IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Polytechnic of Leiria
School of Technology and Management
Department of Electrical Engineering
Master in Electrical and Electronic Engineering

INDUSTRIAL DEVICE INTEGRATION AND VIRTUALIZATION FOR SMART FACTORIES

ANDRÉ FAGUNDES MARTINS

Leiria, November of 2020

IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Polytechnic of Leiria
School of Technology and Management
Department of Electrical Engineering
Master in Electrical and Electronic Engineering

INDUSTRIAL DEVICE INTEGRATION AND VIRTUALIZATION FOR SMART FACTORIES

ANDRÉ FAGUNDES MARTINS
2172502

Project realized under supervision of Doctor Hugo Filipe Costelha de Castro (hugo.costelha@ipleiria.pt) and co-supervision of Doctor Carlos Fernando Couceiro de Sousa Neves (carlos.neves@ipleiria.pt) and Doctor Luís Manuel Conde Bento (luis.conde@ipleiria.pt).

November of 2020, Leiria

ACKNOWLEDGEMENTS

During my master, and especially throughout the writing of this report, I have received a great deal of support and assistance.

I would first like to thank my supervisor, Doctor Hugo Filipe Costelha de Castro, for his constant support, guidance and commitment during these years. Also a special thanks to my co-supervisors Doctor Carlos Fernando Couceiro de Sousa Neves and Doctor Luís Manuel Conde Bento whose expertise was invaluable during the development of my research work.

A special thanks is also due to all other Professors from Polytechnic of Leiria that contributed to my research work, especially to Doctor Luís Miguel Ramos Perdigoto that put a lot of effort into the development of the practical component of "Intelligent Automation and Industry 4.0" subject from the Master in Electrical and Electronic Engineering, essential topics to this project.

I must express my profound gratitude to my parents and to my sister for providing me with unfailing support and continuous encouragement throughout my years as a student. This accomplishment would not have been possible without them. Thank you.

No less important, I would like to thank to all my friends and co-workers, who were of great support in helping me over my research problems, as well as providing happy distraction to rest my mind outside of my research.

The work presented in this report was developed within the context of the project Tooling4G - Advanced Tools for Smart Manufacturing, supported by Programa Operacional Competitividade e Internacionalização (POCI), Portugal 2020 and Fundo Europeu de Desenvolvimento Regional (FEDER). Last but not least, a special thanks to these funding bodies which made this project possible.

RESUMO

Dado o constante crescimento e modernização da indústria, várias tecnologias têm sido introduzidas no chão de fábrica, em particular no que diz respeito aos dispositivos industriais. Cada marca e modelo de dispositivo exigindo diferentes interfaces e protocolos de comunicação, pelo que a diversidade tecnológica torna a interligação automática com software de gestão de produção extremamente desafiante. No entanto, combinando tecnologias-chave como *Machine Monitoring*, *Digital Twin* e *Virtual Commissioning*, juntamente com um protocolo de comunicação completo como o OPC UA, é possível contribuir para a integração de dispositivos industriais no ambiente de uma *Smart Factory*.

Para atingir este objetivo, foram definidas metodologias de integração e desenvolvido um conjunto de ferramentas. Este conjunto de ferramentas, além de facilitar as tarefas de integração, também deve fazer parte de um ambiente virtual, compartilhando o mesmo modelo virtual, o *Digital Twin*, ao longo de todo o ciclo de vida do dispositivo industrial, nomeadamente no projeto, simulação, implementação e execução/monitorização/supervisão e, eventualmente, na fase de descomissionamento.

Um dos principais resultados deste trabalho é o desenvolvimento deste conjunto de ferramentas e metodologias num ambiente virtual, baseadas na comunicação OPC UA e com o *Digital Twin* implementado utilizando o RobotStudio para ter suporte completo no ciclo de vida de um dispositivo industrial, desde as fases de projeto e simulação, até à monitorização e fases de supervisão, adequadas para integração em fábricas da Indústria 4.0. Para avaliar o funcionamento destas ferramentas, foram realizados testes para um cenário de teste com diferentes dispositivos.

