mentioned above, manual programming (flex-pendant) is possible as well. By moving robotic manipulator with the help of Joystick, you determine the manipulators track points, speed and curve passage percentage.

3. RESARCH OF FIRST-TIME OBJECT INSPECTION

For first-time measurements, which means that picked up object is being scanned for the first time, preparations are needed. Different environmental conditions and characteristics of the measured object must be taken into account. For instance, parameters like environment temperature, mirror zones, lights, furniture etc. must be observed before processing. In addition, from the object point of view, the surface of the object and its placement are remarkably important. The size and geometrical shape of the object must be considered before one starts creating the measuring program. All objects shown in the current work were chosen randomly according to their surface, material and geometrical form differences.

Measurement device should be properly attached to the robotic manipulator and a suitable program of the manipulator movement should be created. Both of those tasks influence the amount of measuring points. The more measuring points are acquired from the object, the more precise the CAD model is. Different drive parameters will help us to increase the amount of measuring points. When forcing a robotic manipulator software program to lower the speed of the manipulator, more measuring points will be captured. Low speed is an option only in case of measurements that are non-time-critical like in educational processes or in art. Then the option is lowering the measurement zone in scanning software.

3.1. OBJECT INSPECTION WITH NIKON 3D SCANNER

The first inspected object is a low power frequency converter with dimensions 150x70x147mm (see on Fig. 3, left-side). The measured object is made of aluminum and plastic. The object has a simple geometrical form. The running program lasted 18 minutes and there were 61 steps in the movement program. As a result, all scanned object lines can be seen on right-side of Fig. 3. The few scanning errors have been marked with red circles.

The second inspected object is a blade with dimensions 144x76x40mm (see on Fig. 4, left). The measured object is made of 3D printed plastic. The object has a simple and round geometrical form. The running program lasted 2 minutes and the number of movement program steps was 31. The result can be seen on Fig. 4, right-side.

Fig. 3. Inspected object 1 (original and result)

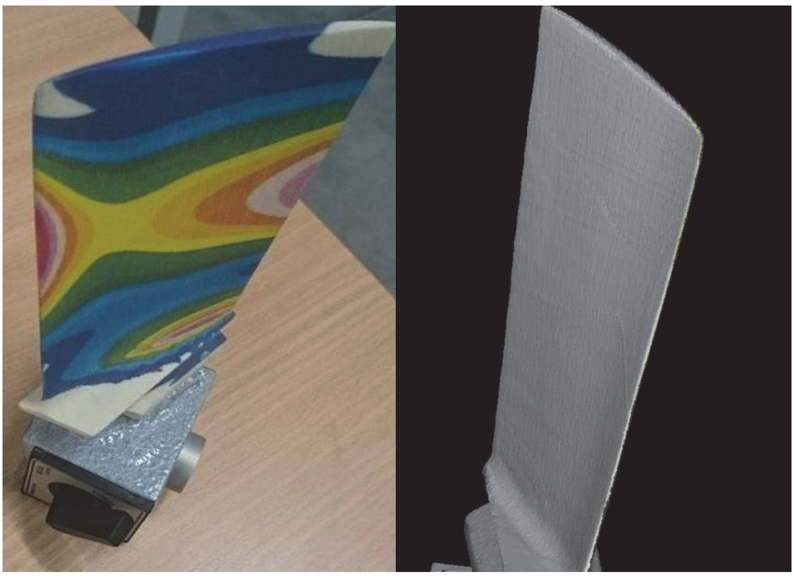


Fig. 4. Inspected object 2 (original and result)

The third inspected object is a part of a ball with diameter 175 mm (see on Fig. 5, left-side). The measured object is made of rubber. The object has a round geometrical form. The running program lasted 5 minutes and the number of movement program steps was 64. The result can be seen on Fig. 5, right-side. The scanning errors have been marked with red again.

Fig. 5. Inspected object 3 (original and result)

3.2. ATOS AND NIKON INSCPECTIONS COMPARISON

Additional inspection of those three objects were done with ATOS white light scanning system. It was done manually in a purpose to have the precise inspection model of the object as much as it is available to do with this system. This process is not described in the current paper but it will be described later in the other article.

Table 1. Objects – Global comparison (mm)

Object	1	2	3
Number of valid points	307 799	544 115	524 432
Maximum Deviation	1.758	4.962	8.707
Minimum Deviation	-1.903	-2.499	-9.312
Range	3.662	7.461	18.018
Mean Deviation	0.122	1.787	0.187
Sigma	0.086	1.540	1.005
Root Mean Square	0.150	2.359	1.023

Comparison of the same objects inspections between manual ATOS scan and automated NIKON scan were performed. Inspection with ATOS system were taken as initial value, because it was done manually and every measured object area was covered.

Measurement done by NIKON 3D scanner attached to the robot manipulator was compared with this initial value. Deviation is shown in tables – numbers show the difference between NIKON system values to the initial values measured with ATOS system. Deviation tables and pictures are shown also to have a better visual overview (see on Table 1 and Fig. 6, 7). Surface deviation analysis were used to obtain the results.

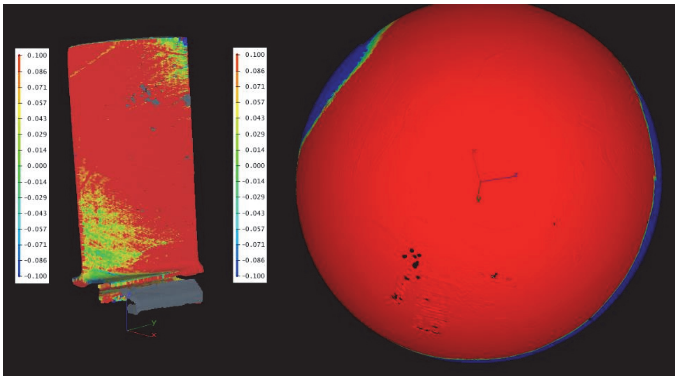


Fig. 6. Deviation analysis of Objects 1 and 2

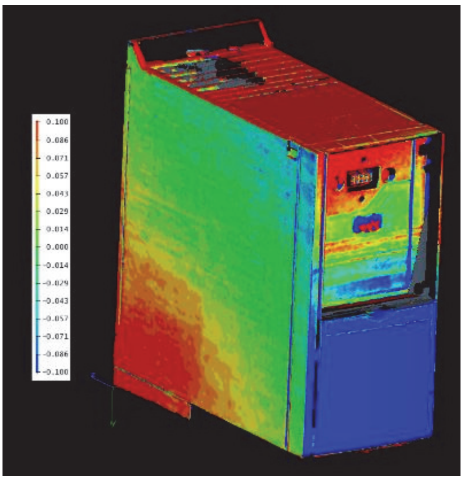


Fig 7. Deviation analysis of Object 3

4. RESULTS AND DISCUSSION

The result of the first-time measurements of three objects have shown us different approaches and the direction to move ahead. Methodology for each type of object is being developed. Generally, the scanning for Object 2 and Object 3 took essentially less time than scanning Object 1 due to experimentations with scanning speed. The fastest scan speed was chosen for Object 2 and the slowest for Object 1. It provided an insight that lower speed does not necessarily give the best result.

Analyzing the measurement results of Object 1, we see that it is not ideal. On the points 1 (Fig. 3), we see the shadow issue, when the laser head measured this side only from up to down and appeared a blank zone on the CAD model. To avoid it in the future, all hard places on the object must be scanned from different directions. Point 2 (Fig. 3) appeared because of the changes in distance between measuring head and object. As the laser line and point must be as close to each other as possible while measuring a flat surface, in case of some deviations (bump), it must be considered that the measuring distance will be still the same, and not to lose the measurement points.

Analyzing the measurement results of Object 2, it can be said that the measurements were successful and previous mistakes from Object 1 were corrected. It can be seen that there are no shadow areas on the result.

The measurement results of Object 3 were successful, but only one issue appeared during the scan, as it seen on Point 1 (see on Fig. 5). There are some losses in the scanned surfaces structure, which are the result of surface reflection, mirroring the surface at this point. The light was falling mostly into this place on the object.

The inspection results comparison done with two different systems have shown us, that the 3D scan using laser scanner is more detailed as a resolution is higher. Deviations were in the reason that NIKON system measures more details as letters, surface changes, paint. This have shown, that more accurate inspection can be done using laser 3D scanner.

4.1. AREA OF USAGE

The area of usage in this technology is large. It can be used in every type of production – e.g. food, machinery, electronics, and medicine.

It can be used in every manufacturing process as an online geometry inspection. Experiments of inspection of 3D printed details have been done to determine their shape and size preciseness. Moreover, a choosing process for spare parts in repair business is possible as well, according to the measured object. After it is converted into a CAD model, all dimensions are determined. According to this information right spare parts can be chosen.

In medicine this technology can be used in the branch of prosthetics and others related to that technology.

4.2. EASE OF USAGE/EDUCATION, LEARNING PROCESS

To use measuring devices attached to the robot manipulators, the person responsible must have different knowledge. First of all, knowledge about controlling the robot must be as good as possible. How to program its movement; use its software, control with flex pendant - all that is more than important in order to succeed in the scanning process. Knowledge of programming language is an advantage.

If a person knows how to use a robotic manipulator then it is easier to use the measuring head properly. One must surely know how to calibrate it, run it and what rules must be considered while measuring. By knowing all the basic processes mentioned above, one can proceed every type of measuring and control. So this type of basic trainings is unavoidable for every engineer related with automated measuring mechanics in production lines, educational process and in arts.

4.3. FUTURE INVESTIGATIONS

In the future, the methodology for each type and size objects must be carried out correctly. It will help carry out one-time measurements with minimum changes to the software to create the needed models quickly. It will be a big step forward to helping small organizations with no massive manufacturing capacities.

5. CONCLUSION

Analysis of robot manipulators and different technologies of measurement devices which can be attached to them was carried out. Measurement systems were configured and set up to measure different types of objects.

Three different objects were measured. The analysis of the results provided us with conclusions about the methodology of measurement. From those results we can see what must be considered in each type of measurement and what to work on in the future.

The next step should be creating a measurement methodology for different objects and gathering the data into one database for the different usage purposes.

REFERENCES

[1] BAUER J.M., BAS G., DURAKBASA N.M., KOPACEK P., 2015, *Development Trends in Automation and Metrology*, IFAC-PapersOnLine, 48/24, 168-172.

[2] LARSSON S., KJELLANDER J.A.P. 2008, *Path planning for laser scanning with an industrial robot*, Robotics and Autonomous Systems, 56/7, 615–624.

[3] LI J., CHEN M., JIN X., CHEN Y., DAI Z., OU Z., TANG Q., 2011, *Calibration of a multiple axes 3-D laser scanning system consisting of robot, portable laser scanner and turntable*, Optik, 122/4, 324-329.

[4] SARKANS M., EERME M., 2011, *Programming time estimation and production planning steps on welding robot cells in SME-s*, Annals of DAAAM for & Proceedings of the 22nd International DAAAM *Symposium*, Ed. Katalinic, Branko. Vienna, Austria: DAAAM International Vienna, 1485-1486.

[5] SARKANS M., ROOSIMÖLDER L., 2010, *Implementation of robot welding cells using modular approach*, Estonian Journal of Engineering, 16/4, 317-327.

[6] SHIBIN Y., YONGJIE R., YIN G., JIGUI Z., SHOURUI Y., SHENGHUA Y.E., 2014, *Development and calibration of an integrated 3D scanning system for high-accuracy large-scale metrology*, Measurement, 54, 65-76.

[7] TALALAEV R., SARKANS M., LAANSOO A., VEINTHAL R., 2012, *Methodology for Configuration of Robot Welding Cell for SMEs Under Conditions of Small and Medium Sized Production Using MIG/MAG Process*, Proceedings of the 8th International Conference of DAAAM Baltic Industrial Engineering, Ed. T. Otto. Tallinn: TUT Print, 591−596.
[8] TÄHEMAA T., TEMERBULATOVA A., KARJUST K., 2012, *Lean production development in Estonian SME's*. Proceedings of the 8th International Conference DAAAM Baltic.
[9] VAGOVSKÝ J., BURANSKÝ I., GÖRÖG A., 2015, *Evaluation of Measuring Capability of the Optical 3D Scanner*, Procedia Engineering, 100, 1198-1206.
[10] WANG Y., FENG H-Y., 2016, *Effects of scanning orientation on outlier formation in 3D laser scanning of reflective surfaces*, Optics and Lasers in Engineering, 81, 35-45.
[11] A guide to 3D scanner technology [Online] http://www.rapidform.com/3d-scanners/(20.01.2016)
[12] ATOS Professional [Online] http://www.gom.com/3d-software/atos-professional.html (14.01.2015)
[13] GOM System Overview [Online] http://www.gom.com/metrology-systems/system-overview.html (16.01.2015)
[14] Industrial Robot Statistics, [Online] http://www.ifr.org/industrial-robots/statistics/ (20.01.2016)
[15] Nikon Metrology [Online] http://www.nikonmetrology.com/en_EU/Products/PortableMeasuring/Optical-CMM/K-Series-Optical-CMM/%28key_features%29 (14.01.2015)

Publication II

Kuts, V.; Sarkans, M.; Otto, T.; Tähemaa, T. (2017). Collaborative work between human and industrial robot in manufacturing by advanced safety monitoring system. In: B. Katalinic (Ed.). DAAAM International (0996−1001). Vienna, Austria: Curran Proceedings.

DOI: 10.2507/28th.daaam.proceedings.138

COLLABORATIVE WORK BETWEEN HUMAN AND INDUSTRIAL ROBOT IN MANUFACTURING BY ADVANCED SAFETY MONITORING SYSTEM

Vladimir Kuts, Martins Sarkans, Tauno Otto, Toivo Tähemaa

Tallinn University of Technology, School Of Engineering, Department of Mechanical and Industrial Engineering, Ehitajate tee 5, 19086 Tallinn , Estonia

This Publication has to be referred as: Kuts, V[ladimir]; Sarkans, M[artins]; Otto, T[auno] & Tahemaa, T[oivo] (2017). Collaborative Work Between Human And Industrial Robot In Manufacturing By Advanced Safety Monitoring System, Proceedings of the 28th DAAAM International Symposium, pp.0996-1001, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/28th.daaam.proceedings.138

Abstract

Collaborative work between human and industrial robot in different areas is developing rapidly. High payload industrial robots that can harm humans execute their tasks in separated protected areas and are regulated by safety standards. The main aim of this study was to design experiments, analyze, and model the collaborative work and safety monitoring systems by using small industrial robots with inbuilt power and force limiting. Different methods of collaboration are standardized by the ISO/TS 15066:2016. Human co-work with robots gives us a variety of options that can be used in manufacturing and other related areas. The advanced multi-level monitoring system of co-work processes can reach the highest level of the human safety with minimum robot downtime. Several methods were analyzed to find the optimal solution for the online monitoring processes. As a result, a safety multi-level monitoring system was designed, tested in simulation software with real-room conditions with an industrial robot. Safety multi-level system, as a part of the control and monitoring online system consist of different types of sensors and a microcontroller that can be attached to the Cyber Physical Production System (CPPS). Those control and monitoring methods and processes were implemented in experimental setup in Robot Operating System (ROS).

Keywords: collaborative industrial robots; Cyber Physical Production System; Robot Operating System

1. Introduction

The collaboration between industrial robots and humans is a widely-discussed topic nowadays. Many research papers have been published in this field, but there are still a few uncovered areas, mostly relating to the devices ranging between collaborative robots and small simple industrial robots. There is also lack of optimized safety systems, which could fulfill all safety requirements and manufacturing needs.

Different system components, software, and other tools were used in making proper methodologies for studying industrial robot and human collaboration [1], whereas ISO/TS 15066:2016 is the latest recognized standard for this kind of systems. As the development process in the field of collaborative robots is so rapid, we predict that this standard gets updates in the nearest years.

1.1. Problem statement

There are two main problems in small industrial robots and human worker collaboration systems:

- Human safety monitoring system control
- Downtime of the industrial robot, while safety system is activated

In the modern robot systems, every large manufacturer has developed its own controller and software where off-line programming (OLP) is done. In this kind of systems, industrial robots are executing their tasks and moving according to the pre-programmed path which is already done before the real work is performed. To gain better control of the movement of the industrial robot it must be possible to affect robot behavior online, while the task is being executed. In this paper, we demonstrate a method for online monitoring and control of the robot with the help of the robot operating system (ROS). In the experimental part of this article, the comparison of the off-line programming (OLP) vs online control, and monitoring method is shown, which was enabled with the new computer technologies and machine vision applications.

1.2. Background and current situation

Non-collaborative industrial robots must usually comply with safety requirements described in standards ISO 12100, ISO 13850, ISO 13855 etc. As there is no need for human intervention during the manufacturing process (i.e. painting, welding, deburring, etc.) strict rules exist for the safety of the human and for emergency halts of the industrial robot. Standards and guidelines describe the allowed speeds, distances, required safety equipment and safety procedures to establish the required safety. When human enters into the workspace of the robot, the process (and the robot) is stopped immediately. To restart the production process, the fault situation must be cleared (worker must leave from the workspace or remove the obstacle) by resetting emergency button outside the cell. As the production processes are growing more complex and interconnected, the human and robot collaboration principles development has become an important issue during the last years. Industrial robots have advantages in areas (processes) where speed, power, repetitiveness, and durability is needed. The human worker can be added into the process to increase agility and flexibility (if the process or product changes rapidly).

Different authors have researched human-robot collaboration topics such as:

- ROS based coordination - For coordination of assembly tasks between human and robot an ROS based software architecture is used. Human and robot tasks are simulated using OLP tool and tasks are recalled using graphical user interface (GUI) thus enabling to separate the tasks for a robot and human operators. [2]
- Human safety - Concept of kinetostatic safety for human-robot collaboration is introduced and its computational methodology is presented. [3]
- Assembly cell - Methodology for task assignment and scheduling for human-robot cooperation (HRC) assembly cell. By using ROS software platform the overall framework for HRC is developed. [4, 5]
- Assembly Factories of the Future (FoF) - Project ROBO-PARTNER is introduced. Its main goal is to develop the integration platform for safe human-robot collaboration (HRC). Different areas like safety, collaboration tasks planning, robot programming, and integration are considered. [6]
- Speed and separation monitoring – Developing a solution for human-robot collaboration in the automotive industry (PSA) for the assembly process. [7]
- Sensors – implementation and integration of different types of sensors to establish required safety for human-robot collaborations [8]

ISO 15066 standard gives advice for human-robot co-operation safety issues. These are recommended for the development of robot cell solutions. By standard, the workspace is divided into two parts: robot workspace (operating space); and collaborative workspace. Collaborative workspace is defined as "space within the operating space where the robot system (including the workpiece) and a human can perform tasks concurrently during production operation" [9]. As long as the industrial robot operates in its allowed operating space then the general rules of safety are applied (robot stops immediately when someone enters the robot workspace). As robot enters into collaboration workspace, the standard ISO 15066 must be applied. Our research paper concentrates mainly on collaboration workspace issues.

Usually, the robot is programmed by using the teach pendant or offline programming (OLP) method. After the program is simulated in the computer, it is uploaded to the robot controller storage memory. To execute this program on the robot the program is uploaded into program (RAM) memory. During this step, the program syntax is controlled to prevent any faulty code or program.

This kind of programming approach is quite rigid and usually does not include any human safety factors. The program can be stopped by using the external stop command, additional sensors, safety stop, or a multi-task option. However, in real-life conditions, these are not flexible and consume additional process time. Also, the response of the robot controller can be too slow to halt (or decelerate) the robot before the human gets harmed.

The proposed solution described in this article is to unload the program (connected to the production process) from the robot controller and to convert robot into the client mode for listening only "external" commands from the server. After the conversion, the program can be divided into smaller subprograms and the safety commands and parameters can be added.

Some of the authors have contributed to the robot communication, machine vision, and robot trajectory planning:

- Planning of the trajectory for the robot by using an external camera for this purpose. [10]
- Communication principles between networked robots and remote control solutions are described [11]
- Use of Robot Operating System (ROS) for implementation and programming of mobile robots for process [12]
- Use of computer vision for mobile robots trajectory planning and mapping [13]

Abovementioned concepts can be used as a base for further development also in this case. Although this case includes only one robot in future several robots will be included in this study.

2. Methodology

Both of the solutions - robot path planning, and preliminary program - are done in OLP software. The difference can be seen on control and execution level.

2.1. OLP and online programming comparison

Many different tools exist for OLP programming – originally all industrial robot manufacturers – ABB, Kuka, Fanuc, Universal Robot, and much more, have their own tools for giving the robot the ability to execute the exact task – welding, measurement, or grabbing depending from the production line/user needs. This way of making a program is nowadays mostly used by users and robotic cell system planners [14]. Generally, the system consists of the robot controller, industrial robot manipulator and a number of sensors. Additional specific components can be added to the system. (See. 1.).

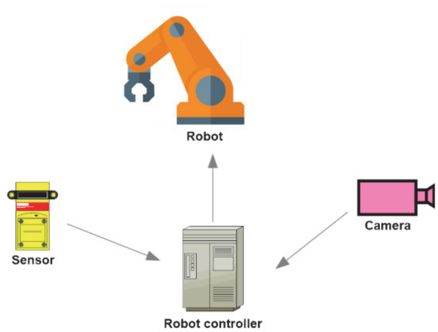


Fig. 1. Robot path executed from OLP

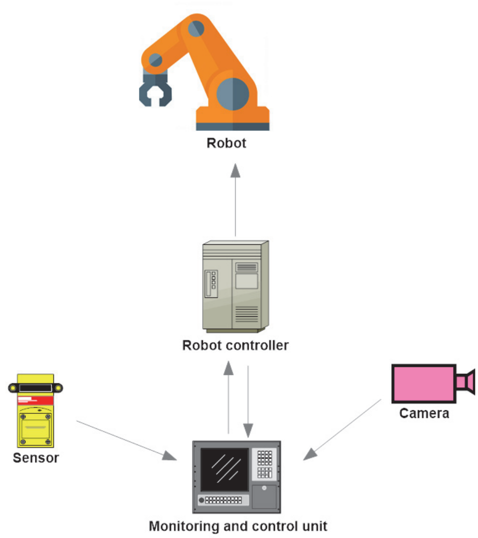


Fig. 2. System with online control

With the online control and monitoring method program for the industrial robot, the path is still programmed by manufacturer's software, but we add an additional control tool, as ROS for example which can be run on a separate controller, or server machine (See Fig. 2.).

This method gives the system unlimited abilities for:

- process monitoring and control;
- program online modifications;
- automated decision making;
- advanced safety system (security levels); and
- automatic path planning with the camera.

For example, if we have rapid code command which consists of:

- MoveL, which is linear movement command
- XYZ – coordinates in the system
- V – speed in mm/sec
- Z – curve on path while moving through coordinates

This is enough to execute the task from point A to point B (see Fig. 3.). However, to gain additional artificial intellect to our robot program we need to add an additional factor to the code. Fully effective usage of this can be done via online monitoring and control tool.

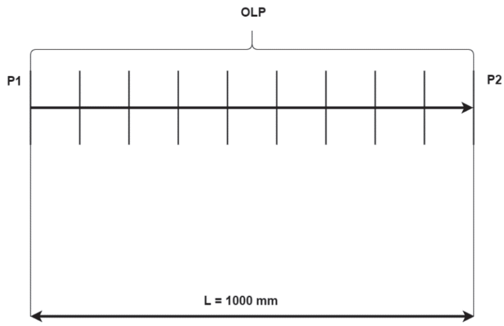


Fig. 3. Pre-programmed robot path

3. Research

3.1 Design of Experiment

An additional factor in our system is security level of each trajectory part. As in the previous study with OLP, we have 10 coordinates, which must be passed through by the industrial robot according to the pre-done program. By adding the additional factor, it is possible to give to every one of those points a security level on a scale from 1 to 10, where 10 means "highly secure". So, each step will be monitored with the refresh rate of 0.01 seconds online during the whole program execution progress. If sensor or camera will detect any object in the range of the industrial robot, additional control unit, will decide and give a command to shut down, go to sleep mode, change the path or jump over to the next task. So, as the safety status drops (somebody or something enters the area), the parameter gets a smaller value (0...9) depending on the distance between the object and the duration of the breach. And when the object/human leaves the zone, the program automatically gives permission to proceed with the same step or to go back to the not performed task. On the next round of the program, control unit already remembers where the fault appeared and moves with an exact level of additional notice into that coordinate, and reduces it with every cycle (See Fig.4).

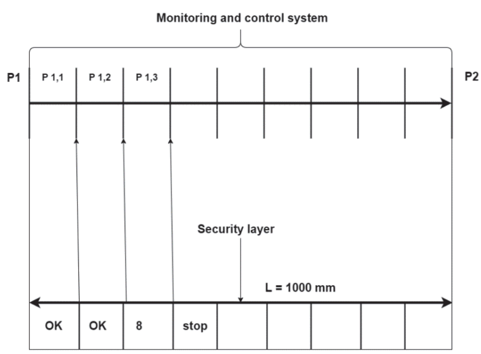
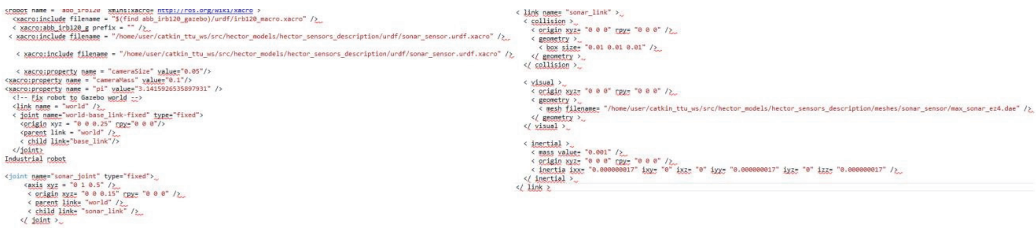


Fig. 4. Robot path with online monitoring

3.2. Case study – safety system example

The example was done using ROS [15] and ABB robot IRB120. Industrial robot and sonar scanner with data about objects, joints location coordinates and physical parameters (See Fig.5) were added to the system.



```
<robot name = "abb_irb120" xmlns:xacro="http://ros.org/wiki/xacro" >
  <xacro:include filename = "$(find abb_irb120_gazebo)/urdf/irb120_macro.xacro" />
  < xacro:abb_irb120_g prefix = "" />
 < xacro:include filename = "/home/user/catkin_ttu_ws/src/hector_models/hector_sensors_description/urdf/sonar_sensor.urdf.xacro" />

   < xacro:include filename = "/home/user/catkin_ttu_ws/src/hector_models/hector_sensors_description/urdf/sonar_sensor.urdf.xacro" />

   < xacro:property name = "cameraSize" value="0.05"/>
<xacro:property name = "cameraMass" value="0.1"/>
<xacro:property name = "pi" value="3.1415926535897931" />
  <!-- Fix robot to Gazebo world -->
  <link name = "world" />
  < joint name="world-base_link-fixed" type="fixed">
    <origin xyz = "0 0 0.25" rpy="0 0 0"/>
    <parent link = "world" />
    < child link="base_link"/>
  </joint>
Industrial robot

<joint name="sonar_joint" type="fixed">
      <axis xyz = "0 1 0.5" />
       < origin xyz= "0 0 0.15" rpy= "0 0 0" />
       < parent link= "world" />
       < child link= "sonar_link" />
    </ joint >
```

```
< link name= "sonar_link" >
  < collision >
    < origin xyz= "0 0 0" rpy= "0 0 0" />
    < geometry >
      < box size= "0.01 0.01 0.01" />
    </ geometry >
  </ collision >

  < visual >
    < origin xyz= "0 0 0" rpy= "0 0 0" />
    < geometry >
      < mesh filename= "/home/user/catkin_ttu_ws/src/hector_models/hector_sensors_description/meshes/sonar_sensor/max_sonar_ez4.dae" />
    </ geometry >
  </ visual >

  < inertial >
    < mass value= "0.001" />
    < origin xyz= "0 0 0" rpy= "0 0 0" />
    < inertia ixx= "0.000000017" ixy= "0" ixz= "0" iyy= "0.000000017" iyz= "0" izz= "0.000000017" />
  </ inertial >
</ link >
```

Fig. 5. Sonar scanner parameters

The robot is executing its task. After an object was put into the range of the sonar sensor in the simulation software – the robot stopped executing the task and automatically proceeded with work, when the object was removed (See Fig. 6)

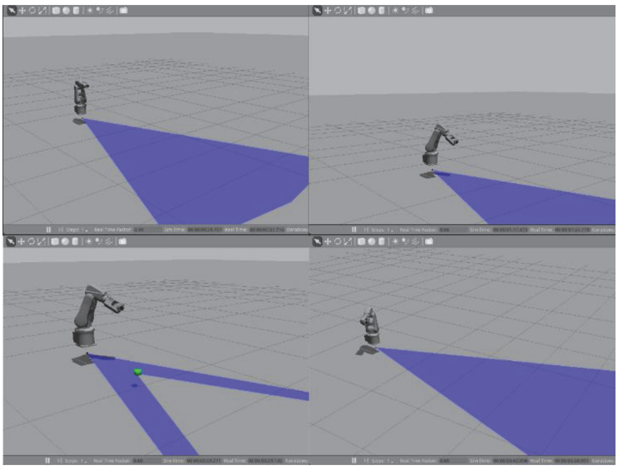


Fig. 6. Case study with safety system

3.3 Comparison of different methods

This experiment, comparing OLP and online monitoring methods, has shown that the side control of the robotic path leads to more effective results. While executing the path, additional factors can be added and in most cases, a pre-done program is not needed – according to the commands from the sensors, the robot path execution can be generated online.

This method makes industrial robots more intelligent and gives them the ability to decide about one or another task by themselves. Still, many factors must be considered like movement kinematics and maximum speeds of the robot joints. Considering this and implementing an online monitoring method into manufacturing process will give an unlimited amount of additional features to the manufacturing line. Moreover, first steps to create a tool for online programming in Virtual Reality (VR) were done with the development of real and virtual manufacturing line communication architecture [16], where methods described in this research can be tested.

4. Discussion

OLP and online monitoring and control methods were compared and discussed in order to find an optimal way for robot re-programming. Proposed solution based on ROS enables pre-check unattended human intervention and stop the robot cell work until danger of harming the human is de-activated. The methodology was designed for ABB small sized IRB series industrial robot, but the due openness of ROS it can be used also in other manufacturers' industrial robots. With online monitoring of the industrial robot process, there is no limit for the features of the robotic system. The method described in this paper and the case study is only the beginning of the future development of this topic.

Next steps are:

- Control of different manufacturers' robots with one tool
- Simulation improvements (reducing response time and increasing flexibility)
- Human presence simulation in the VR
- Flexible robot cell design using VR and simulations tools

In the future, this area should be further investigated and new tools for more precise simulations developed as the human precision simulation using VR tools.

5. Acknowledgements

The research was supported by I4MS project SmartIC Robotics - Regional Digital Innovation Hub in Robotics in Estonia. Authors are grateful to the Integrated Engineering students of Tallinn University of Technology – Mohammad Tavassolian, Aleksei Tanjuhhin, and Tengiz Pataraia helping in experiments and simulations.

6. References

[1] Vysocky A., Novak P. (2016) Human-Robot Collaboration in Industry. MM (Modern Machinery) Science Journal, June 2016

[2] Tsarouchia P., Makrisa S., MichalosaG., Matthaiakisa A.-S., Chatzigeorgioua X., Athanasatosa A., Stefosa M., Aivaliotisa P., Chryssolourisa G. (2015) ROS based coordination of human-robot cooperative assembly tasks - An industrial case study. Procedia CIRP 37, 2015, 254 – 259

[3] Polverini M.P., Zanchettin A.M., Rocco P. (2017) A computationally efficient safety assessment for collaborative robotics applications, Robotics and Computer-Integrated Manufacturing, 46, 2017, 25-37.

[4] Ore F., Vemula B.R., Hanson L., Wiktorsson M. (2016) Human – Industrial Robot Collaboration: Application of Simulation Software for Workstation Optimisation, Procedia CIRP, 44, 2016, 181-186.

[5] Tsarouchi P, Matthaiakis AS, Makris S, Chryssolouris G. (2016): On a human-robot collaboration in an assembly cell, International Journal of Computer Integrated Manufacturing

[6] Michalos G, Makris S, Spiliotopoulos J, Misios I, Tsarouchi P, Chryssolouris G. (2014) ROBO-PARTNER: Seamless Human-Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future, Procedia CIRP 23 (2014) 71-76

[7] Cherubini A, Passama R, Crosnier A, Lasnier A, Fraisse P. (2016) Collaborative manufacturing with physical human-robot interaction, Robotics and Computer-Integrated Manufacturing 40 (2016) 1-13

[8] Bdiwi M. (2014) Integrated sensors system for human safety during cooperating with industrial robots for handing-over and assembling task, Procedia CIRP 23 (2014) 65 – 70

[9] ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots (2016)

[10] Katsurin, A. (2016). Planning Trajectory of the Mobile Robot with a Camera, Proceedings of the 26th DAAAM International Symposium, pp.0407-0416, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria, DOI:10.2507/26th.daaam.proceedings.055

[11] Andreev, V.; Pletenev, P.; Poduraev, Y. (2016). The Method of Network Control of Robotic Systems of Different Models and Manufacturers, Proceedings of the 26th DAAAM International Symposium, pp.0727-0730, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734- 07-5, ISSN 1726-9679, Vienna, Austria, DOI:10.2507/26th.daaam.proceedings.101

[12] Khassanov, A.; Krupenkin, A.; Borgul, A. (2014). Control of the Mobile Robots with ROS in Robotics Courses, Proceedings of the 25th DAAAM International Symposium, pp.1475-1484, B. Katalinic (Ed.), Published by DAAAM International, Published by Elsevier in Procedia Engineering 100 (2015) 1475 – 1484.

[13] Gryaznov, N.; Lopota, A. (2014). Computer Vision for Mobile On-Ground Robotics, Proceedings of the 25th DAAAM International Symposium, pp.1376-1380, B. Katalinic (Ed.), Published by DAAAM International, Published by Elsevier in Procedia Engineering 100 (2015) 1376 – 1380.

[14] Kuts, V.; Tähemaa, T.; Otto, T.; Sarkans, M.; Lend, H. (2016). Robot manipulator usage for measurement in production areas. Journal of the Machine Engineering, 16 (1), 57−67.

[15] ROS documentation, Available from: http://wiki.ros.org/ Accessed: 2016-11-25 25.11.2016)

[16] Kuts, V.; Modoni, G. E.; Terkaj, W.; Tähemaa, T.; Sacco, M.; Otto, T. (2017). Exploiting factory telemetry to support Virtual Reality simulation in robotics cell. Augmented Reality, Virtual Reality, and Computer Graphics, 1: 4th International Conference, AVR 2017, Ugento, Italy, June 12-15, 2017. Ed. L. Tommaso De Paolis, P. Bourdot, A. Mongelli. Springer, 212−221. (Lecture Notes in Computer Science; 10324).10.1007/978-3-319-60922-5_16.

Publication III

Kuts, V.; Modoni, G. E.; Terkaj, W.; Tähemaa, T.; Sacco, M.; Otto, T. (2017). Exploiting factory telemetry to support Virtual Reality simulation in robotics cell. Augmented Reality, Virtual Reality, and Computer Graphics, 1: 4th International Conference, AVR 2017, Ugento, Italy, June 12-15, 2017. L. Tommaso De Paolis, P. Bourdot, A. Mongelli (Ed.). Springer, 212−221. (Lecture Notes in Computer Science; 10324).

Exploiting Factory Telemetry to Support Virtual Reality Simulation in Robotics Cell

Vladimir Kuts[1(✉)], Gianfranco E. Modoni[2], Walter Terkaj[3], Toivo Tähemaa[1], Marco Sacco[3], and Tauno Otto[1]

[1] Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, Estonia
vladimir.kuts@ttu.ee

[2] Institute of Industrial Technologies and Automation, National Research Council, Bari, Italy
gianfranco.modoni@itia.cnr.it

[3] Institute of Industrial Technologies and Automation, National Research Council, Milan, Italy
walter.terkaj@itia.cnr.it

Abstract. Significant efforts of the current manufacturing companies are devoted to the implementation of the full synchronization between the real world at the shop-floor level and its digital counterpart (so-called Digital Twin). Indeed, a true reflection of the real factory can be exploited to monitor and simulate the factory performance, allowing to adjust and optimize processes, anticipate failures and also investigate problems. One of the major challenge to be tackled in order to realize the Digital Twin is the handling of the factory telemetry, which can track the evolution of the objects in the real world. This paper investigates the potential of an application for supporting and handling the factory telemetry, thus allowing to create a snapshot of the real system that can dynamically augment and enhance the data-driven simulation applications supporting the manufacturing execution phase. As a proof of concept of the architecture, a prototype has been developed in the field of robotics. In such context, the proposed architecture is on the basis of a Virtual Reality tool to simulate human presence for development of safety systems in robotic cells.

Keywords: Digital Twin · Factory telemetry · Robotic cell · Virtual Reality

1 Introduction

The concept of Digital Twin (DT) is creating and maintaining a digital representation of the real world of the factory and supporting its management and reconfiguration by the means of optimization and simulation tools, which are fed with real and updated factory data. This concept is not new as it was first used by NASA research in 1957, when the satellite Vanguard was sent into orbit [1]. More than half a century later, recent advances in ICT are offering new opportunities to fully exploit the potential of the DT in the manufacturing field. Such a potential has been recently analyzed in many articles and publications [2, 3]. Specifically, a DT based approach is described in [4] to enable a new strategy for taming organization complexity and simulating various

L.T. De Paolis et al. (Eds.): AVR 2017, Part I, LNCS 10324, pp. 212–221, 2017.
DOI: 10.1007/978-3-319-60922-5_16

outcomes across the whole product lifecycle. In [5], the authors leveraged the digital model of the factory provided by the DT to conceive a new generation of autonomous manufacturing systems, that can execute high-level tasks without detailed programming and without human control. Another research work proposed a model for enhancing data exchange between different systems included in a Cyber Physical System and that are connected with the DT [6].

An effective DT implementation asks for:

- a data model to represent the evolution of the objects by integrating streams of data coming from different sources, e.g. monitoring systems (i.e. a real history), production-planning methods (i.e. a planned history), performance evaluation tools (i.e. a simulated history) [7];
- the acquisition and validation of data via an appropriate monitoring system to enable the synchronization between the real and digital factory [8]. This paper will focus on this aspect in particular;
- setting the level of detail (LoD) of the DT in order to make it effective and efficient while meeting specific goals.

One the technologies that can benefit from the DT is Virtual Reality (VR), which provides a virtual and realistic view of the environment where the flow of real-time and historical data is integrated with the human presence. In particular, if VR is integrated and connected with the DT, it can be exploited to:

- evaluate possible system reconfigurations via simulation (passive mode);
- remotely control the system (active mode).

The aim of this research project is investigating the potential of an approach to connect VR applications with the DT in the specific domain of industrial robotics and collaborative robots. All these enabling technologies are particularly relevant in the modern factories, as their inclusion in the scope of Industry 4.0 [9] can demonstrate. Equally important it is their integration. In this regard, the herein introduced approach combines VR with an enabler of the DT, the Factory Telemetry (FT), i.e. the data acquired by means of sensors distributed across the plant and then exploited by specific applications to monitor ongoing processes [10].

FT can be consumed by the VR applications under the form of a real time stream of data or as historical data. Real-time telemetry allows to acquire live data from the properly configured components (e.g. sensors, etc.) distributed across the whole factory network. In the scope of the DT, it allows to combine and overlap real and virtual processes that can be executed at the same time, thus giving an amount of additional functions for monitoring, analysis, learning. On the other hand, historical telemetry gives the opportunity to save data related to the real process execution and use it in order to simulate new configurations (e.g. new layouts) of the manufacturing line.