Outro resultado relevante está relacionado com a integração de um dispositivo industrial específico - máquina CNC. Dada a variedade de sistemas de monitorização e protocolos de comunicação existentes, é feita uma abordagem onde várias soluções disponíveis no mercado são combinadas num único sistema. Esse tipo de solução *all-in-one* daria aos gestores de produção acesso às informações necessárias para uma monitorização contínua, contribuindo para a melhoria de todo o processo produtivo.

Palavras-chave: Indústria 4.0, Virtualização, Gémeo Digital, Comissionamento Virtual, Monitorização de Máquinas, OPC UA.

ABSTRACT

Given the constant industry growth and modernization, several technologies have been introduced in the shop floor, in particular regarding industrial devices. Each device brand and model usually requires different interfaces and communication protocols, a technological diversity which renders the automatic interconnection with production management software extremely challenging. However, combining key technologies such as machine monitoring, digital twin and virtual commissioning, along with a complete communication protocol like OPC UA, it is possible to contribute towards industrial device integration on a *Smart Factory* environment.

To achieve this goal, several methodologies and a set of tools were defined. This set of tools, as well as facilitating the integration tasks, should also be part of a virtual engineering environment, sharing the same virtual model, the digital twin, through the complete lifecycle of the industrial device, namely the project, simulation, implementation and execution/monitoring/supervision and, eventually, decommissioning phases.

A key result of this work is the development of a set of virtual engineering tools and methodologies based on OPC UA communication, with the digital twin implemented using RobotStudio, in order to accomplish the complete lifecycle support of an industrial device, from the project and simulation phases, to monitoring and supervision, suitable for integration in Industry 4.0 factories. To evaluate the operation of the developed set of tools, experiments were performed for a test scenario with different devices.

Other relevant result is related with the integration of a specific industrial device – CNC machining equipment. Given the variety of monitoring systems and communication protocols, an approach where various solutions available on the market are combined on a single system is followed. These kinds of all-in-one solutions would give production managers access to the information necessary for a continuous monitoring and improvement of the entire production process.

Keywords: Industry 4.0, Virtualization, Digital Twin, Virtual Commissioning, Machine Monitoring, OPC UA.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface.
CNC	Computer Numeric Control.
CPS	Cyber-Physical Systems.
CS	Companion Specifications.
DT	Digital Twin.
ERP	Enterprise Resource Planning.
MES	Manufacturing Execution Systems.
OPC	Open Platform Communications.
PLC	Programmable Logic Controller.
PLM	Product Lifecycle Management.
PubSub	Publish-Subscribe.
RAMI 4.0	Reference Architecture Model Industrie 4.0.
RS	RobotStudio.
SC	SmartComponent.
SDK	Software Development Kit.
UA	Unified Architecture.
VC	Virtual Commissioning.
XML	Extensible Markup Language.

1

INTRODUCTION

The so called shop floor digitalization is the basis of all developments considered within the scope of the 4$^{\text{th}}$ industrial revolution. In fact, the knowledge about the process, preferably obtained automatically, is the base of the architecture of a "Smart Factory" (Zhong et al., 2017). The work presented in this report aims for the development of a set of tools to allow industrial device integration and virtualization for Smart Factories in the context of Industry 4.0.

To achieve this goal, some key technologies were studied and methodologies were proposed. One of the main contributions is the developed methodology to build monitoring solutions for machining equipment, considering the most common equipment brands and operations used in the molds industry companies in one of the Portuguese machining clusters (known worldwide[1]). The developed methodology is based on Open Platform Communications (OPC) Unified Architecture (UA) for high-level communication between the various systems. The virtualization and Digital Twin (DT) are also key technologies that were studied, having in mind the need to have a virtual model with broader use through the complete lifecycle of an automated system.

Throughout the development of this work, three papers were published, namely:

- *Developing an OPC UA Server for CNC Machines* (André Martins, Lucas, et al., 2020) – This paper addresses the concept of Industry 4.0 from the perspective of the molds industry, a key industry in today's industrial panorama. With the constant modernization of this industry, several technologies have been introduced in the shop floor, in particular regarding machining equipment. Given the variety of monitoring systems and communication protocols of these equipment's, the developed approach combines different machine interfaces in all-in-one solution, in order to cover a relevant subset of machining equipment currently in use by the molds industry.
- *Supporting the design, commissioning and supervision of smart factory components through their digital twin* (Andre Martins et al., 2020) – This work

[1] https://www.moldmakingtechnology.com/blog/post/state-of-mold-manufacturing-in-portugal

explores the use of ABB RobotStudio (RS) software combined with OPC UA standard in order to achieve a DT with a broader use through the complete lifecycle of an automated system, namely the project, simulation, implementation, and execution/monitoring/supervision and, eventually decommissioning phases. Methodologies are defined to integrate both new generation and legacy equipment, as well as robot controllers and guidelines for equipment development.

- *Shop Floor Virtualization and Industry 4.0* (André Martins, Costelha, and Neves, 2019) – Describes the virtualization of a typical production process, the digital twin in the scope of Industry 4.0, involving different devices, such as robotic arms, conveyors, automatic warehouses and vision systems. Including both legacy and recent equipment, with different characteristics and communication capabilities. The developed work aims at industrial implementations, while allowing for educational purposes, using OPC UA protocol as high-level communication for a standardized approach.

1.1 MOTIVATION

Nowadays, many industrial sectors use some type of direct data collection in their industrial processes. However, neither its application is universal, nor the level of adhesion from companies is similar and the depth of existing implementations is uniform (Giessbauer et al., 2018). The best examples are in industries with automatic assembly lines, such as electronics and automotive industries, areas which are extensively automated and where process data, therefore, is more easily available (Giessbauer et al., 2018).

This availability stems directly from the presence of specific instrumentation needed for the automation of the process. Moreover, more useful data, possibly at a higher-level of abstraction, may be generated during their regular operation.

Automated manufacturing as long since been a part of the molds industry, namely in Computer Numeric Control (CNC) technologies, such in milling, turning and electric discharge machining, with the connection to the machines for program transmission from machining software being largely semi-automated. In the reverse direction, i.e., collecting information about the process or equipment states, the current level of automation is still very low, being neither universal nor done with the same depth-level in most companies (Neves et al., 2018).

The gap referred above exists mainly due to the diversity of CNC controllers on the market, and to the current lack of a *de facto* standard for machining interfaces, worsened by the fact that significant part of the machining equipment is either closed to third-party applications, or offers only a vendor-specific proprietary interface. All of the above combined, translates into an substantial difficulty for automated machine data collection.

Additionally, in a context of greater complexity of Smart Factories, the commissioning time for automated systems needs to be shortened. The use of Virtual Commissioning (VC) tools, like DT, is a good contribution to achieve this goal. A growing number of companies have recently shown an interest in this technology (VC), mostly related with Industry 4.0, as it reduces the time and cost of introducing new products and allows different scenarios to be evaluated and validated with the manufacturing controllers in the virtual environments, prior to the physical commissioning (Riera and Vigário, 2017).

1.2 GOALS

The main goal of this work is to develop a set of tools and methodologies to integrate industrial devices in Smart Factories environments, as well as the virtualization and the development of a virtual model of such industrial devices, which can be used to predict events and to test what-if scenarios without the loss of productivity to the companies.

1.3 DOCUMENT STRUCTURE

The remainder of this project report is organized as follow: This first chapter defines the field, the motivations and the problem to approach. Besides that, a description of the main goals of the research work, as well as the main contributions are provided. Chapter 2 provides the reader with a description of the fundamental concepts that are the basis of this work and a literature review on those subjects. Chapter 3 describes the developed work in what concerns CNC machine monitoring. Chapter 4 describes the developed work in virtualization, digital twin and virtual commissioning. Chapter 5 presents the tests and results comparing both laboratory and industrial environment tests. Finally, in Chapter 6, the main conclusions of the work are presented and some paths for future work are suggested.