Usage of FT can bring several benefits during various stages of the factory lifecycle. In the field of robotics, it can dynamically augment and enhance the data-driven simulation applications supporting engineers in the development and programming of robotic cells. The novelty of the approach consists of the combination of the telemetry (historical and live) and the VR simulation tools, since a current state of the art shows a lack of solutions for synchronizing the digital and real environments [10]. Furthermore, a hardware and software configuration integrated into a whole "ready to use" system is designed to test the connection of the real and digital environments. Finally, with reference to this configuration, a case study is developed to show how the collaboration between two different geographic locations can be enabled by sharing the DT model. The remainder of this paper is structured as follows. Section 2 presents the benefits of an approach based on the DT, while Sect. 3 envisions an application supporting DT, eliciting their requirements. Section 4 illustrates the conducted experiments. Finally, Sect. 5 draws the conclusions, summarizing the major findings.

2 The DT Approach

One goal of this study is creating a digital clone of a system so that it is possible to remotely monitor and control the real existing facilities via its virtual twin. Regarding this, the synchronization between the real and virtual worlds can contribute to its implementation. Making a digital copy of an existing real manufacturing system or subsystem (e.g. a robotic cell) gives the ability to control both the real and virtual environment with the same system controller, thus allowing to apply corrective decisions to the real system based on the information the virtual system receives and analyses. As seen in Fig. 1, the system controller gathers all data from sensors and sends them to the virtual system in order to synchronize digital and real worlds. According to received information, virtual system processes data also based on some simulation application and then gives the feedback to main controller, which in turn sends commands to the real system, thus implementing the closed loop between the virtual and real factory.

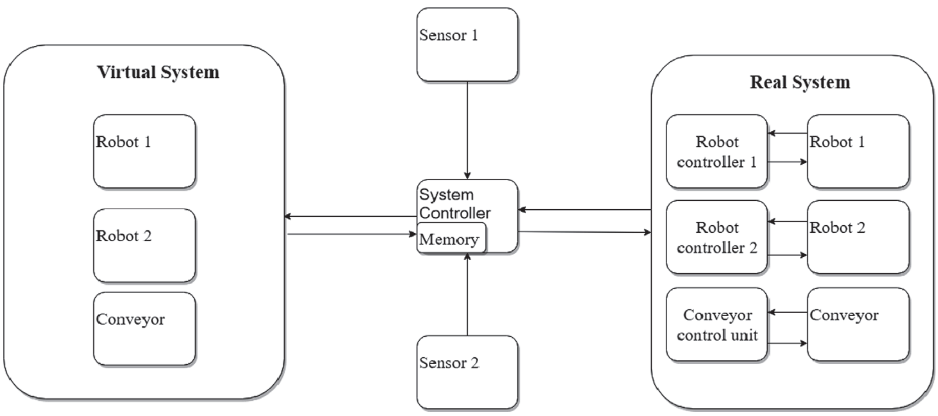


Fig. 1. Virtual and real world synchronization

Leveraging the digital clone, it is possible to support the simulations of different system configurations (see Fig. 2), thus helping optimize the production processes. A snapshot of the system can be taken by elaborating the flow of real data. Starting from the snapshot of the system, one or more simulations can be run to generate and playback a flow of simulated data. After the analysis of simulated data, reconfigurations or planning decision can be taken, creating a flow of data from the virtual world to

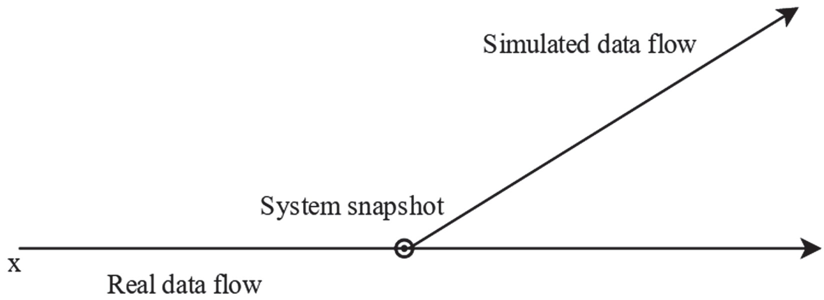


Fig. 2. Data flow – real and historical cut-off (time is on the x-axis)

reality. Historical data can be used in the virtual simulations to check how the real system can be better optimized or reprogrammed from the exact step of the process flow. This way, the idle time of the robot or other manufacturing system components can be reduced while they are being optimized or re-programmed. Moreover, the historical data can be used as an input of prediction tools, in order to foresee the behavior of robots, machines, and other equipment of a manufacturing context.

Another interesting aspect of the DT is that it enables an efficient interaction between smart objects and its surrounding environment, thus making the overall system smarter. For example, a robot, acting as a consumer of telemetry, can get data from sensors and according to this information may change its behavior. If a human being enters a dangerous zone, then the robot will stop and generate a new path around or proceed with the next task, exploiting a simulation of the safety system. Moreover, the co-work between multiple robots or components in the system can be improved. For example, one robot could ask for the position of another robot and based on this information the path of a second robot can be generated. This can be particularly relevant if different types and brands of robots are involved (e.g. ABB, Universal Robot, Kuka, Fanuc, MIR or others).

The DT cannot be realized without the development of a digital model of the environment, which must be as precise and detailed as its real twin in order to execute accurate simulations and evaluations [7]. Under these conditions, it is essential to represent the characteristics, behavior and relations of the system components like operators, products, resources, transporters, sensors. For instance, the kinematics of a robot must be properly modeled. However, the digitalization of the system and its components can be fully exploited only if a proper software application is able to handle the telemetry data, integrating the latter with other digital tools. The next section delves into the main characteristics of such a telemetry-based application.

3 Overview of the Envisioned Application Supporting FT

The approach introduced in this section is agnostic to the application domain but from this point onward we'll focus on the case study of a robotic cell, which helps to elicit the requirements of the analyzed application. Behind this case study, there is the need to enhance the DT through a system of synchronization similar to the one in Fig. 1.

3.1 The Requirements

The following requirements must be satisfied by a telemetry application in order to support the synchronization between the real system and its DT.

Firstly, it must enable the communication between a real industrial robot controller and an external controller in order to monitor the robot presence. It must also be able to acquire data from the sensors (connection to the real world) and then process these data in order to control robot by giving it feedback in real-time. For example, if any object enters the dangerous zone in robotic cell, the controller re-calculates the robot's movement and gives a command to modify its task in order to avoid a collision [11]. Also, in order to enable the communication between a generic virtual system and the external controller, proper protocols and command signals should be used.

Finally, a data storage should be used to collect all data mentioned above, also under the form of Big Data, in order to save historical points of the manufacturing line working process. Regarding this, in order to take the snapshot of the current robot's task and path data, it is needed to take into consideration different static and dynamic robot conditions (see Fig. 3). Moreover, all the handled data must be expressed in the same format in order to guarantee the digital continuity and support the semantic interoperability [12, 13].

Saved historical data	
Static	- Static object coordinates (tables, walls) - moment position coordinates of each movable part (each robot arm)
Dynamic	- Speed of the movable parts of the system - robot arm axis - External factors - operator, loader route - Movement vector of the movable parts - manufacture device program step, when moment saved

Fig. 3. Static/dynamic data for enabling historical data flow

3.2 The Implementation

An implementation of the envisioned application should provide two complementary ways to handle the telemetry data:

One is real time and exploits a binary protocol in order to be optimized for high performance. This solution allows various connected and enabled devices to receive real time data stream from the sensors. A specific study will investigate which is the best protocol and libraries to be used in order to support such as solution. In this regard, the UDP [14] protocol is faster than TCP [14], but UDP is unreliable without any acknowledgment, while TCP is a reliable three-way handshaking protocol. For streaming data, the UDP protocol is more suitable, as it represents a trade-off between speed and reliability.

Moreover, a Multicast transmission of the data over the factory network allows the software application (telemetry consumers) to receive the telemetry regardless of the PC where the software application runs, as long as it is connected to the network and enabled [15].

The second is Web service-oriented and allows to persist the data on the database. Since this solution does not have to support real time process, we could handle more data (both in Volume and Velocity) than through the solution 1. Under these conditions, we could say that solution 2 allows to visualize more data than solution 1 since it can support higher rates or even a higher number of sensors, which are instead limited due within solution 1 due to the real time requirements

Both the solutions provide proper connectors towards the telemetry consumers (e.g. simulator packages, VR tools and robots). A proper API (Application Programming Interface) should guarantee the access to telemetry data, enabling data analysis in external tools. In this regard, simulator packages, VR tools and robots can act as telemetry consumers.

Figure 4 shows the three steps through which data flow from the real system to the virtual system and back:

1. Data flow from real to virtual system;
2. Persistence of the real-time data into the data storage;
3. Integration of the optimized data in the virtual system, which was based on the historical data.

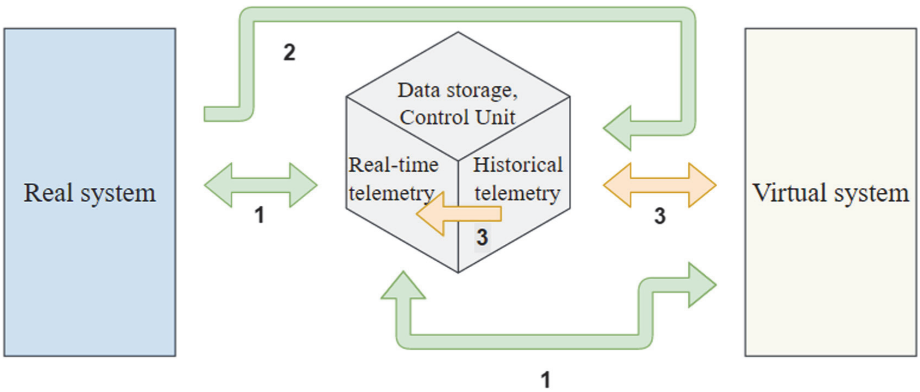


Fig. 4. Data flows between virtual and real system

A valid starting point for the implementation of both the options is represented by the OPC Unified Architecture [16], which in its transport layer defines two different mechanisms for data communication (factory telemetry): a TCP protocol for communication at high performance and a protocol based on the Web Services for communication Internet firewall-friendly. It will be also essential to exploit the new generation of storage systems which can run on distributed eventually cloud-based cluster system (e.g. Hadoop, Database sharding, NoSQL databases, etc.).

4 The Experiment: VR – Human Presence Simulation

The case of a measurement robot cell has been considered to test the ongoing developments related to the approach presented in the previous sections. The DT in robotics field can be applied within different use cases, bringing several benefits such as the following:

- decrease of safety errors and mishaps;
- improvements in run-time efficiency;
- improved accurateness of traceability;
- compression of lead time in production.

One significant use case is to foresee the response of a robot to human interaction and uncover previously unknown issues before they become critical in the real environment by comparing predicted and current responses.

In the presented experiment, the real system (see Fig. 5) consists of one industrial robot with attached 3D laser scanner probe and camera. The machine vision camera is used for detection of movement and shape of the static and dynamic objects in the cell. A robot controller is also a part of the robotics cell. Moreover, the robotic cell includes two laser sensors, which are rotating on 360° and map the room on the exact set up range, so they can detect changes in the environment and give the signals to the controller, when interferences are detected. A common table is used to place object for the inspection [17], as the main purpose of the presented robotic cell is to perform 3D scanning on site. The presented experiment is ongoing and involves two different geographical locations. The robot cell is located in Tallinn University of Technology Lab in Estonia, while VR environment is being developed at the Virtual Reality Lab of the Institute of Industrial Technologies and Automation in Italy.

Fig. 5. Robot cell in real world

The digital model of the robotic cell was developed using Unity software tool (Fig. 6). The main components of the real environment were reproduced in the digital model while considering a proper scale of the room and of the related equipment. In order to be able to run experiments which involve human/object presence simulations and to not interrupt the real system sensors work, perception components are simulated as a triggers in the virtual environment and configured as their real counterparts. Input for the movement of the camera is gathered from the keyboard, joysticks or from the VR headsets with the usage of the generic avatar, which have the solid mesh for simulation of the person presence in the digital world.

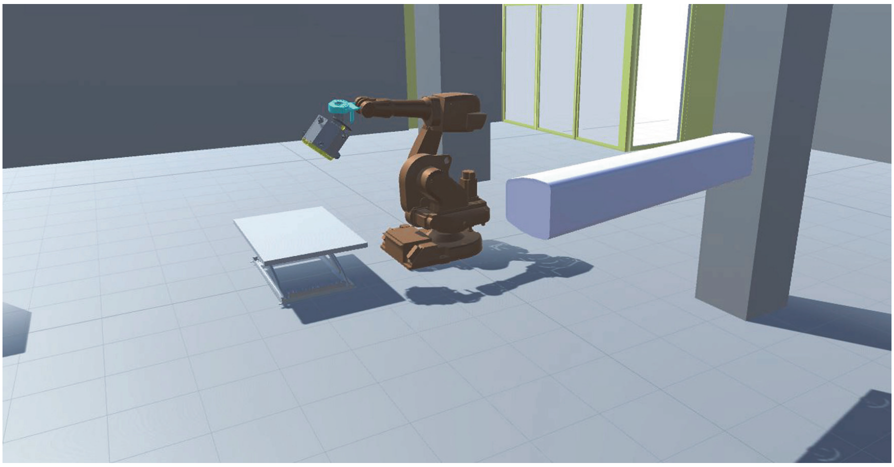


Fig. 6. Robot cell in virtual world

In order to enable an accurate simulation process, the movement of the robot and presence of the objects are modeled in the virtual. Regarding the data feedback, axis speed and placement of the robot in the real room are automatically sent by the external software Robot Operating system (ROS) [18] which is integrated into the external controller. Moreover, the ROS integration is important in order to use historical data for furthers simulations in the virtual environment and to send the feedback to the real system after processing the data.

4.1 Two Preliminar Experiments with VR Tools in the Robotic Cell

The virtual system developed in Unity can be used to run the two following experiment.

Human Presence Simulation to Test the Safety System in the Robotic Cell. The ISO/TS 15066:2016 [19] standard regulates the presence of human beings in a robot cell while it is working. The safety system in robotic cell, described above, can be tested in the VR environment by means of an avatar. In this way it can be avoided that an operator enters a dangerous zone for validation purposes. Sensors simulated in the virtual system react onto the avatar, which is being controlled by the operator with VR headset and give feedback to the external controller.

Moreover, the interaction with the historical data can be evaluated. Indeed, avoiding to interrupt the real process but using the persisted historical data, it is possible to interact with all simulated manufacturing line aspects and see how the system reacts. For example, it can be exploited to evaluate how operator takes raw material away from the conveyor and see how the system equipment reacts. Overall, this approach gives many options for simulations of the real processes without any risk to damage human health or expensive equipment, while giving an educational value for the new operators and students, who can test every aspect of the system processes and faults reactions without affecting the real environment.

Remote Online Monitoring of the Robot Cell. Leveraging the VR tools, the robotic system can be accessed remotely from any geographical point, giving the control over the processes. Operators exploiting clothes/glasses/tools RFID sensors can move around the robot cell with their ordinary daily routine. Data from the sensors are transferred towards the VR environment. In order to have an update about the presence information of the operators, machine vision cameras and laser scanners are also being used.

5 Conclusion

This paper has introduced an approach to combine three different enabling technologies of the Industry 4.0 (i.e. VR, FT and robotics). The approach is expected to offer several benefits regarding the quality of monitoring and simulation process in the manufacturing. Some preliminary experiments have been conducted in the specific domain of industrial robotics and collaborative robots. These experiments are ongoing and involve two different labs in Estonia and in Italy. However, the presented work is only a first step of a larger research agenda aiming at realizing a full synchronization between the real world of the plant and its virtual clone.

Future developments will address two main goals. First of all, it will be essential to evaluate the quality of service in terms of scalability and performance of the solution proposed, in order to verify its capability to support situations of intensive Big data. The second goal will regard the evaluation of the approach in various domains different from robotics.

Acknowledgments. The research was supported by I4MS project SmartIC Robotics - Regional Digital Innovation Hub in Robotics in Estonia.

Authors are grateful to the Integrated Engineering students of Tallinn University of Technology – Vladislav Minakov, Mohammad Tavassolian, Aleksei Tanjuhhin and Tengiz Pataraia helping in experiments and simulations.

The research was partially funded by the project "Smart Manufacturing 2020" within the "Cluster Tecnologico Nazionale Fabbrica Intelligente".

References

1. Computers in Spaceflight: The NASA Experience. http://history.nasa.gov/computers/Ch8-2.html. Accessed 23 Feb 2017
2. Grieves, M.: Digital twin: manufacturing excellence through virtual factory replication (2014). http://www.apriso.com/library/. Accessed 23 Feb 2017
3. Fritz, R., Kohlhoff, S.: Integration of Virtual and Physical Production Connection of Engineering Design with the Shop Floor (2015). Accessed 23 Feb 2017
4. Grieves, M., Vickers, J.: Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen, F.-J., Flumerfelt, S., Alves, A. (eds.) Transdisciplinary Perspectives on Complex Systems, pp. 85–113. Springer, Cham (2017). doi:10.1007/978-3-319-38756-7_4
5. Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.: About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine **48**(3), 567–572 (2015). ISSN 2405-8963. http://dx.doi.org/10.1016/j.ifacol.2015.06.141
6. Schroeder, G.N., Steinmetz, C., Pereira, C.E., Espindola, D.B.: Digital twin data modeling with automationML and a communication methodology for data exchange. IFAC-PapersOnLine **49**(30), 12–17 (2016). ISSN 2405-8963. http://dx.doi.org/10.1016/j.ifacol.2016.11.115
7. Terkaj, W., Tolio, T., Urgo, M.: A virtual factory approach for in situ simulation to support production and maintenance planning. CIRP Ann.- Manuf. Technol. **64**(1), 451–454 (2015)
8. Kádár, B., Lengyel, A., Monostori, L., Suginishi, Y., Pfeiffer, A., Nonaka, Y.: Enhanced control of complex production structures by tight coupling of the digital and the physical worlds. CIRP Ann. – Manuf. Technol. **59**(1), 437–440 (2010)
9. Brettel, M., Friederichsen, N., Keller, M., Rosenberg, M.: How virtualization, decentralization and network building change the manufacturing landscape an industry 4.0 perspective. Int. J. Mech. Ind. Sci. Eng. **8**(1), 37–44 (2014)
10. Modoni, G.E., Sacco, M., Terkaj, W.: A telemetry-driven approach to simulate data-intensive manufacturing processes. Procedia CIRP **57**, 281–285 (2016)
11. Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraisse, P.: Collaborative manufacturing with physical human–robot interaction. Robot. Comput.-Integr. Manuf. **40**, 1–13 (2016)
12. Modoni, G.E., Doukas, M., Terkaj, W., Sacco, M., Mourtzis, D.: Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models. Int. J. Comput. Integr. Manuf. (2016). Taylor & Francis
13. Capozzi, F., Lorizzo, V., Modoni, G., Sacco, M.: Lightweight augmented reality tools for lean procedures in future factories. In: De Paolis, L.T., Mongelli, A. (eds.) AVR 2014. LNCS, vol. 8853, pp. 232–246. Springer, Cham (2014). doi:10.1007/978-3-319-13969-2_18
14. Forouzan, B.A.: TCP/IP Protocol Suite. McGraw-Hill, Inc., New York City (2002)
15. Modoni, G.E., Veniero, M., Sacco, M.: Semantic knowledge management and integration services for AAL. In: Cavallo, F., Marletta, V., Monteriù, A., Siciliano, P. (eds.) ForItAAL 2016. LNEE, vol. 426, pp. 287–299. Springer, Cham (2017). doi:10.1007/978-3-319-54283-6_22
16. Mahnke, W., Leitner, S.H., Damm, M.: OPC Unified Architecture. Springer Science & Business Media, Heidelberg (2009)
17. Kuts, V., Tähemaa, T., Otto, T., Sarkans, M., Lend, H.: Robot manipulator usage for measurement in production areas. J. Mach. Eng. **16**(1), 57–67 (2016)
18. ROS documentation. http://wiki.ros.org/. Accessed 7 Mar 2017
19. ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots

Publication IV

Kuts, V.; Otto, T.; Tähemaa, T.; Bukhari, K.; Pataraia, T. (2018). Adaptive industrial robots using machine vision. Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition: IMECE2018, November 9-15, 2018, Pittsburgh, PA, USA. ASME.

Proceedings of the ASME 2018
International Mechanical Engineering Congress and Exposition
IMECE2018
November 9-15, 2018, Pittsburgh, PA, USA

IMECE2018-86720

ADAPTIVE INDUSTRIAL ROBOTS USING MACHINE VISION

Vladimir Kuts
Tallinn University of Technology
Tallinn, Estonia

Tauno Otto
Tallinn University of Technology
Tallinn, Estonia

Toivo Tähemaa
Tallinn University of Technology
Tallinn, Estonia

Khuldoon Bukhari
Tallinn University of Technology
Tallinn, Estonia

Tengiz Pataraia
Tallinn University of Technology
Tallinn, Estonia

ABSTRACT

The use of industrial robots in modern manufacturing scenarios is a rising trend in the engineering industry. Currently, industrial robots are able to perform pre-programmed tasks very efficiently irrespective of time and complexity. However, often robots encounter unknown scenarios and to solve those, they need to cooperate with humans, leading to unnecessary downtime of the machine and the need for human intervention. The main aim of this study is to propose a method to develop adaptive industrial robots using Machine Learning (ML)/ Machine Vision (MV) tools. The proposed method aims to reduce the effort of re-programming and enable self-learning in industrial robots. The elaborated online programming method can lead to fully automated industrial robotic cells in accordance with the human-robot collaboration standard and provide multiple usage options of this approach in the manufacturing industry. Machine Vision (MV) tools used for online programming allow industrial robots to make autonomous decisions during sorting or assembling operations based on the color and/or shape of the test object. The test setup consisted of an industrial robot cell, cameras and LIDAR connected to MATLAB through a Robot Operation System (ROS). The online programming tests and simulations were performed using Virtual/Augmented Reality (VR/AR) toolkits

NOMENCLATURE

AGV – Automatic Guided Vehicle
AIA – Automated Imaging Association
AR – Augmented Reality
DT – Digital Twin
DWENN – Dynamic Wave Expansion Neural Network
HRC - Human - Robot Collaboration
LIDAR - Light Detection and Ranging
MDP – Markov Decision Process
ML – Machine Learning
MR – Mixed Reality
MV – Machine Vision
RFID - Radio Frequency Identification
ROS – Robot Operation System
URDF - Unified Robot Description Format
VR – Virtual Reality
XML - Extensible Markup Language

INTRODUCTION

Industrial robots are widely used in different manufacturing processes [1]. However, problems appear due to insufficient flexibility and unpredictable human factors, which lead to failures and increases in the downtime of the machine.

Human-robot collaboration (HRC) with conventional and co-bot type robots have been researched with various methodologies [2, 3, 4, 5, 6] considering the ISO/TS 15066:2016 standard. The main approach was to make robots more intelligent and safe by adding additional perception layers, such as different types of sensors: Light Detection and Ranging sensors (LIDARs), sonar, radio frequency identification devices (RFID), etc. In addition to sensing, web and stereo cameras like Kinect have been used for the visual recognition of objects or movements. All of these tools have been used to give robots the ability to map their surroundings and detect movement or changes in it. This data can be stored locally or in cloud storage [7], or post-processed with hybrid approaches of different methods' usage [8]. To enable data gathering from individual or multi-robot cells and sensors for use in the development of control algorithms, middleware software Robot Operating System (ROS) [9, 10, 11] together with behavior model analysis tools [12] and the control software MATLAB have been used. These components together form a toolkit of Machine Learning (ML). In addition, the implementation of the ML toolkit with Virtual and Augmented Reality (VR and AR) tools for the testing of visual and sensor data gathering are able to solve HRC issues and increase robot cell flexibility.

Machine Vision (MV) has proven its ability to observe objects in industrial automation solutions since 1984 when the Automated Imaging Association (AIA) was established. It has been implemented in different areas such as robot surgery [13], insurance [14] and robots' localization in manufacturing [15].

A boost to ML technologies was given by autonomous vehicle development at the end of the last century. Industrial robots face almost the same problems known from unmanned ground vehicles but those driverless cars have much more uncertainty in their work. Tilbury and Ulsoy [16] have developed a set of algorithms, based on Markov Decision Process (MDP) models, which the vehicle can use to decide when to operate autonomously versus when to request more information from the operator. A similar situation is the core problem in Starship Technologies' self-driving delivery robots developed and tested in Estonia. All the aforementioned autonomous machines do not have the ability to recognize and predict the manners of unknown objects detected by an MV device.

In human-machine relations, both with industrial and collaborative robots, a simplified control process can decide how to act without any databases. The decision is made based on two main parameters:

- How large is the unknown object in the robot's work area (to avoid objects like a butterfly or bird)?; and
- Is it approaching the robot's work area (to avoid still objects placed there by humans)?

These functions are already mature for use in many MV systems such as Cognex's in-sight explorer software.

Modern MV technologies can easily measure objects, compare them with the contents of their databases, and act according to the program if the objects fit the models found in the databases.

The main aim of this study is to make robots more intelligent by adding perception, ML and online programming tools as well as using the approaches developed for AGVs in the automotive industry. This adaptive for robot approach will eliminate human-factor mistakes from the robots' decision-making tasks, and to develop a methodology for generating smart learning algorithms of the industrial robot for real manufacturing processes. This main research scenario task for intelligent self-learning industrial robots is to sort different shape and color objects by itself, without any previous offline programming. As a result, these methods will enable and transfer a control of ROS-based autonomous vehicles and decision-making systems onto industrial robots.

AN APPROACH TOWARDS INTELLIGENT INDUSTRIAL ROBOTS

Industrial robots are the basic building blocks of the automated manufacturing systems of today because of their speed, accuracy, and precision. However, in the paradigm of "smart manufacturing", the robots will need to be intelligent as well. A contemporary palletizing robot is most likely pre-programmed to pick a work piece from a pre-defined location coordinate [x, y, z] and place it on coordinate [x', y', z']. In a perfectly coordinated world, the pre-programmed setup would work without problems but real manufacturing environments are still far from the perfect Industry 4.0 dream. Thus, the strictness of a pre-programmed setup makes it incredibly difficult for humans or advanced industrial systems like Automated Guided Vehicles (AGVs) to collaborate with industrial robots. However, this problem can be solved by the introduction of a machine perception layer into the decision-making process of industrial robots. A perception layer makes it possible for robots to make autonomous decisions and develop a certain level of intelligence. In pursuance of intelligence in industrial robots, there are three key essential components:

- object recognition,
- object localization, and
- path planning.

Object recognition allows a robot to perceive objects of interest by taking advantage of the breakthroughs in the field of MV. In proper conditions, objects can be recognized based on their shape and color with a success rate of over 99% [17]. This allows robot cells equipped with cameras to identify work objects and make decisions for grasping orientation based on geometry. In addition, it enables robots to recognize humans and other obstacles, resulting in a safer environment. However, the technology has its limitations when it comes to locating recognized objects. In addition, object recognition by itself is not helpful when it comes to handling the objects. Therefore, object localization techniques are required that can provide location information of the identified objects in 3-dimentional space. LIDAR is such a device that can generate precise location data of its surroundings in real time, and by calibrating the camera and LIDAR together, the location of the identified object can be determined [18]. With this knowledge of the 3-dimentional location of the identified objects, the robot can be instructed to reach the object by calculating the joint angles using inverse kinematics [19]. However, inverse kinematics only calculates the different possible joint angles to reach a specific coordinate. A safe path or trajectory for reaching the desired coordinate can only be generated by using path-planning algorithms.

The aim of the path planner is to find the shortest geometric path from the initial position to the goal position avoiding any obstacles on the way. Although this branch of robotic manipulators has been studied for over three decades, advancement towards truly autonomous planning was only introduced in 2005 by the introduction of dynamic wave expansion neural networks (DWENN), having the ability to generate dynamic distance potentials for real-time path planning in a time-varying environment [20]. Given the desired final location, DWENN can be used to generate a collision free trajectory to the desired location with a set orientation.

A use case for pick and place robots

When considering an exemplary scenario where a pick and place robot is supposed to place red objects in a red box and the rest of the objects in a white box, then in the pre-programmed setups, the robot would always expect the red objects to arrive at a pre-defined location of coordinate [x y z], and rest of the objects to arrive at coordinate [x' y' z']. Although it is possible to place the objects as required by the robot, in a real dynamic environment it is incredibly inefficient and time consuming. However, the process would be much simpler if the object could be placed anywhere within the reach of the robot. The robot would execute the motion based on the instructions provided, whereas the instructions could be as simple as "place red objects in the red box and non-red objects in the white box".

Examining the three key components of intelligent industrial robots, it can be observed that all three given components (object recognition, object localization and path planning) are present. Any industrial robot can execute this task with a bare minimal instruction set. Figure 1 depicts the process of solving the tasks. The robot is activated upon identifying a work object, when the robot decides the action based on the instructions provided by the human operator. The location of the work piece object and its destination are then calculated by the object localization algorithms. Later, the path-planning algorithm decides the path of the gripper and then the robot takes the chosen path avoiding obstacles on the way. Finally, the object is placed on the desired location while the robot waits for the next object.

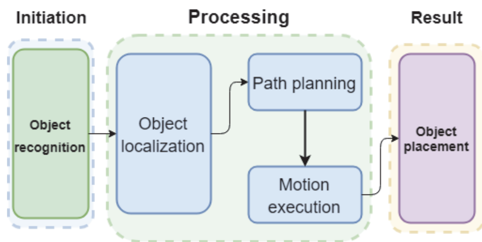


Figure 1. Steps for solving the smart path planning of a robot

SIMPLIFIED IMPLEMENTATION FOR THE USE CASE

The aim of the simplified implementation is to experiment with the proposed concept of the three key components of intelligent industrial robots for palletizing tasks. This also enables the use of the Digital Twins (DT) concept for visualizing and optimizing the experimental robot work cell. The experimental work cell consists of a six-axis industrial robot (Motoman GP8, see Fig. 3), two 2-D LIDAR (L1 and L2 in Fig. 2), and one stereo camera (Cam in Fig. 2).

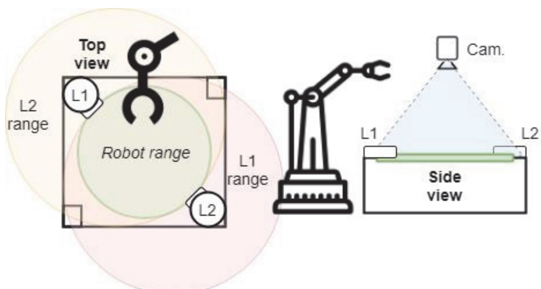


Figure 2. Sorting robot setup with additional perception tools

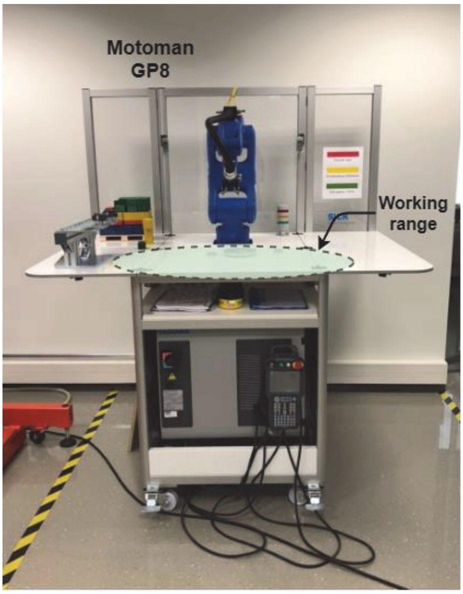


Figure 3. **Use-case robot work cell front view**

In addition, the work cell is equipped with two computers running Linux and Windows operating systems. The Linux computer contains ROS and its open source libraries, such as the camera-LIDAR calibration setup, Moveit, Rviz, Gazebo and BLAM for localization and mapping. The Windows machine, on the other hand, operates with Matlab and its toolboxes, such as the Robotic System Toolbox, Signal Processing Toolbox and Computer System Vision Toolbox. Moreover, the virtual reality headsets utilizing a VR engine (Unity3D) run on Windows. The robot and two computers are networked using TCP/IP protocols.

Tools such as Matlab and ROS make it easy to coordinate large amounts of data and keep order on all levels of communication. Algorithm-hardware connectivity on Matlab and ROS is set with the help of the Robotic System Toolbox, designed to provide inverse kinematic algorithms based on iterative gradient-based optimization for manipulator robots and serves as an interface between Matlab and ROS.

The implementation of the system takes place on several platforms and machines. In the proposed architecture (see Figure 4), ROS serves multiple purposes, where the primary use for ROS is as a data exchange library from Matlab to the rest of the hardware setup: the robot controller, LIDAR and camera. The secondary use of ROS is to visualize the camera, robot and sensor data in Rviz and in the VR headset, as well as to record the sensor data using a set of tools for recording from and playing back (rosbag) for later analysis and post processing.

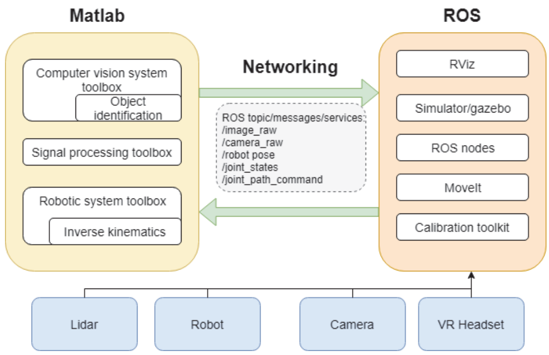


Figure 4. ROS system setup

Setting up the robot model in a workspace and calculating inverse kinematics and dynamics on Robotic system toolbox requires a Unified Robot Description Format (URDF) file, an Extensible Markup Language (XML) code containing all the information of the robot including location and the length of the joints as well as robot angle movement limitations. This file is also used to create a tree for a manipulator of the robot containing Rigid Body and Joint objects. URDF in our system is created with the SolidWorks add-in, which exports URDF from the robot's 3D model. In the workspace, all the calculations of the robot's inverse kinematics are performed on Matlab and later messages are published with specific message types through ROS to move the robot in Cartesian space.

IMPLEMENTATION OF IMMERSIVE TECHNOLOGIES

Immersive technologies such as Augmented, Virtual and Mixed Reality (AR, VR and MR) tools have been used in the manufacturing industry and robotics for a while as visualization tools. Different simulation and offline programming software (Visual Components, ABB RobotStudio, etc.) integrate plug-ins for visualization and interactions in VR. There are many areas of manufacturing and robotics where the usage of immersive technologies gives an advantage. Some examples are troubleshooting, monitoring, control, remote co-work, pre-evaluation of machine launch and education. Our Industrial Virtual and Augmented Reality lab (IVAR lab - http://ivar.ttu.ee) is working in the direction of connecting real manufacturing lines and virtual reality. Thus, the method described in this paper is being merged together with immersive technologies in a few different ways.

The above described experimental work cell (see Fig. 5) and a whole laboratory were digitalized on a real-life scale. For the test model, our own universal robot control software layer was developed, enabling control and synchronization between the real and virtual worlds (see Fig. 6) from one control unit. This connection approach, which will be described in more detail in another research paper, is being used in this paper as an example of the use-case tests described below.

Figure 5. Robotic work cell in Industrial Virtual and Augmented Reality laboratory for DT prototyping

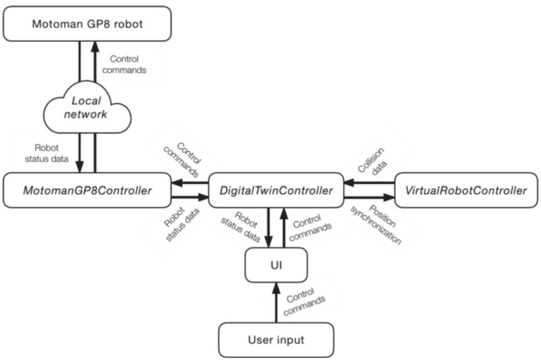


Figure 6. DT software model

Automation process visualization

When the MV of the robot is implemented and tested, other means of perception can be introduced in a secondary flow of the research to advance the work cell towards a higher level of automation. With the means of 3D Velodyne (VLP - 16) LIDAR, a camera and an Oculus Rift VR headset, it was possible to visualize all the data in VR and create a more dynamic environment as well as give us an immersive experience of a work process visualization (see Fig. 7).

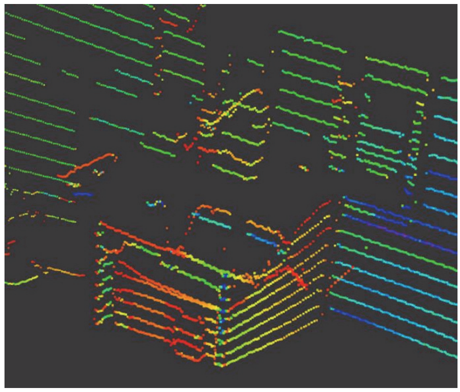


Figure 7. Scanned 3D image of the work cell

Bringing 3D LIDAR and its calibration with the camera into the work cell enabled us to track each position of the robot in 3D space as another layer of safety by detecting objects/humans in 3D space with a high level of precision. In this advanced scenario, each sensor camera and robot has predefined coordinates in 3D space and their data is visualized with the help of RViz. Our research indirectly benefits recent developments in autonomous vehicles to integrate some well-tested ROS packages such as 3D LIDAR and camera calibration, and could find a perfect fit in the manufacturing industry. The abovementioned calibration setup in our research acts as a monitoring system where the camera is used as an image detection tool for humans and the LIDAR locates the detected object in 3D space giving distance values from the object to the LIDAR itself. The concept of having one sensor and cameras for all the machines and having the precision of the tracking system elements down to centimeters will introduce a new dynamic and open up another door for future research.

Technology for machine learning

In this research, the methodology described and practical implementation example can be used together with VR and AR technologies as visual learning tools. These are already being used in wider research topics with robotic cells [21, 22, 23] scenarios. Our aim of integrating these tools is to enable ML with virtual objects. The AR sorting task from the scenario can be done with holographic and augmented objects (see Fig 8) in order to teach a robot how to pick up and sort them correctly. A real industrial robot executes the tasks and learns in order to use these learning experiences on real objects in the future. To enable this method, the synchronization of the visual application in Unity3D together with real robots should be performed. For this purpose, the kinematics data, mathematical models and coordinates must be the same. In addition, just having the virtual models of the object for teaching is not enough. To synchronize two flows together, an AR model of the copied robot (i.e., DT data) should also include all movements and gripper grabs. Feedback data flow from augmented sorting algorithms can be used to teach real robots to execute the task without the real use of the work piece objects. Learning time is saved by the lack of a need to put objects back on the table.

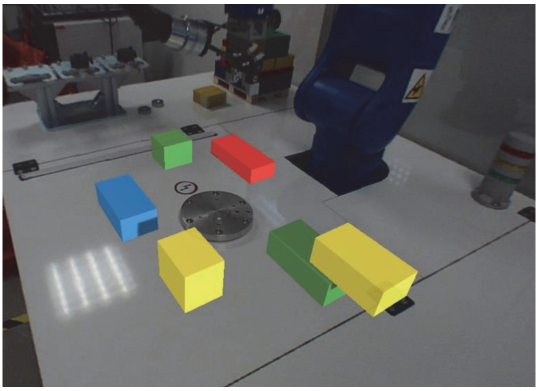


Figure 8. Work cell for AR implementation

Another developed method is using DT telemetry [21] between a real and virtual robotic cell, gather the data and use it to teach a robot in the digital world (see Fig. 9). Similar to the AR approach, preliminary work should be performed to enable machine learning in VR. The DT method was tested on a real system. A DT manufacturing line or robotic cell should be digitalized with consideration of the following points:

• a data model of every component in the system, production process and program is generated,
• data flow and format is kept the same between the synchronized objects; whereas, real and virtual flows should both be in the same format and controlled from a single control unit,
• optimized data with a high level of detail should be set up regarding the needs or purposes, separating secondary-important data for the visualization of textures.