2 FUNDAMENTAL CONCEPTS AND RELATED WORK

In this chapter, relevant concepts and terminology are reviewed as well as some related work on these topics, which are significant for the embodiment of subsequent chapters. Firstly, Industry 4.0, the main concept of this work, is defined and its relation with the previous industrial revolutions is reviewed. In the scope of Industry 4.0, some key features are commonly referred and virtualization is one of them. This concept ranges from simpler approach, where only a set of data is gathered and made available to the virtual world, to the realistic DT of an industrial system. Another concept widely mentioned is the VC, the core advantage of it being the reduced physical commissioning time and the additionally uses that can be given to the developed DT throughout the real system life cycle. In order to achieve the ambitions of Industry 4.0 and all its key features, data needs to be collected from the machines on the industrial processes. In this report, Machine Monitoring in the specific case of CNC milling machines is approached. OPC UA standard is the main supporting technology of the development, due to its characteristics of open connectivity, interoperability, built-in security, scalability, compliance and communication mechanisms, characteristics that are inline with Industry 4.0 ambitions.

2.1 INDUSTRY 4.0

The first industrial revolution, appears in the end of the XVIII century, with the mechanization of manufacturing, first using water as a power source, and then with the use of the steam engine, the latter being one of the most important developments of that era.

The second step up, called the second industrial revolution, starts in the very end of the XIX century, with the introduction of the work division and the assembly lines associated to industrial electrification.

The third revolution appears largely with the adoption of automation in industries, between the 60s and 70s decades from the last century, and considerably increased the productivity and quality of the manufacturing process and manufactured products, respectively.