Within the digitalized robotic cell, controlled from the single control unit in real line, data can be gathered both in historical flow and real-time independently from the real-life system. The data gathered can be used for learning processes without using the real machine, which can perform other tasks at the same time.

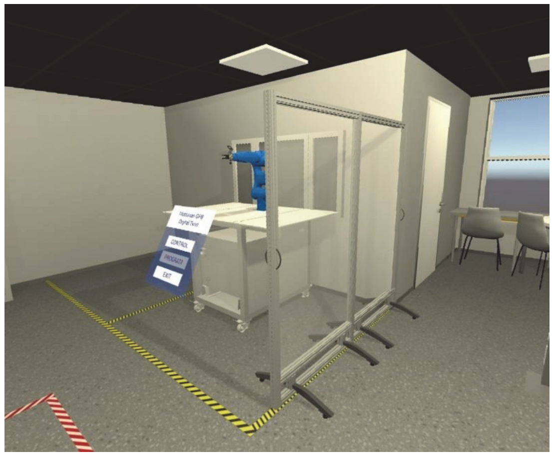


Figure 9. Digital Twin (in VR) of robotic work cell

Both approaches of DT via AR and VR will increase efficiency and teaching time, but it will decrease the usage of the real physical robot and allow it to be done independently of human factors.

ACHIEVED RESULTS AND FUTURE DEVELOPMENTS

A methodology for the successful adaptive control of robots was introduced in this paper. During this research, the following steps were achieved:

- URDF file of the robot has been created, which can be effectively used in other use-cases,
- Robot control using ROS motion planning framework,
- Proposed sub-systems have been unit tested: Velodyne, image detection,
- Different Matlab toolkits were successfully unit tested,
- Motion execution together with a VR toolkit has been implemented, giving the ability for an unlimited amount of test cases without interfering with a real robot system, but still allowing it to be reprogrammed and taught from the virtual robot self-learning experience.

As for future development, sub-experiment systems must be tied together and tested in one setup: The URDF file tested on another robot type and VR and AR open-source toolkits for robot self-learning as introduced should be tuned to and developed for the final platform.

CONCLUSION

In this paper, an adaptive industrial robot control methodology, which is ready for practical implementation, was introduced. The elaborated methodology based on AGV control systems enables one to increase the efficiency and speed of ROS based industrial and autonomous robot systems. In addition, a case study enabling immersive technologies was introduced. The experiments described are ongoing in the Industrial Virtual and Augmented Reality laboratory in Tallinn, Estonia and are part of wider research into the dynamic control of industrial robots according to the Industry 4.0 concept and development of DTs of production systems.

Future developments mainly address the modelling of fully automated detection, analysis and decision making for industrial robots. Further research also involves the use of VR/AR for machine learning in the production system as a precise Digital Twin with a full robot work cell control option has already been developed.

ACKNOWLEDGMENTS

This research was supported by project AR16077 Smart Industry Centre (SmartIC) No. 2014-2020.4.01.16-0183, supported by the EU Regional Development Fund.

The authors are grateful to the Integrated Engineering Students of the Tallinn University of Technology – Yevhen Bondarenko, and Ninel Dudnyk, for their help in the experiments with Augmented and Virtual Reality implementations.

REFERENCES

[1] Kuts, V., Tahemaa, T., Otto, T., Sarkans, M., & Lend, H. (2016 (1)). "Robot manipulator usage for measurement in production ares". *Journal of Machine Engineering*, 57-67.

[2] Kuts, V., Sarkans, M., Otto, T., & Tahemaa, T. (2017). "Collaborative work between human and industrial robot in manufacturing by advanced safety monitoring system". *DAAAM International* (pp. 996-1001). Vienna, Austria: Curran Proceedings.

[3] Tsarouchi, P., Makris, S., Michalos, G., Matthaiakis, A.-S., Chatzigeorgiou, X., & Athanasatos, A., Stefos, M., Aivaliotis, P., Chryssolouris, G. (2015). "ROS Based Coordination of Human Robot Cooperative Assembly Tasks-An Industrial Case Study". *Procedia CIRP, Volume 37* (pp. 254-259). CIRP.

[4] Bdiwi, M. (2014). „Integrated sensors system for human safety during cooperating with industrial robots for handing-over and assembling tasks". *Procedia CIRP, Volume 23* (pp. 65-70). CIRP.

[5] Vysocky, A., & Novak, P. (2016). "Human-Robot Collaboration in Industry". *MM (Modern Machinery) Science Journal*, (pp. 903-906).

[6] Tsarouchi, P., Matthaiakis, S., Makris, S., & Chryssolouris, G. (2016). "On a human-robot collaboration in an assembly cell". *International Journal of Computer Integrated Manufacturing*, (pp. 1-10).

[7] Wan, J., Tang, S., Yan, H., Li, D., Wang, S., & Vasilakos, A. V. (2016). "Cloud robotics: Current status and open issues". *IEEE Access, vol. 4*, (pp. 2797-2807).

[8] Hrabia, C. E. (2016). "A Framework for Adaptive and Goal-Driven Behaviour Control of Multi-robot Systems". *IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, (pp. 275-280). Augsburg.

[9] Proctor, F., Franaszek, M., & Michaloski, J. (2017). "Tolerances and Uncertainty in Robotic Systems". *ASME 2017 International Mechanical Engineering Congress and Exposition* (p. V002T02A075). Tampa, Florida, USA: ASME.

[10] Falleiros, E., Calvo, R., & Ishii, R. P. (2015). "PheroSLAM: A Collaborative and Bioinspired Multi-agent System Based on Monocular Vision". *In: Gervasi O. et al. (eds) Computational Science and Its Applications -- ICCSA 2015. Lecture Notes in Computer Science, vol 9156.* (pp. 71-85). Springer, Cham.

[11] Tran, T., Nguyen, T., Nguyen, M., & Pham, T. (2016). "A computer Vision Based Machine for Walnuts Sorting Using Robot Operating System". *Advances in Information and Communication Technology: Proceedings of the International Conference, ICTA 2016*, (pp. 9-18).

[12] Steck, A., Lotz, A., & Schlegel, C. (2011). "Model-Driven Engineering and Run-Time Model-Usage in Service Robotics". *GPCE'11 - Proceedings of the 10th International Conference on Generative Programming and Component Engineering*, (pp. 73-82). Portland, Oregon, USA.

[13] Petlenkov, E., Nomm, S., Vain, J., & Miyawaki, F. (2008). "Application of self organizing Kohonen map to detection of surgeon motions during endoscopic surgery". *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, (pp. 2806-2811). Hong Kong.

[14] Vassiljeva, K., Tepljakov, A., Petlenkov, E., & Netšajev, E. (2017). "Computational intelligence approach for estimation of vehicle insurance risk level". *2017 International Joint Conference on Neural Networks (IJCNN)*, (pp. 4073-4078). Anchorage, AK.

[15] Pedre, S., Krajnk, T., Todorovich, E., & Borensztejn, P. (2016). "Accelerating embedded image processing for real time: a case study". *J Real Time Image Process*, (pp. 1-26).

[16] Tilbury , D. M., & Ulsoy, A. G. (2010). "Reliable Operations of Unmanned Ground Vehicles: Research at the Ground Robotics Reliability Center".

[17] J. P. N. Cruz, M. L. (2013). "Object recognition and detection by shape and color pattern recognition utilizing Artificial Neural Networks". *International Conference of*

Information and Communication Technology (ICoICT), (pp. 140-144). Bandung.
[18] Muhammad, N., & Lacroix, S. (2010). “Calibration of a rotating multi-beam lidar”. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 5648-5653). Taipei.
[19] Kucuk, S., & Bingul, Z. (2004). “The inverse kinematics solutions of industrial robot manipulators”. *Mechatronics, 2004. ICM '04. Proceedings of the IEEE International Conference on*, (pp. 274-279).
[20] Lebedev, D. V., Steil, J. J., & Ritter, H. J. (2005). “The dynamic wave expansion neural network model for robot motion planning in time-varying environments”. *Neural Networks, Volume 18, Issue 3*, (pp. 267-285).
[21] Kuts, V., Modoni, G. E., Terkaj, W., Sacco, M., & Otto, T. (2017). “Exploiting factory telemetry to support Virtual Reality simulation in robotics cell”. *Augmented Reality, Virtual Reality, and Computer Graphics, 1:4th International Conference, AVR 2017* (pp. 212-221). Ugento, Italy: Springer.
[22] Guhl, J., Tung, S., & Kruger, J. (2017). “Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft HoloLens”. *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, (pp. 1-4). Limassol.
[23] Brizzi, F., Peppoloni, L., Graziano, A., Stefano, E. D., Avizzano, C. A., & Ruffaldi, E. (2018). “Effects of Augmented Reality on the Performance of Teleoperated Industrial Assembly Tasks in a Robotic Embodiment”. *IEEE Transactions on Human-Machine Systems, vol. 48, no. 2*, (pp. 197-206).

Publication V

Kuts, V.; Otto, T.; Tähemaa, T.; Bondarenko, Y. (2019). Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality; Journal of the Machine Engineering, 2019.

Journal of Machine Engineering, 2019, Vol. 19, No. 1, 128–145
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received: *19 December 2018* / Accepted: *15 February 2019* / Published online: *11 March 2019*

digital twins, industrial robots,
virtual factory, virtual reality

Vladimir KUTS[1*]
Tauno OTTO[1]
Toivo TÄHEMAA[1]
Yevhen BONDARENKO[1]

DIGITAL TWIN BASED SYNCHRONISED CONTROL AND SIMULATION OF THE INDUSTRIAL ROBOTIC CELL USING VIRTUAL REALITY

During the years common understanding of the possibilities and perspectives of Virtual Reality (VR) usage has been changed. It is thought that VR is mainly used in entertainment purposes, but it is being used already for many years in different industries, and now with easier access to the hardware it became a helpful and accessible tool that could be used and developed in any field of human activities. In manufacturing, immersive technologies are mainly used nowadays for the visualisation of processes and products combining those visuals into the factory Digital Twin (DT) which is possible to view from the inside look. This feature is already being used in several manufacturing simulation tools, which enable to view onto industrial line / robotic cells via Virtual Reality glasses. However, the potential of using simulations with VR in manufacturing is not fully uncovered. The main aim of this, industrial robotics targeted research is to enable besides simulation also universal control algorithms through Virtual Reality experience, produced by game engine Unity3D, which can be easily modified for a wide range of industrial equipment. The primary outcome of this work is the development of the synchronisation model of real and virtual industrial robots and experimental testing the developed model in Virtual Reality and shop floor labs

1. INTRODUCTION

Years ago, most people thought that Virtual Reality (VR) could be used only for gaming and another type of entertainment purposes, but actually, it is being used in research already few decades. Now VR could be used in any sphere of human activities. For example, it can be used in architecture or design, where people can firstly try their projects in a digital environment before going to real projects. This way can prevent dozens of mistakes and errors without any loss. Also, VR can be a good helper for education in schools and universities. Students will be able to see how theoretical knowledge from lessons could be implemented in real life. One more approach is professional work power

[1] Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering, Estonia, Tallinn
[*] E-mail: vladimir.kuts@taltech.ee
https://doi.org/ 10.5604/01.3001.0013.0464

training. VR technologies will be the proper apparatus to simulate working environments for medical, military or aviation purposes. To be more precise – there are different research areas, where control and improvement of how to better use and visualise different information, about the VR and Augmented Reality (AR). VR simulation could also be used in manufacturing, robotics and control systems [1–4] by simulating different algorithms and control methods. Manufacturing, robotics, and control Architecture [5] of varying scale rooms and buildings, for finding the most efficient way of design. Audio [6] – to use VR as a sound visualisation tool for the artist or blind people. Moreover, smart racks and simulations are done with force feedback [7, 8] for a better haptic feeling of tested in VR joysticks, devices or manipulators. So of course entertainment and education [9, 10] area for precise training simulations for different level workers and students.

From the other side, some issues may arise. First of them is the lack of people familiar to the VR field. As this technology is new and not so widely known, there is still a lack of developers, who have enough experience to create a ready-to-go project, because the production process of VR environment is hard work. In comparison to mobile applications, where one programmer or artist can do all the stuff, as pixel graphics and writing the code, VR requires high-quality 3D models, as well as, an understanding of human feelings - motion sickness. Also, it is essential to have a team, because developing in VR means testing application alongside with programming and doing it alone may be very time-consuming. The last, but not the least, it is hardware. Technologies such as VR requires very powerful newest personal computers (PC) with headsets and stations, such as HTC Vive or Oculus Rift, which can be very costly for small and medium enterprises (SME-s) and stand-alone developers. This entire means that integration of VR technologies to the SME-s is still complicated because of the lack of workforce or knowledge about the possibilities.

The practical aim of the research is to create Industrial Digital Twin (DT) – a digital copy of the real manufacturing system, which can be controlled and programmed in real-time directly from the computer application model of the industrial robot. It includes the creation of the precise model of the robot and developing a software package to control and program it directly from VR. The work also analyses how creating the DT can improve workspace awareness of the real robot without using any additional physical equipment, but an only accurate computer simulation. Moreover, our tool, developed during this research, with the usage of immersive technologies, is able to visualize in real scale manufacturing lines and robotics cells not only in a purpose of simulation and demo, but also for control of the actual work process – interactive online tools, which gives ability to re-program line in real-time. With it, downtime is reduced to the minimum, as all optimisations and new product production are being done in DT and then via network transferred to middle layer controller in seconds, allowing using the new program from a new loop of the process in reality. Thus, reduces downtime and money waste on re-programming and it increases the overall efficiency of the manufacturing process. Moreover, the aim of this project is a creation of the very flexible, user-friendly and modular environment, which can be easily modified, connected to the real manufacturing assets, and accessible by the broad public.

Toward this research, an experimental approach to the development of the methodology is being introduced. Both method, how-to and a use-case are combined into the main section. The outcomes of this work were tested and validated as a part of a more significant project in Tallinn University of Technology – a recreation of the Industrial Virtual and Augmented Reality Laboratory (TalTech IVAR laboratory) in VR.

2. DEVELOPMENT OF THE EXPERIMENTAL ENVIRONMENT

This part of the research describes the main development steps that were taken to implement the DT of industrial robot – Motoman GP8 in VR. Each of the steps, which include model preparation, programming and optimising for VR, their challenges and outcomes are analysed in detail in the corresponding sections. The primary software tool selected for the experimental research realisation is the Unity3D game engine. This approach can be developed in various similar engines, for example, Unreal Engine (UE), but because of the previous author's experience was chosen Unity, not to spend time on re-learning. Unity provides a simple but powerful development environment with a modular approach to programming and also offers integration with all commercially available VR systems, which is perfect for the defined task. 3DS Max and Maya from Autodesk were used as 3D modelling software for this project. All development cycles mentioned above are thoroughly explained in the corresponding sections of this paper. Also, the project described above was initially designed to work with the specific hardware: HTC Vive headset for VR capabilities and Yaskawa Motoman GP8 industrial robot for testing the DT concept. However, the software was developed with a modular programming approach in mind and, as a result, it can be easily extended to support other current VR platforms and to control DT of many different models of industrial robots and other equipment. Mainly, immersive technologies and tools of how to make those are being used for visualization and just a simulation of production processes [11], but more and more research is done on integration of different control inputs like Robot Operation System (ROS) to the different layers of simulations and modelling a test environment [12–15]. In the methodology described below, we intend to combine all mentioned above together and propose an alternative to the ROS environment.

2.1. PREPARATION OF THE ROBOT MODEL

The simulation required both models of an industrial robot and its training station. Digital models of the robots can be taken from the manufacturer website or related software libraries. Though the models imported, were built in proportions exactly corresponding to their real-world counterparts, there were several important issues to address before importing them to Unity3D game engine:

- Robot model had to be rigged. Rigging in related context means defining the location of pivot points in the models so that the program can get the axes around which robot's links are rotated. This operation had to be done precisely to keep the quality of the simulation.

- The robot stand model had to be simplified for VR because the original geometry was too “heavy” to be rendered in real time.
- Both models had the correct proportions when exported from manufacturer website or software, but their scale had to be checked and adjusted manually after importing to Unity3D.

Rigging of the robot model can be done using Blender, Autodesk 3DS Max or other similar software which can convert CAD models into .fbx file format – for this research 3DS Max were used. The process of rigging consisted of defining the correct coordinates of the model’s pivot points (robot’s axes) and aligning robot meshes (3D geometry) to them. Besides, it was essential to set up all links and joints of the robot into the correct hierarchy. Hierarchy allows the model to be controlled by game engine in the desired manner: when the parent geometry is moved (for example, the first joint of the robot is rotating), all child geometry follows along (part of the arm located above the actuated joint is rotating too). The hierarchy was set up in such a way that the robot's rig (relative locations of the pivot points) is not affected when adding and scaling link meshes to it. That is useful because the exported geometry itself consists of polygons, and never can correctly copy the dimensions of the real object. However, it is possible to give exact coordinates to the model's pivot points and maintain an ideal accuracy between the real robot and model in joints and end-effector positions.

Figure 1 demonstrates how pivot points of the model in 3D modelling software copy the robot’s geometry from the mechanical drawing.

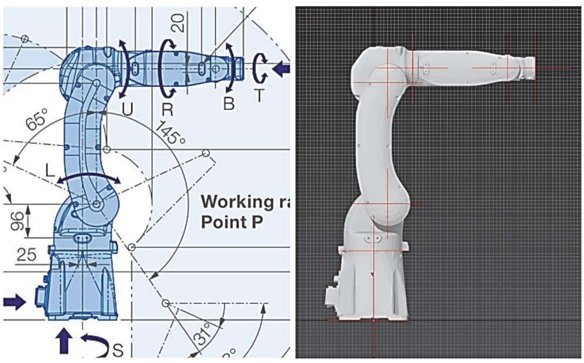


Fig. 1. Comparison of the robot’s mechanical drawing and its pivot point representation in 3DS Max

Optimising of the industrial robot stand model was done in Autodesk Maya. It was chosen with reason as this 3D modelling software simplifies work with high-polygonal models and provides tools to simplify their geometry. The first part of optimisation was reducing the polygon count of the model, which, as a result, has decreased from more than 1.000.000 faces to around 80.000 without losing dimensional precision of the model. The second part of the optimisation process was minimising the count of materials used in the model after importing it to Unity3D. Materials of the model define the way it looks in a 3D application (i.e., colour, surface texture). The problem was in the number of materials generated for the model when it was exported from manufacturer software – several thousands of materials were created and added to the model because the program produced new material for each model's mesh. As a result, the model was unsuitable for real-time rendering, especially in VR. The solution was .to remove the generated materials and create several new ones to replace them. Achieved result is the model, which is ready to use in the simulation and without the significant loss of visual quality

Scaling the models was the last part of geometry preparation before the actual programming of DT. To represent the real robot station in the correct size in VR, both the stand and the robot models had to be downscaled to the right dimensions and positioned correctly relative to each other. The operation was non-complex due to the dimension system of Unity, where 1 unit of distance in the game scene equals 1 meter in the real world. This dimensioning is being preserved when inside a VR simulation, so the user can experience the robot's model in 1:1 scale stereo environment, which gives to person very precise presence feeling.

2.2. ROBOT CONTROL SCRIPTS

To power the Digital Twin system created in this project, some scripts in C Sharp (C#) language was designed and tested in the Unity3D game engine. Unity3D uses a modular approach to application development, which is implemented as Game Objects (models, geometry, effects, etc.) and Components (C# scripts which control Game Object behaviour) attached to them. To make further future developments simpler was decided to maintain this modular approach when developing the project's programming base as well.