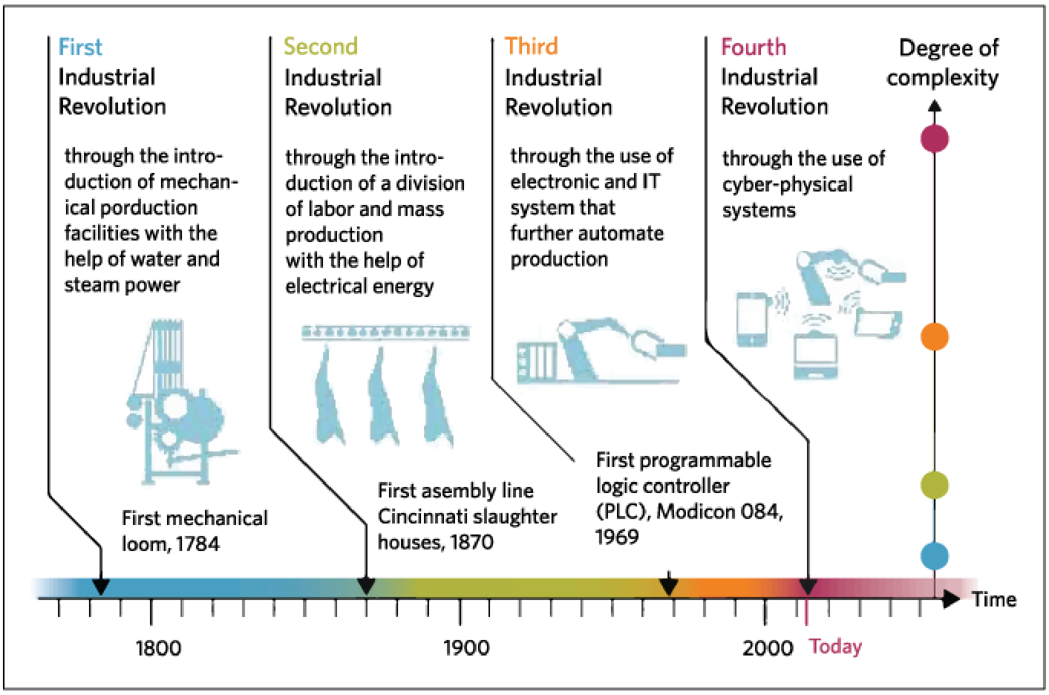

First
Industrial
Revolution
through the introduction of mechanical porduction facilities with the help of water and steam power
Second
Industrial
Revolution
through the introduction of a division of labor and mass production with the help of electrical energy
Third
Industrial
Revolution
through the use of electronic and IT system that further automate production
Fourth
Industrial
Revolution
through the use of cyber-physical systems
Degree of complexity
First mechanical loom, 1784
First asembly line Cincinnati slaughter houses, 1870
First programmable logic controller (PLC), Modicon 084, 1969
Time
1800
1900
2000
Today

Figure 1: Industrial revolutions.[1]

Nowadays, we identify the 4th industrial revolution as the interconnection of the production systems by information networks, as shown in Figure 1. Although, there are some people who prefers the use of "evolution" instead of "revolution", stating that the transformation is gradual and not immediate (Bartodziej, 2017). In spite of this, the paradigm in industrial production is changing.

Pressed by the need for higher production flexibility, stemmed from the shorter production series brought by massive customization, where the customer more and more establishes product characteristics, economy agents, such as sector associations, higher education institutions, research institutes, governments and industry are developing initiatives to simultaneously disseminate and normalize this new paradigm.

2.2 VIRTUALIZATION

Within the scope of the 4th industrial revolution, virtualization is one of the key features to attain the embodiment of the concept of Cyber-Physical Systems (CPS)

[1] http://nearyou.imeche.org/eventdetail?id=12995

in industry. Virtualization allows significant time reductions in implementing changes within the shop floor, adding the possibility to test realistic what-if scenarios of shop floor configurations, production organization and scheduling or maintenance prediction, impossible to test otherwise without stopping production (Straus et al., 2018). The virtualization of an industrial system encompasses several components, ranging from a simpler approach, where only a set of data is gathered and made available to the virtual world, to very complete simulators, allowing a physically, computationally and graphically realistic DT of an industrial system or process (Tao and Zhang, 2017).

There is a significant body of knowledge related with the virtualization of industrial shop floor situations as well as about the design of applications for interconnecting industrial devices. Regarding virtualization, there are many robot simulators (Staranowicz and Mariottini, 2011) (Pitonakova et al., 2018), e.g., Gazebo, Tecnomatix Process Simulate, V-REP, Visual Components and RS.

The open-source nature of Gazebo is an advantage. However, significant part of the applications and solutions available with this tool are related with mobile robotics and not as much with industrial (robotics) scenarios (Koenig and Howard, 2005). Among the many related examples of virtualization based on this simulator, the one by Vrabič et al. (Vrabič et al., 2018) shows the development of a differential drive mobile robot used as a test bed for a DT research where the motion model is implemented trough Gazebo. Here, the effort needed for model integration is identified as a drawback of this simulator.

In the case of Tecnomatix Process Simulate, owned by Siemens, the connection to Programmable Logic Controller (PLC) and other devices of their brand can be as close to plug-and-play as possible, but it does not have a true physics engine (Harrison and Proctor, 2015). Arjoni et al. (Arjoni et al., 2018) describe a retrofit process of manufacturing equipment existing in an academic laboratory environment, using Process Simulate to commission the virtual plant.

Visual Components, recently acquired by KUKA, provides for a generic simulation environment, with an easy learning curve, but also does not include physics simulation capabilities. Both Visual Components and Process Simulate do not include a true physics engine, making the simulations less realistic.

V-REP includes more industrial robotic arms related scenarios when compared with Gazebo but, like the other tools mentioned above, it does not support using vendors specific programming languages which, in practice, means the robotic arm programs are more limited, and not immediately transferable to the real robots

(Visual Components includes converters for vendors specific languages, but only for a small set of instructions).