The scripts used in the experiment can be separated into three major parts according to their functions:

- Control scripts family, which provides methods to control a generic industrial arm robot model. DT controller belongs here as well.
- Programming scripts, which allow creating simple systematic programs, which can be later, run on the robot models managed by Control scripts.
- Collision detection scripts, which monitor the position of the robots led by Control scripts to check for potential collisions with environment objects and stop when a possible collision is detected. These scripts are intended to be used with virtual robot models inside Unity3D.

Robot control scripts are built in a hierarchy structure, which can be seen in Fig. 2. The functions of each script are described in the following sections.

Base control script

RobotController is an abstract base class, which means that it contains no actual code to be run, but the definitions of methods and logic, which has to be implemented by any class inheriting from it.

RobotController class is the core of all application structure because it is an element, which enables the universality and consistency among the interfaces of all other Controller classes. Due to it any other script, for example, RobotProgrammer can send commands to the controlled robots (whether real or virtual) without the risk of producing an error because of a non-existing method called.

Here is the list of public (which can be accessed by any other scripts) methods and properties defined in base RobotController class:

- public bool isMoving – returns true if the controlled robot is currently moving,
- public void SetSpeed() – sets the speed of the robot (0 to 100%),
- public float GetJointAngle(int jointNumber) – returns the angle of the given robot joint,
- public List<float> GetJointsAngles() – returns the list with all current joint angle values of the robot,
- public void MoveJointToAngle(int jointNumber, float angle) – moves given robot's joint to a given angle with the currently set speed,
- public void MoveJointsToAngles(List<float> targetAngles) – accepts a list of angle values, and then moves all joints of the robot to these corresponding angles with the currently set speed,
- public abstract void MoveToEndpoint(Transform endPoint) – accepts Transform (Unity's representation of position in 3D space), then moves robot's end-effector to this point using Inverse Kinematics (IK) with the currently set speed,
- public abstract void MoveRobotToZero() – a shortcut command to move all robot's joints to their zero positions,
- public abstract void Stop(bool emergency = false) – stops the current robot movement as a default; if the given emergency parameter is true, stops the robot urgently (i.e., disables servos when controlling the real robot).

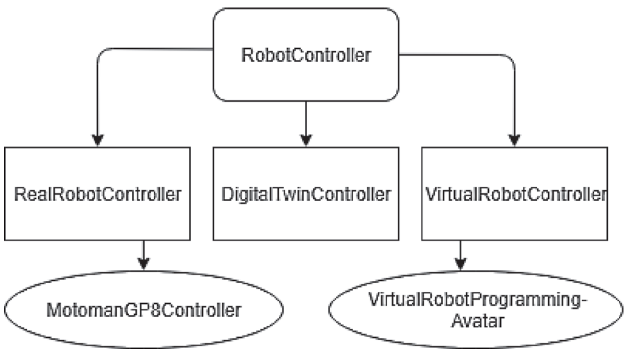


Fig. 2. Robot control scripts hierarchy

Also, RobotController defines some protected methods to be used only inside the deriving classes:

- protected abstract void InitializeRobot() – this method is run internally when the control script loads; intended for setting references and initialising all required parameters of the robot,
- protected abstract void MoveRobotJoints(List<float> targetJointAngles) – moves all joints of the robot to the corresponding angles set in targetAngles list; this function is

executed internally each time MoveJointToAngle(…), MoveJointsToAngles(…), MoveToEndpoint(…) or MoveRobotToZero () methods are called on the robot,
- protected IEnumerator RobotMovement_Coroutine(List<float> targetJointAngles) – an internal Coroutine (the method that is run in a loop over multiple game frames) which implements joints' movement process to the given angles with set speed,
- protected IEnumerator RobotMovement_Coroutine(Transform targetPoint) – an internal Coroutine, an analogue of the previous one, but is used for movement using IK.

Again, these methods are implemented in each class inheriting from RobotController. The code that is run inside these methods for each class can be different depending on the type of robot being controlled; however, the input and output parameters always follow the same pattern. Given this, if somebody creates a new controller script for a new robot model, it is going to work with this application – thus simplifying the development of new DT controllers.

Virtual robot control

VirtualRobotController is created to control a generic virtual industrial robot model inside Unity3D. It uses a supplementary VirtualRobotJoint script to manipulate its joints and can also connect to Inverse Kinematics solver to support MoveToEndpoint(...) command.

To set up a new virtual industrial robot for control in Unity using this script, a developer needs to execute several steps:
- Import a correctly rigged model of the desired robot to Unity.
- Add VirtualRobotJoint script to each joint of the robot model and define the joint's rotation limits in this script's Component interface.
- Add VirtualRobotController script to the root of the robot model hierarchy.
- (Optional) Add Inverse Kinematics solver script if MoveToEndpoint(...) command needs to be implemented.

The virtual robot has a default maximum speed, which can be set by the developer inside Unity; actual robot speed is set inside the application as a per cent of this maximum value. If at some point the speed of the VirtualRobotController is set to zero, it does not stop moving but instead moves immediately to the given joint angles or endpoint. This feature is implemented for the cases when a virtual robot needs to immediately synchronize its position according to some values (in case of DT application, sync with the real robot).

Real robot control

Regarding this work the control, Application Programming Interface (API) created by the industrial robot manufacturer was used to control the robot over the local network. Because the API itself is distributed under the Non-Disclosure Agreement (NDA), the code samples using it and the explanations of its inside functionality cannot be published in this work. However, it is sufficient to explain the underlying logic of how the real robot control is implemented in this DT project to understand the principle and apply it in other experiments.

The RealRobotController script is intended to be used for real industrial robot control over the network. It inherits directly from RobotController but is also declared as an abstract class, because it contains a couple of extra properties and methods which are dictated by the necessity to connect and continuously monitor the state of the linked real robot. These consist of the following:

- public bool isConnected – returns true if the connection with the robot is successfully established, returns false otherwise,
- Protected IEnumerator RobotStateMonitor_Coroutine() – an internal Coroutine which manages the link to the robot and periodically updates the status information about it (i.e., joint positions, system status, error messages).

These small additions create a base for writing scripts, which can be used to control and monitor real industrial robots from Unity3D. While the interface stays the same (RobotController), class methods can now utilise an internal protocol for communication with the real robot controller over the network, and the script can act following the state data received from the actual controller, creating a closed feedback loop with the robot. The protocol implementation depends solely on the company, which has produced the robot and can be integrated into the solution as a new script inheriting from RealRobotController. The control of researched robot was implemented in the form of MotomanGP8Controller class, which is inherited from RealRobotController and implements its methods.

It is also important to note that in the controller script it was managed to achieve the joint angle setting accuracy which is identical to the actual precision of the robot – which is as high as 0.001°. That means that any time the robot's joint angles are set from the simulation, the real robot moves to the target angles with the same precision.

Digital Twin system

With the programming basis of the project implemented inside RobotController family of scripts, namely VirtualRobotController and RealRobotController, creating a Digital TwinController script, which would enable Digital Twin functionality in the developed application, was near to a “plug-and-play” process. DT solution for industrial robots developed concerning this research work according to the following logic:

- DigitalTwinController script acts as a coordinator between two scripts in Unity: VirtualRobotController and MotomanGP8Controller.
- MotomanGP8Controller connects to Motoman GP8 robot over the local network and continuously monitors its state.
- DigitalTwinController reads position data received by MotomanGP8Controller and redirects it to VirtualRobotController script.
- VirtualRobotController synchronises the virtual robot’s joints’ positions according to received values, and, as a result, copies all movements of the real robot.

When a movement command is sent to DigitalTwinController (for example, MoveRobotToZero()), the script first sends this command to the real robot, and in the next update loop virtual robot gets synchronised with the real one again, creating a smooth DT experience. Moreover, if the teach pendant, which is the control panel attached to the real robot, overrides the real robot program control, the twin is still going to mimic the movements of its real counterpart. The solution also remains safe, because even when somebody is manipulating the robot from VR environment, the proximity sensor built into the experimental stand continues monitoring surroundings, and stops the robot if someone comes dangerously close in the real world. A schematic representation of the Digital Twin system can be seen in Fig. 3.

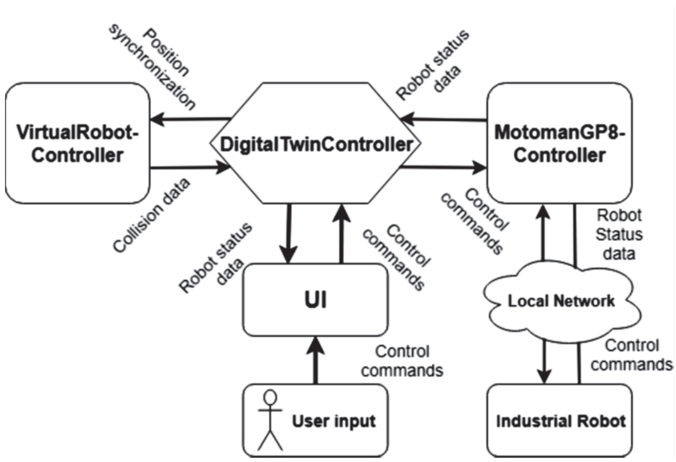


Fig. 3. Digital Twin system diagram

2.3. ROBOT CONTROL SCRIPTS

Robot programming scripts family contains three classes, which are responsible for providing robot programming and testing functionality as well as methods to simplify these processes. These scripts are:

- RobotProgrammer.
- RobotTester.
- VirtualRobotProgrammingClone.

Main programming script

The programming functionality for Control scripts is added using the Robot Programmer Component. It is a simple programming manager, which was created to demonstrate how it is possible to program robot directly from Unity without using the native commands, which can be different depending on the robot's model and implementation.

The script contains methods to create new programs for robot controllers and store them for later use. RobotProgrammer must be connected to a specific RobotController inside Unity to make it possible to run the created application. Currently, three types of commands can be added to the program using the following methods:

- public void AddPointToProgram(List<float> jointAngles) – adds a point the robot should move to; the point is given as the list of similar joints' angles,
- public void AddWaitToProgram(float waitTime) – adds pause which lasts for the number of seconds specified in the waitTime parameter,
- public void AddGripperAction (bool action) – adds gripper action to the program (if the given action parameter is true, the robot will close the gripper, if false open it).

These three simple command types allow creating demonstrative which can be run on all robot controllers, from virtual to DT.

Internally, RobotProgrammer stores the programs as RobotProgram objects (a utility class declared inside RobotProgrammer script), which, in turn, contain the lists of ProgramAction objects (another utility class inside RobotProgrammer, which is responsible for storing and interpreting the steps added to the algorithm utilising methods described above.

To run the currently selected program in RobotProgrammer, one has to call a RunProgram(...) method from it. RunProgram(...) will automatically parse the currently selected program and send the corresponding commands to the connected robot using the same RobotController API, with a 32,875 ms of average delay between each command to ensure stability.

Robot testing script

A script called RobotTester was created to speed up the development cycle by providing a simple tool, which can be used to test the newly written RobotController scripts. It contains methods, which call the usual position and movement functions from RobotController, but provides a Unity3D interface, which helps to quickly switch between these methods, as, can be seen in Fig. 4.

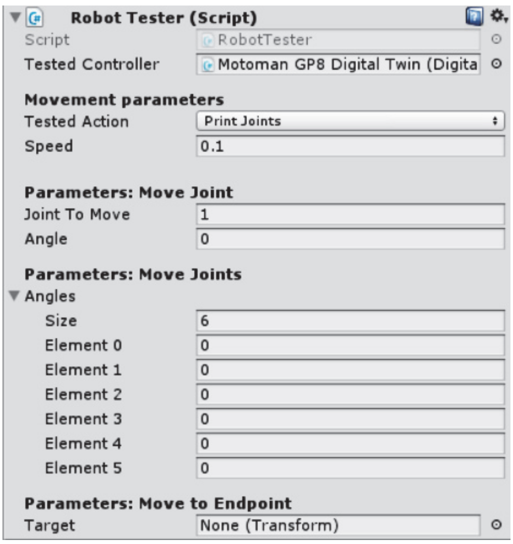


Fig. 4. RobotTester script Component interface

As can be seen from Fig. 4, this component provides a simple interface where the developer can define the input parameters and select the tested method. Action to be tested is selected using a drop-down menu, which provides the next options: “Print Joints”, “Move Joint”, “Move Joints” and “Move To Endpoint”. Input parameters for each of these commands can be given further in the interface (joint number and angle for “Move Joint” command, joints' angles list for "Move Joints" and target Transform for "Move To Endpoint"). After the necessary action type and parameters are selected, the test can be performed by merely calling Test() method of this RobotTester script. This script has dramatically simplified the development and testing of new RobotController scripts during this research.

Robot programming virtual representation (digital clone)

The final goal in the development of robot programming system was to implement the way to make the programming process itself more understandable and straightforward for the user. It resulted in the creation of the VirtualRobotProgrammingClone script. This script is inherited from the VirtualRobotController, and its purpose is to create a copy of the programmed virtual robot with the same parameters placed in the same position as the original. This digital clone can be used to visually set the target points when programming the robot, with the need to move the original – this feature is especially useful when programming a real robot controller (see Fig. 5).

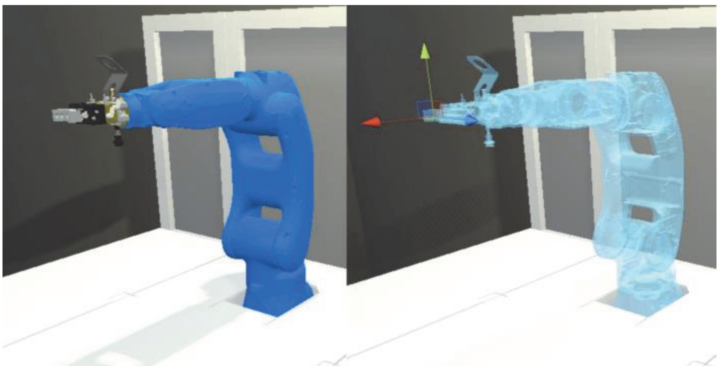


Fig. 5. Digital Twin robot (left) and its digital programming clone with IK endpoint target (right)

VitrualRobotProgrammingClone allows setting the positions of the joints as well as endpoints using steps, which makes it simpler to set the points from an interface. While joints are actuated the same way as it is done in other RobotController scripts, digital programming clone in this project also uses IK solver, which allows setting the robot position by moving its endpoint target. There was no need to develop the IK solver from scratch, as the Unity engine already provides different ready-to-go IK options.

The first available option is to use the built-in Unity IK system called Mecanim, but another tool was selected for this project – Final IK Unity plugin. Final IK is a full-packed IK system developed in Estonia specifically for Unity game engine. Though this plugin is not free, the license has to be purchased only one time, and it is more flexible and user-friendly than the Unity's built-in Mecanim system. Final IK also allows creating custom IK chains and can be used in the industrial robot model. So, Final IK solver was used to calculate the joints' positions of the digital programming clone when controlling it in IK mode. A schematic of the robot programming system can be seen in Fig. 6.

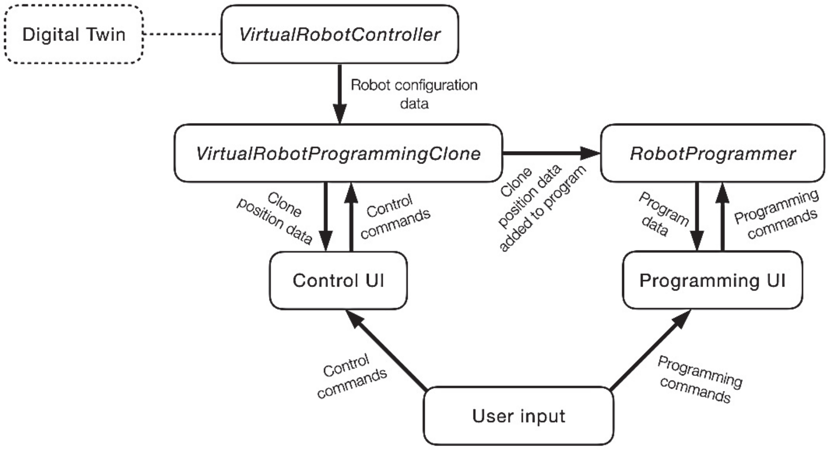


Fig. 6. Robot programming system diagram

Programming User Interface (UI)

As a final goal, the DT system developed in this project was integrated into the full-scale simulation of the University's IVAR Laboratory. To the projected were implemented the User Interface (UI) for robot control and programming using APIs of the scripts created concerning this use-case development. It created a possibility to test the developed scripts right from VR, using HTC Vive headset, resulting in positive outcomes. The visual look of the UI is presented here for the reference of what can be done using the scripts developed concerning this project (see Figs 7 and 8).

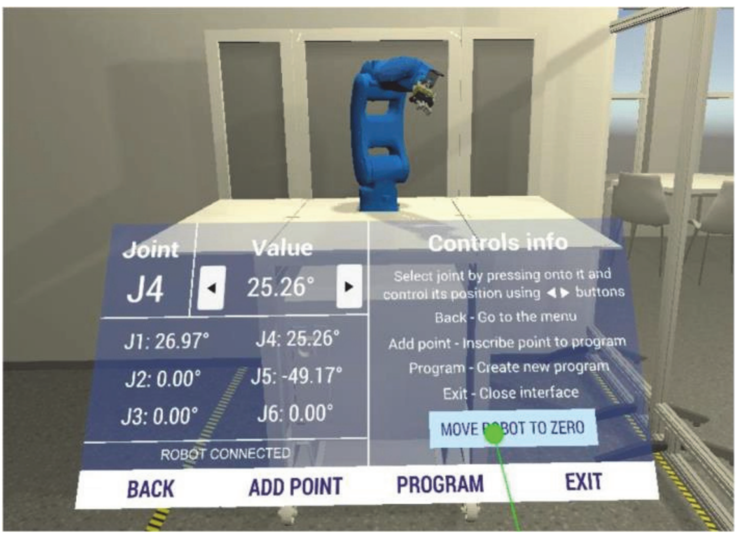


Fig. 7. Digital Twin control menu

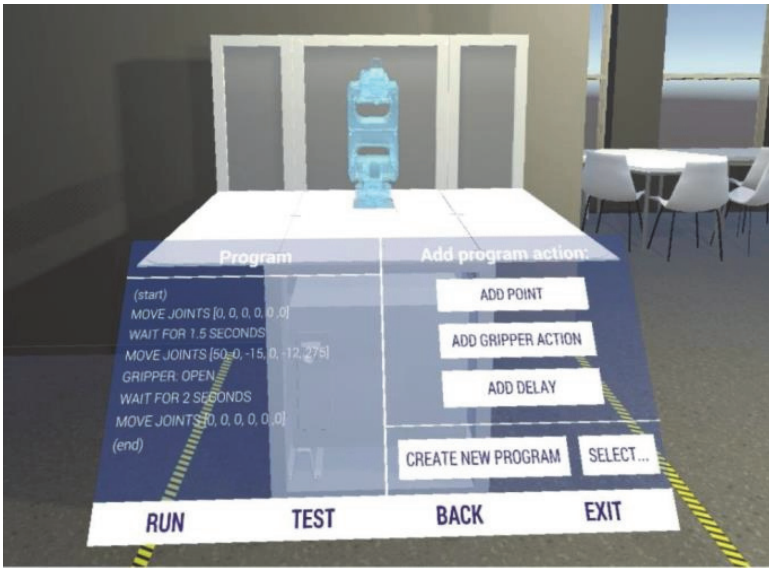


Fig. 8. Digital Twin programming menu (the virtual robot is replaced with its digital programming clone)

2.4. COLLISION DETECTION SCRIPTS

One additional goal of this research was to develop a method, which can improve the workspace awareness of the real robot by using its DT. As a result, two scripts were developed: RobotCollisionRollback and RobotCollisionAware. Both scripts have the same basic idea implemented in them: a DT is an exact digital copy of the real robot; given this, it should be possible to use the DT's geometry to monitor the position of the robot in its workspace and prevent it from accidental collisions with the environment using Unity physics system. If the models of both the robot and its working cell are made precisely (which is the case for this project), they can be used with a quite high certainty for collision prevention. Of course, the joint limits can always be set on the real-world robot itself, but this process takes time and requires thorough planning, and never guarantees that all possible collision scenarios are eliminated. Another option is to supplement the real-world robot with proximity sensors on each of its links and monitor collisions using these sensors – but this method is very costly and even more time-consuming. The DT collision detection approach, proposed in this work, provides almost the same level of reliability but does not require any extra physical equipment or significant setup time (given that the DT model is already made and has correct dimensions set). All that is needed for this approach to work is to generate colliders for the robot's model in Unity3D and attach one of the collision detection scripts described below.

The RobotCollisionAware script uses Unity physics system to detect collisions of the robot with the environment. Trigger colliders attached to the robot's model can detect collisions with other objects before the robot's geometry is going actually to encounter them. That can be used for example for prohibiting certain positions when using a digital programming clone and trying to add points, which can cause collisions to the program.

RobotCollisionRollback is used for the DT model itself. It is an enhanced version of RobotCollisionAware script, which is intended to prevent the real robot from colliding with its stand during direct control. In a case when the robot is about to bump into the stand, this script will call an emergency stop command, and then automatically move it back to the previous safe position. This method makes controlling the robot from VR safe for both the robot and its surroundings.

2.5. PROJECT OPTIMISATION FOR VIRTUAL REALITY

To enable the VR capabilities in the developed Unity project, Virtual Reality Toolkit (VRTK) Unity script library was used. VRTK is an exceptional example of open-source software. This library contains scripts, which simplify the process of building VR applications for Unity to a great extent, and furthermore, the applications developed using VRTK are elementary to modify and add support of different platforms. The project of this research was initially built to be used with the HTC Vive headset, but thanks to its VRTK basis it can be ported to all common VR platforms, including Oculus Rift and Microsoft Mixed Reality (MR).

During the DT control scripts tests in VR, it appeared that the code was causing a significant framerate drops when sending the commands to the connected robot. Framerate is the speed at which the application is rendered, measured in frames per second (FPS). The stable framerate of 90 FPS is crucial for good VR experience because framerate drops to the numbers lower than this are immediately causing discomfort or even nausea to the user of the application.

After the problem investigation, it was revealed that the problem was caused by industrial robot API used in MotomanGP8Controller to send commands to the robot – while the commands had to be sent asynchronously to avoid affecting framerate; API sent synchronous calls to the robot. Because of this, an application had to wait for the command to be successfully transmitted over the network and confirmed, causing it to freeze in the same frame until the command is executed. This effect caused the framerate to drop the framerate below 20 FPS – making the application practically unusable for VR. This problem had to be overcome to finish the experiment successfully.

The solution was to add multithreading functionality to the application. Unity API itself is single-threaded, so the multithreading solution had to be implemented independently of Unity scripts. After some experimental research, the solution was found – to create a ThreadedJob class, which can be extended into separate job classes, which run their code in parallel threads. With the usage of these classes as a base, the calls to the robot were moved into separate job classes called MotomanMonitorJob, MotomanMoveJointsJob, and MotomanStopJob. Because of such architecture change, all framerate problems cause by the robot API were eliminated – what made application to be able to run smoothly, and its performance does not depend on the commands sent to the robot.

3. EXPERIMENT/USE-CASE

As it was mentioned earlier in this paper, the VR environment for the robot's Digital Twin was created concerning the project of digitalisation of TalTech IVAR Laboratory. Thus said, before the system could be used for the demonstrations and educational purposes in the Laboratory, it had to undergo testing for safety and working stability.

The most critical part of the DT simulation was collision detection, as the safety of the real robot and its surroundings directly depended on this feature. After subsequent control tests, it was determined that the DT collision prevention system, described in section 2.4 of this article, was capable of timely stopping both virtual and real robots if the upcoming collision with the surrounding geometry was detected. It is important to note here, that the effectiveness of this system depends on the precision of the robot work cell/environment reproduced in VR simulation. If it is represented incorrectly, there is a chance of robot hitting objects in the real world. On the contrary, this hazard can be quickly eliminated by careful planning and dimensioning when reproducing the robot system in 3D.

Another essential feature of the system confirmed during testing was the possibility to override control over the system from inside the real world – whether by the teach pendant or by the proximity sensors installed in the robot's stand. In the first case, the robot could be set to manual control mode and manipulated directly from it's taught pendant, while the DT inside the running VR simulation would continue reproducing all movements of its real counterpart. In the second case, if some person approached too close, it would trigger the proximity sensor, automatically stopping the real robot from moving and thus guaranteeing safety to the people around the machine; again, the DT would stop as well, staying synchronised with the real robot. From the point of practical usability, the system has proven itself an excellent tool for DT technology demonstration and education, confirmed by the university's robotics specialist. DT makes interaction with the robot much safer, simpler and more engaging by utilising all the unique features the VR has to offer, i.e., real-size stereo picture, highly flexible 3D interfaces, etc. The system was also operated by numerous students and was said to be easier to understand and control for people unacquainted with robotics. Currently, it is being used in the TalTech IVAR Laboratory demo centre for research projects and demonstration purposes daily.

With certain improvements in stability and added features, such as support for custom industrial robot tools control, the developed concept can with the most certainty be used for industrial control and production purposes.

4. RESULTS AND FUTURE WORK

The simulation system developed in this work, which consists from the full synchronisation between real and virtual industrial robot, implements a universal software base, which can be extended to control and program DT of different industrial robots by the usage of the Unity3D game engine. It also presents an idea of how DT can improve the workspace awareness of its real-world counterpart using the collision prediction system inside the simulation – a viable alternative to cost- and time-consuming physical sensor solutions.

Overall developed DT system provides the following advantages for the manufacturing lines and robotic cells:

- Enlargement of equipment functionality with smaller cost – virtual sensors can control real machines, as with collision detection case, described in above sections.
- Flexible – modularity and universal and user-friendly UI adjustable for various type of equipment.
- Production monitoring – data visualisation from sensors, fast optimisation of production, preliminary maintenance – based on data received from DT.
- Historical data in real time – saved into log kinematics data allow to "travel back" in VR headset and check what went wrong or how processes could be optimised better for a previously produced product.
- Safety systems tests – human-machine interaction scenarios tested in the digital environment.
- Personnel training – training on the machine without interfering with real one, not causing loose of production time of it.
- Machines are not stopped while re-programming, which lead to a decrease of downtime, which reduces cost on re-programming.

This project, however, can be extended and improved to fit a much higher usability scale. The Unity3D script base of the project is built in a modular manner, which can allow industrial robot manufacturers and researchers to extend the current program and add support of their machines without exposing the internal control protocols, and still leaving the solution compatible with the interfaces used in this application. The project code development is being continued to improve stability and add new useful features.

Next step in the development is also the analysis of the connection and exploiting manufacturing equipment and virtual model telemetry [16, 17] data to find a most effective and precise way of the dual-way communication and create the framework for most effective connection between real and virtual environments.

5. CONCLUSION

Virtual and augmented reality technologies are indeed becoming the practical tools of Industry 4.0 and rapidly expanding their markets. While AR enhances the way industry workers can interact with the real-world equipment, VR presents another breakthrough concept – DT, a technology that allows overseeing and controlling existing manufacturing systems from a safe yet highly intuitive and interactive real-size stereo simulation. The goal of this research was to implement such a system on the example of a real industrial robot, and it was fulfilled successfully.

The DT robot created is the central framework part of the TalTech IVAR laboratory digitalisation project, where it works in tandem with the real industrial robot located in the actual laboratory. The solution was already assessed and tested by several students and robotics specialists, who confirmed its applicability for industrial demo and educational purposes.

As a conclusion, it can be stated that Digital Twin concept is a practically viable industrial solution, which can start driving control and management systems of enterprises in the nearest future. As for future development, the done environment will be redefined for the modular approach and will be continued work towards optimisation of synchronisation framework between two worlds, what is a part of ongoing research on existing DT model optimisation.

ACKNOWLEDGEMENTS

This research was supported by project AR16077 Smart Industry Centre (SmartIC) No. 2014-2020.4.01.16-0183, supported by the EU Regional Development Fund. The authors are grateful to the Integrated Engineering and System Control students of the Tallinn University of Technology – Ninel Dudnyk and Tengiz Pataraia, for their help in the experiments with Virtual Reality implementations and laboratory digitalisation.

REFERENCES

[1] OYEKAN J.O, HUTABARAT W., TIWARI A., et al., 2019, *The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans*, Robotics and Computer-Integrated Manufacturing, 55, 41–54.

[2] KOT T., NOVÁK P., BAJAK J., 2018, *Using HoloLens to Create a Virtual Operator Station for Mobile Robots*, 19th International Carpathian Control Conference (ICCC), Szilvasvarad, 422–427.

[3] CHEN C., SU B., GUO M., ZHONG Y., YANG Y., KUO H.L, 2018, *Applying virtual reality to control of logical control mechanism system*, IEEE International Conference on Applied System Invention (ICASI), Chiba, 520–523.

[4] COVACIU F., PISLA.A., CARBONE G., PUSKAS F., VAIDA. C., PISLA D., 2018, *VR interface for cooperative robots applied in dynamic environments*, IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, 1–6.

[5] KOSE A., PETLENKOV E., TEPLJAKOV A., VASSILJEVA K., 2017, *Virtual Reality Meets Intelligence in Large Scale Architecture*, De Paolis L., Bourdot P., Mongelli A. (eds) Augmented Reality, Virtual Reality, and Computer Graphics, AVR 2017. Lecture Notes in Computer Science, 10325. Springer, Cham, Ugento (Lecce), Italy.

[6] KOSE A., TEPLJAKOV A., ASTAPOV S., DRAHEIM D., PETLENKOV E.K., VASSILJEVA K., 2018, *Towards a Synesthesia Laboratory: Real-time Localization and Visualization of a Sound Source for Virtual Reality Applications*, Journal of Communications Software and Systems, 14/1, 112–120.

[7] BURDEA G.C., 1999, *Invited review: the synergy between virtual reality and robotics*, IEEE Transactions on Robotics and Automation, 15/3, 400–410.

[8] MCNEELY W.A., 1993, *Robotic graphics: a new approach to force feedback for virtual reality*, Proceedings of IEEE Virtual Reality Annual International Symposium, Seattle, WA, USA, 336–341.

[9] JI W., YIN S., WANG L., 2018, *A Virtual Training Based Programming-Free Automatic Assembly Approach for Future Industry*, IEEE Access, 6, 43865–43873.

[10] MINER N.E., STANSFIELD S.A., 1994, *An interactive virtual reality simulation system for robot control and operator training*, Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 2, 1428–1435.

[11] GUCWA K.J., CHENG H.H, 2018, *RoboSim: a simulation environment for programming virtual robots*, Engineering with Computers, 34/3, 475–485.

[12] KUTS V., SARKANS M., OTTO T., TÄHEMAA T., 2017, *Collaborative Work Between Human And Industrial Robot in Manufacturing by Advanced Safety Monitoring System*, Proceedings of the 28th DAAAM International Symposium, Vienna.

[13] VUNDER V., VALNER R., MCMAHON C., KRUUSAMÄE K., PRYOR M., 2018, *Improved Situational Awareness in ROS Using Panospheric Vision and Virtual Reality*, 1th International Conference on Human System Interaction (HSI), Gdansk, 471–477.

[14] SURESH A., GABA D., BHAMBRI S., LAHA D., 2019, *Intelligent Multi-fingered Dexterous Hand Using Virtual Reality (VR) and Robot Operating System (ROS)*, Kim JH. et al. (eds), Robot Intelligence Technology and Applications 5, RiTA 2017, Advances in Intelligent Systems and Computing, 751, 459-474.

[15] ROLDÁN J.J., PEÑA-TAPIA E., GARZÓN-RAMOS D., JORGE DE LEÓN, GARZÓN M., JAIME DEL CERRO, 2019, *Multi-robot Systems, Virtual Reality and ROS: Developing a New Generation of Operator Interfaces*, *Koubaa A. (eds)*, Robot Operating System (ROS), Studies in Computational Intelligence, 778, 29–64.

[16] MODONI G.E., SACCO M., TERKAJ W., 2016, *A telemetry-driven approach to simulate data-intensive manufacturing processes*, 49th CIRP Conference on Manufacturing Systems, Stuttgart.

[17] KUTS V., MODONI G.E., TERKAJ W., TÄHEMAA T., SACCO M., OTTO T., 2017, *Exploiting Factory Telemetry to Support Virtual Reality Simulation in Robotics Cell*, Augmented Reality, Virtual Reality, and Computer Graphics, 1: 4th International Conference, *AVR 2017*, Ugento, Italy.

Publication VI

Kuts, V.; Modoni, Gianfranco E.; Otto, T.; Sacco, M.; Tähemaa, T.; Bondarenko, Y.; Wang, R. (2019). Synchronizing physical factory and its Digital Twin through an IIoT middleware: a case study. Proceedings of the Estonian Academy of Sciences, 2019.

Synchronizing physical factory and its Digital Twin through an IIoT middleware: a case study

Vladimir Kuts, [1 a *], Gianfranco E. Modoni[2], Tauno Otto[1], Marco Sacco[3], Toivo Tähemaa[1], Yevhen Bondarenko[1] and Ruxin Wang[1]

[1]Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering

[2]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing

National Research Council of Italy, Bari

[3]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing

National Research Council of Italy, Milano

[a]vladimir.kuts@taltech.ee

Keywords: Digital Twin, IIoT, Virtual Reality, Industrial Robots

Abstract

Digital Twin (DT) is the virtual clone of a factory that represents the latter's static and dynamic aspects (e.g., processes, systems, products, etc.) according to various granularities. Among the significant challenges that a manufacturing company has to face to implement the DT, one of the most demanding is the realization of an appropriate software infrastructure that enables the synchronization of the physical factory with its DT. In fact, in this case, the deep range of DT's capabilities can be exploited in its full potential. In particular, under these conditions, the DT can be used to enable various operations within the shop floor, such as to simulate and assess the factory performance. To support companies in addressing this challenge, this paper presents a potential solution based on the Industrial Internet of Things (IIoT) middleware that implements a fully dual-way synchronization between the real and virtual worlds. Also, the application possibilities of the solution are investigated within a real case study, conducted in TalTech Flexible Manufacturing System and robotics demo center of Tallinn, to demonstrate the approach correctness and validity.

Introduction

One of the key aspects for the success of a modern factory consists in the realization of its Digital Twin (DT), which is a virtual clone of its manufacturing system (or part of it) that represents the latter's static and dynamic aspects according to various granularities [1]. The creation of this virtual representation can be exploited to enable various strategic operations within the shop floor, such as to simulate and assess the factory performance. For example, the simulation environment provided by DT can allow production managers and factory automation system integrators to check the changes to be applied to a factory layout without the need to assess them in the production environment. Also, this environment is enhanced by virtual realities capabilities, which can allow users to enter the envisioned factory in Virtual Reality (VR) to inspect manufacturing systems in the real scale.

Authors of this research focus on a specific application of the DT for management and control of manufacturing systems mainly in the field of industrial robots. In particular, the idea behind the work is that, with the usage of VR tools, human presence can be simulated, accompanied, and enhanced by the presence of other operators also from remote sites (multiplayer capability). In previous studies of the robotics area, the simulations have been mainly performed for Virtual Commissioning (VC) to optimize robots' algorithms during the development phase. In the more general manufacturing field, the simulations have been exploited to represent the complex behavior of a system, also considering the possible consequences deriving from external factors like human interactions or design constraints.

The potential of the DT has been recently analyzed in various publications [1-4]. Schluse and Rossmann [2] described an idea of DT as "Versatile Simulation Database" (VSD), which is a static data container containing related algorithms/interfaces to manipulate the data. Moreover, it also provides functionalities for parallel and distributed simulation. For a specific use-case, Grinshpu and others [3] illustrated an application involving KUKA LWR4 robot using VEROSIM as a simulation environment. VEROSIM provides a scripting language called SOML++ where engineers can easily monitor force and torque occurring during the manufacturing simulation. DT is seen as a composition of different models and data by Schroeder [4], and it can be aggregated into an AutomationML model enhancing the exchange data between heterogeneous systems via a middleware named FIWARE [5].

However, according to numerous works in literature [6–8], a valid realization of the DT asks for the following key features:

a) the data acquisition via a monitoring system;

b) setting the granularity of the DT synchronization in order to make it efficient while meeting specific goals; c) a data model to represent the real factory and its evolution;

d) the synchronization process between the real and digital factory.

This paper focuses mainly on the last feature, proposing the idea that behind the synchronization between a robot and its DT there is a middleware application which propagates the data stream of the factory telemetry from the physical factory assets towards their digital representation and back. The presented work has also its root in and extends previous works of the same authors which concern an architectural infrastructure for the synchronization [7,8] and the industrial robot tests combining VR and DT [8,9,10]. Also, this work takes as reference other studies in similar fields and concerning remote laboratories [11], initial design [12] and virtual training [13]. The remainder of this paper is structured as follows. Section 2 presents the software infrastructure, while Section 3 illustrates its application and evaluation within a case study. Section 4 presents the benefits of an approach based on a synchronized DT. Finally, Section 5 concludes, summarizing the major findings.

A software infrastructure for the bidirectional synchronization of real and virtual assets

Modern manufacturing systems are projected into a new era in which they will be capable of exploiting the IIoT protocol in its full potential. Leveraging this potential, manufacturing systems will become dynamic networks comprising different kinds of physical and virtual resources. In particular, thanks to the advanced communication and networking capabilities provided by the IIoT, each production component connected to the IIoT network will be paired with its fully synchronized DT. To contribute to realizing this vision, this work envisions a software infrastructure to constantly synchronize each-other the physical components with their digital representations. This infrastructure leverages a message-oriented Industrial Internet of Things (IIoT) middleware based on a publish-subscribe interaction [14]. The middleware handles the factory telemetry, i.e., the real-time data stream which involves all data such as monitored controlled variables coming from the plant and actuator directives to be promptly executed on a physical resource. Thus, the middleware allows any (real or virtual) component connected to IIoT protocol to propagate significant information to keep other interested components (information consumers) updated about occurred events (e.g., the position change of a device). Middleware can filter the content of each message and dispatches the information updates only to interested resources, where information dispatching adopts an event-based PUSH strategy, thus announcing the information updates under the form of events. Thanks to the middleware, notifications can be distributed as messages so that the information consumers can take further evaluations and then perform some needed actions. Also, the middleware allows the interested components to specify their requests to be subscribed to updates related to specific contextual factory information. Specifically, each controller of the physical component (*PhysicalResourceController*), after having acquired and collected the monitored information, can share the context information with other components and in particular with the component that manages the DT (*DigitalTwinController*). In turn, the latter can send directives to the *PhysicalResourceController*s according to the input received within of the virtual environment. Under these conditions, *PhysicalResourceControllers* and *DigitalTwinController* play the role of prosumers, i.e., they can be both information producers and consumers.

The case study

A proof of concept of the software infrastructure to synchronize a real robot with its DT was implemented within the Flexible Manufacturing Systems (FMS) and Robotics Demo Centre and ProtoLab of the Tallinn University of Technology. In particular, the latter includes the Industrial Robotics section, which consists of two heavy load robots. One is Yaskawa Motoman GP8 with a changeable tool for picking and placing tasks. The herein proposed infrastructure was built focusing on this robot. In this regard, its virtual model was created through the Unity3D game engine platform. The visualization of this model requires the use of standard VR equipment (e.g., head-mounted display units such as Oculus Rift/HTC Vive).

For the re-creation of a real Industrial Robot, a cell was created with the aid of 3D modeling software, producing a virtual representation of the laboratory with all its assets (Fig. 1). The more precise process of creation of a precise digital model is described in previous research papers of the authors in all aspects of the technical side [9]. A short list of most essential aspects what should be considered while digitalizing manufacturing equipment, when are used own 3D models based on drawings or imported from equipment manufacturer's libraries follows:

- The rigging of the model –the central pivot points for movable objects should be appropriately defined.
- Rendering time – the 3D objects geometry should be simplified to improve the performance of the visual simulation. However, the mesh should be correct.
- For the accuracy – 1 to 1 scale should be re-checked and kept.
- Proper kinematic of the portable equipment.

Without consideration of this list – the accuracy of a DT will be not efficient for the experiments.

Figure 1. Digitalized laboratory – FMS and Robotics Demo Centre and ProtoLab (a, b, c – real; d, e, f – replica)

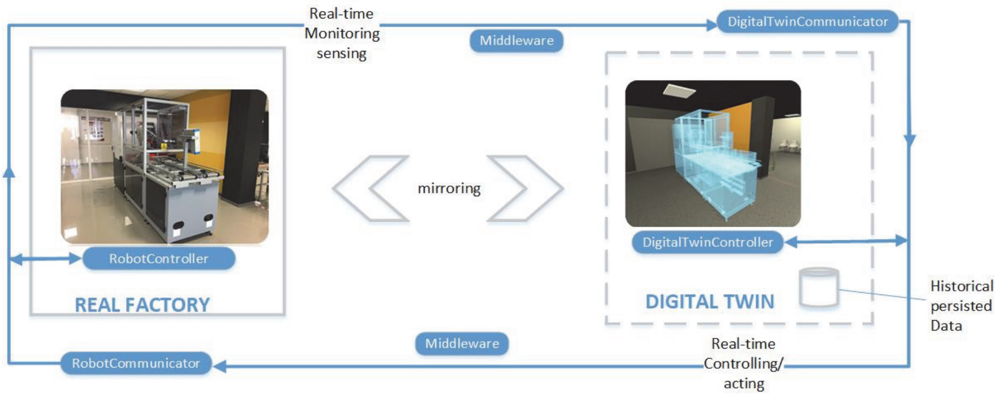


Figure 2. Synchronization process through the proposed software infrastructure

Figure 2 reports the five different components of the software infrastructure: a) *RobotController* which controls the robot through a specific API; b) *DigitalTwinController* which manages the DT model; c) a publish-subscribe middleware as introduced in the previous section; d) *RobotCommunicator* that handles the communication for the real component; e) *DigitalTwinCommunicator* that handles the communication for the virtual component. Also, Figure 1 represents how these components are linked to enable the synchronization process. In particular, the latter is enabled by the middleware and two of its clients (*RobotCommunicator* and *DigitalTwinCommunicator*). The role of middleware is in turn played by MQTT publisher/subscriber protocol, which is standardized by ISO [15] and is known for being a lightweight and fast solution for IoT applications. Another potential option for middleware could have been Kafka platform [16], which allows greater scalability and data persistence. However, MQTT was adopted for its ease of implementation and shorter setup time, while also providing the needed features for the considered case study. Also, current MQTT implementations provide countermeasures against various failure or error situations by offering features, which make middleware more resilient.

All messages published by the MQTT clients are sent to the MQTT broker, which then forwards the messages to other clients subscribed to the same messaging topic (the two *Monitors* are at the same time data publisher and consumer). The presented solution utilizes six messaging topic pairs, which correspond to control inputs and monitoring outputs of the *RobotListener* (i.e., robot position setting and monitoring). The MQTT broker was hosted on a public IP server managed by Digital Ocean [17] and located in Frankfurt (Germany). The real robot can be managed by running *DigitalTwinCommunicator* script which allows the robot to be connected using MQTT.

Another faced challenge was related to Unity. Indeed, since it does not feature MQTT support out of the box, it was necessary to implement a custom solution which would allow starting MQTT client scripts directly from Unity runtime. However, MQTT protocol is universal, and the clients for it were already implemented in many programming languages, including C# used by Unity. The final solution for this demo was based on .NET MQTT client implementation by Eclipse [18] and existing test Unity project available on GitHub [19].

An important reason for choosing a Unity3D game engine for development was also an option to build both VR-supported and classic desktop versions of the application. Though both of the versions feature the same DT control interface, VR build was also complemented with user avatar networking, allowing researchers to test the robot online while being together in the same virtual environment (Fig. 3). This feature allows demonstrating an important aspect of future connected factories: the ability to monitor equipment and collaborate in real-time, even from remote locations.

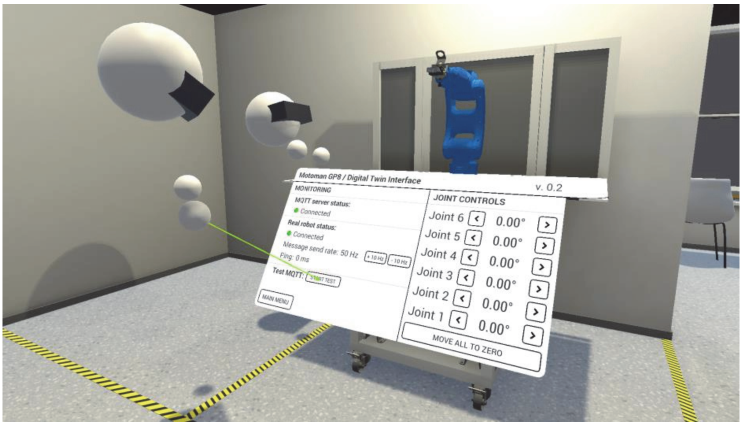


Figure 3. Researchers' avatars controlling the robot in VR

The aspects that can affect the success for such a real-time, multi-user, internet-based collaborative VR systems are the latency, i.e., the time needed to synchronize the virtual world with the real one and vice versa, the scalability and user experience. These aspects depend in turn on the number of the connected IIoT devices, the throughput of the information shared between the devices and the size of the 3D scene (nodes, geometries, assets). Thus, after the implementation, the solution was evaluated according to these three following criteria:

a) latency (the aim of the test is measuring it and demonstrating that it is under a fixed threshold);

b) capability to scale according to the throughput, which can be increased by acting on the status update frequency of the robot;

c) capability to control DT from remote locations.

To perform the above mentioned experiment, a test was performed by varying two parameters: robot send rate (how often *RobotListener* sends updates about the robot's position to the broker) and MQTT Quality of Service (QoS) level (it determines how reliably the messages are delivered by the MQTT protocol) Send rate values were taken in the range from 10 to 100 Hz, with a step of 10 Hz, while QoS level can take values of 0, 1, or 2, with greater values requiring more computation time for processing messages. Finally, the time required to send messages from *DigitalTwinCommunicator* to *RobotCommunicator* and back (ping) was taken as a metric of connection speed. For each combination of parameters, the robot was sent 50 movement commands, and ping was continuously sampled to calculate the average for each experiment. The test was run in two different places at premises of STIIMA-CNR (Bari, Italy) and of TalTech Industrial Virtual and Augmented Reality lab (IVAR Lab), which is part of Flexible Manufacturing Systems and robotics demo center. Data collected after running tests in these two locations can be seen in Figure 4.

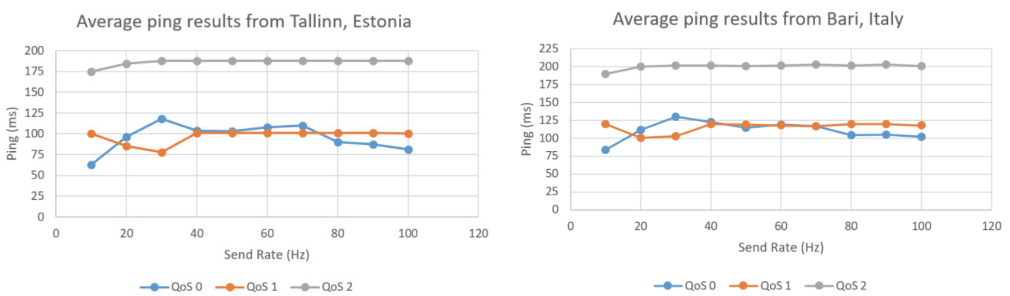


Figure 4. Experiment results

A significant result obtained from the experiment is the fact that the different values of the send rate used in the experiment do not affect the time of synchronization. The only parameter affecting the communication speed is QoS: rising level to 2 results in longer ping times (as the message processing time increases), while levels 0 and 1 provide approximately equal results. This proves that it is possible to maintain a fast and reliable connection even with high send rate and an acceptable QoS level.

Benefits of the proposed solution and future developments

A list of benefits deriving from the exploitation of the proposed infrastructure is presented in the following:

- Ability to work with the system from remote locations in real time, eliminating the need to come in close contact with dangerous equipment or hazard environments.
- Support for multiplayer mode and remote distance collaboration. IIoT-enabled devices can efficiently share their state with a multitude of clients, allowing them to create a single networked virtual environment where workers/ can directly communicate while managing the connected equipment. DT can support the training with all the equipment with which the training, in reality, is not feasible (e.g., too expensive, etc.).
- Simulation capabilities which can allow to "rewind" the state of the virtual system to an arbitrary point in time for identifying failures or analyzing system actions, exploiting both real data coming from the robot and historical data previously collected and then persisted on the database (a feature to be implemented). Moreover, these capabilities can allow creating virtual sensors which can send commands to the real machines in order to evaluate performance and track possible collisions with personnel or solid bodies.
- DT brings a new level of production efficiency. Indeed, instead of taking down the manufacturing line to apply some changes, everything can be tested and verified on the virtual clone, and then safely put into production. Human-robot interactions can be tested in the virtual environment in offline mode. Also, industrial equipment does not need to be stopped while re-programming which leads to time-saving.

Future developments have to address the efforts towards speed optimization of the synchronization and integration to the middleware of other assets also from new and existing different labs, also investigating the potential of Kafka in the role of middleware. Moreover, the DT will be paired with a database to persist the history of robot positions which can be then used to replay past robot movements (See Fig. 1) [20]. Finally, the overall infrastructure will be tested in other fields also different from manufacturing (e.g., telemedicine).

Conclusion

The result of this work consists of the implementation of IIoT middleware, which supports full dual-way synchronization between a real robot and its virtual counterpart. The experimental ready application showed efficient connectivity, and the researchers from STIIMA-CNR (Bari, Italy) was able to control and manage industrial robot situated in TalTech IVAR Lab (Estonia, Tallinn). Moreover, the work gave the opportunity for an international collaboration which is creating the basis for the integration of different laboratories and factory environments into a digital network.

Acknowledgments

This research was supported by project AR16077 Smart Industry Centre (SmartIC) No. 2014-2020.4.01.16-0183, supported by the EU Regional Development Fund. The authors are also grateful to the Computer and Systems Engineering student of the Tallinn University of Technology – Ninel Dudnyk, for her help in the experiments with laboratory digitalization.

References

[1] Modoni, G. E.; Caldarola, E. G.; Terkaj, W.; Sacco, M. (2018). Synchronizing Physical and Digital Factory: Benefits and Technical Challenges. Proceeding of the 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gulf of Naples, 2018.

[2] Schluse, Michael, and Juergen Rossmann. "From simulation to experimentable digital twins: simulation-based development and operation of complex technical systems." Systems Engineering (ISSE), 2016 *IEEE International Symposium on. IEEE,* 2016.

[3] Grinshpun, Georgij, et al. "From Virtual Testbeds to Real Lightweight Robots: Development and deployment of control algorithms for soft robots, with particular reference to." ISR 2016: *47st International Symposium on Robotics; Proceedings of. VDE*, 2016.

[4] Schroeder, Greyce N., et al. "Digital twin data modeling with automationML and a communication methodology for data exchange." IFAC-PapersOnLine 49.30 (2016): 12-17.

[5] Glikson, Alex. "Fi-ware: Core platform for future internet applications." Proceedings of the 4th annual international conference on systems and storage. 2011.

[6] Modoni, G. E.; Sacco, M.; Terkaj, W. (2016). A Telemetry-driven Approach to Simulate Data-intensive Manufacturing Processes, Procedia CIRP, Volume 57, Pages 281-285

[7] Kuts, V.; Modoni, G. E.; Terkaj, W.; Tähemaa, T.; Sacco, M.; Otto, T. (2017). Exploiting factory telemetry to support Virtual Reality simulation in robotics cell. Augmented Reality, Virtual Reality, and Computer Graphics, 1: 4th International Conference, AVR 2017. Springer, 212−221. (Lecture Notes in Computer Science)

[8] Kuts, V.; Sarkans, M.; Otto, T.; Tähemaa, T. (2017). Collaborative work between human and industrial robot in manufacturing by advanced safety monitoring system. In: B. Katalinic (Ed.). DAAAM International (0996−1001). Vienna, Austria: Curran Proceedings

[9] Kuts, V.; Otto, T.; Tähemaa, T.; Bondarenko, Y. (2019) Digital Twin based synchronised control and simulation of the industrial robotic cell using Virtual Reality. Journal of Machine Engineering, vol. 19, no. 1, pp. 128-145, 2019.

[10] Kuts, V.; Otto, T.; Tähemaa, T.; Bukhari, K.; Pataraia, T. (2018). Adaptive industrial robots using machine vision. Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition: IMECE2018, November 9-15, 2018, Pittsburgh, PA, USA. ASME

[11] Ji, Wei, Shubin Yin and Lihui Wang. "A Virtual Training Based Programming-Free Automatic Assembly Approach for Future Industry." *IEEE Access* 6 (2018): 43865-43873.

[12] Sell, R. (2013). Remote laboratory portal for robotic and embedded system experiments. International Journal of Online Engineering, 9(8), 2013, Pages 23-26

[13] Sell, R., Coatanéa, E., Christophe, F (2008). Important aspects of early design in mechatronic. Proceedings of the International Conference of DAAAM Baltic Industrial Eng

[14] Modoni, G. E., Veniero, M., Trombetta, A., Sacco, M., & Clemente, S. (2017). Semantic based events signaling for AAL systems. Journal of Ambient Intelligence and Humanized Computing, 9(5), 1311–1325.

[15] "ISO/IEC 20922:2016 Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1". iso.org. International Organization for Standardization. 2016.

[16] "Apache Kafka at GitHub". github.com. Retrieved 12 December 2018.

[17] "Digital Ocean cloud hosting platform". www.digitalocean.com. Retrieved 17 December 2018.

[18] "Eclipse Paho MQTT library for .NET at GitHub". github.com. Retrieved 16 December 2018.

[19] "M2MqttUnity at GitHub". github.com. Retrieved 16 December 2018.

[20] Modoni, G. E., Doukas, M., Terkaj, W., Sacco, M., & Mourtzis, D. (2017). Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models. International Journal of Computer Integrated Manufacturing, 30(10), 1043-1059.

Curriculum Vitae

Personal Data

Name: Vladimir Kuts
Date of Birth: 29.03.1989
Place of Birth: Estonia
Citizenship: Estonian

Contact Data

E-mail: vladimir.kuts@hotmail.com

Education

2015–… Tallinn University of Technology—PhD
2012–2015 MSC, Tallinn University of Technology
2008–2012 BSC, Tallinn University of Technology
2005–2008 High school, Tallinna Mustjõe Gümnaasium

Language Competence

English: Fluent, Estonian: Fluent, Russian: Fluent, Ukrainian: Average, Italian: Basic

Professional Employment

2018–Present IMECC Ltd, Development Engineer (0,50)
2017–Present Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering, Early Stage Researcher (1,00)
2017–2017 The Institute of Industrial Technologies and Automation (ITIA) of the National Research Council of Italy (CNR), Early Stage Researcher (1,00)
2017–2017 Tallinn University of Technology, School of Engineering, Department of Mechanical and Industrial Engineering, Engineer (1,00)
2016–2016 TUT Mektory Nanosatellite Programme, Mechanics Area Coordinator (0,20)
2016–2016 Tallinn University of Technology, Faculty of Mechanical Engineering, Department of Mechanical and Industrial Engineering, Chair of Production Engineering, Engineer (1,00)
2014–2016 Inducont OÜ, Sales and Application Engineer (1,00)
2012–2014 Ektaco AS, Technician (1,00)

R&D Related Managerial and Administrative Work

2018−Present Tallinn University of Technology, School of Engineering - Member of Doctoral Studies Curriculum Board of Mechanical Engineering
2018−Present TTÜ ÜLIÕPILASKÜLA MTÜ - Board Member
2017−Present Tallinn University of Technology - Member of Board of School of Engineering
2017−Present Tallinn University of Technology - Department of Mechanical and Industrial Engineering - Head of Industrial Virtual and Augmented Reality Lab (IVAR)
2018−2018 Koo-Met OÜ - Mechatronics, Level 4, Qualification Giving Expert Committee Member
2016−2018 Tallinn University of Technology - Member of Council
2016−2016 Tallinn University of Technology, Faculty of Mechanical Engineering - Chairman of Marketing Committee
2016−2016 Estonian Quality Agency for Higher and Vocational Education - Engineering, Manufacturing and Technology Area Evaluation Committee Member
2014−2015 Tallinn University of Technology - Member of Council
2014−2015 Tallinn University of Technology - Member of Board of Faculty of Mechanical Engineering

Elulookirjeldus

Isikuandmed

Nimi: Vladimir Kuts
Sünniaeg: 29.03.1989
Sünnikoht: Eesti
Kodakondsus: Eesti

Kontaktandmed

E-post: vladimir.kuts@hotmail.com

Hariduskäik

2015–… Tallinna Tehnikaülikool —PhD
2012–2015 MSC, Tallinna Tehnikaülikool
2008–2012 BSC, Tallinna Tehnikaülikool
2005–2008 High school, Tallinna Mustjõe Gümnaasium

Keelteoskus

Inglise keel – Kõrgtase, Eesti keel – kõrgtase, Vene keel – kõrgtase, Ukraina keel – kesktase, Itaalia keel – algtase

Teenistuskäik

2018–... OÜ IMECC, Arendusinsener (0,50)
2017–... Tallinna Tehnikaülikool, Inseneriteaduskond, Mehaanika ja tööstustehnika instituut , doktorant-nooremteadur (1,00)
2017–2017 The Institute of Industrial Technologies and Automation (ITIA) of the National Research Council of Italy (CNR), Nooremteadur (1,00)
2017–2017 Tallinna Tehnikaülikool, Inseneriteaduskond, Mehaanika ja tööstustehnika instituut , Insener (1,00)
2016–2016 TTÜ Mektory Satelliidiprogramm, Mehaanika valdkonna koordinaator (0,20)
2016–2016 Tallinna Tehnikaülikool, Mehaanikateaduskond, Mehaanika ja tööstustehnika instituut, Tootmistehnika õppetool, Insener (1,00)
2014–2016 Inducont OÜ, Müügi- ja rakendusinsener (1,00)
2012–2014 Ektaco AS, Hooldustehnik (1,00)

Teadusorganisatsiooniline ja -administratiivne tegevus

2018−... Tallinna Tehnikaülikool, Inseneriteaduskond - Doktoriõppe õppekava Mehhanotehnika programminõukoja liige
2018−... TTÜ ÜLIÕPILASKÜLA MTÜ - juhatuse liige
2017−... Tallinna Tehnikaülikool - Inseneriteaduskonna nõukogu liige
2017−... Tallinna Tehnikaülikool, Mehaanika ja tööstustehnika instituut - Tööstusliku virtuaal- ja liitreaalsuse labori IVAR juht
2018−2018 Koo-Met OÜ - Mehhatroonik, tase 4 kutseandmise ekspertkomisjoni liige
2016−2018 Tallinna Tehnikaülikool - Nõukogu liige
2016−2016 Tallinna Tehnikaülikool, Mehaanikateaduskond - Reklaamikomisjoni esimees
2016−2016 Eesti Kõrg- ja Kutsehariduse Kvaliteediagentuur - Tehnika, tootmine ja tehnoloogia valdkonna hindamiskomisjoni liige
2014−2015 Tallinna Tehnikaülikool - Nõukogu liige
2014−2015 Tallinna Tehnikaülikool - Mehaanikateaduskonna nõukogu liige