RS provides a simulation environment with some similarities with Visual Components, but specific for ABB robots. Nevertheless, it is capable of simulating many components present on today's factories by importing their 3D models and adding a dynamic and responsive behaviour, which can be extended by the user through the available RS Software Development Kit (SDK)[2], a well documented feature reach tool. Furthermore, besides having recently added physics simulation capabilities, it allows for the complete set of RAPID (ABB robots programming language) instructions to be programmed and tested, albeit only limited with ABB robots (additional non-ABB robots could be added, but would required a considerable amount of work and would not provide the same degree of compatibility as with the ABB robots). Forgó et al. (Forgó et al., 2018) address the benefits of using ABB RS software with the use of SmartComponent (SC), which can interact with the robot virtual controller as inputs and outputs, from and to the simulated process.

2.3 DIGITAL TWIN

One of the concepts of Industry 4.0 with many uses in Smart Factories, is the DT, referred to for the first time by Michael Grieves in 2002, during a presentation on Product Lifecycle Management (PLM) (Grieves and Vickers, 2016). After that, the National Aeronautics and Space Administration (NASA) and the US Air Force applied DT in their maintenance and for the remaining useful life prediction of aerospace vehicles (Tao, Qi, et al., 2019). The DT is the virtual and computerized counterpart of a physical system, which can also be divided into three subcategories, according to their level of data integration: Digital Model, Digital Shadow and DT (Kritzinger et al., 2018).

Ayani et al. (Ayani et al., 2018) presented an industrial application where emulation tools are used to create a DT of an old industrial device to support the reconditioning project. They propose a methodology that tries to maximize the uses of the built DT in several phases of the system's lifecycle, in order to make the most of the required extra time consumed to build it, i.e. employing DT in other phases rather than restricting its use to the design, development and simulation.

Concerning the use of virtual engineering to achieve lifecycle support of automated systems, Konstantinov et al. (Konstantinov et al., 2017) demonstrate the use of

[2] https://developercenter.robotstudio.com/robotstudio-sdk

CPS-enabled virtual engineering tools within a practical workflow to complement existing engineering tools and methods, using a real industrial scenario as a testbed. This work demonstrated the use of the VueOne[3] toolset up to the build phase of a machine using a common model. The authors mentioned that their future work will show how this model can be truly exploited throughout product lifecycle. Jbair et al. (Jbair et al., 2019) also used the VueOne toolset for automatic PLC structured code generation, based on a virtual engineering model and available process planing data, together with OPC UA communication blocks, aiming to obtain the connected DT and gain other benefits, such as data analytics.

2.4 VIRTUAL COMMISSIONING

VC is typically viewed as a Hardware-in-the-Loop (HiL) configuration that is used before real (physical) commissioning of an industrial device or plant, with the purpose of testing and verifying the control logic. However, there are other related concepts, such as Hybrid Commissioning, where the real controller is simultaneously connected to the real industrial equipment and to the simulation model. The core advantage of carrying out VC is the reduced physical commissioning time. Additionally, an accurate simulation model, the so-called DT, which can be used throughout the real system's life span, is created. (Khan et al., 2018).

Fernandez et al. (Fernández et al., 2019) use the Simumatik3D[4] software to achieve the VC of PLC programs for a robotic cell in an academic environment. This software is presented as a recently launched open emulation platform, which allows the development of digital models of industrial equipment to support the entire lifecycle of automated solutions. The authors suggested, as future work, the use of this technology in other phases of the engineering lifecycle, not only in the validation of PLC programs.

The Factory I/O[5] software, from Real Games, is described by Riera et al. (Riera and Vigário, 2017) as a new generation of 3D factory simulation for learning automation technologies. The highly interactive environment, which allows the introduction of disturbances on the controlled plant and faults in sensors and actuators, is identified as an important feature of this software. The authors mentioned that, at the time they wrote the paper, the VC use of Factory I/O for pedagogical applications

[3] https://warwick.ac.uk/fac/sci/wmg/research/digital/automation/
[4] https://www.simumatik.com/
[5] https://factoryio.com/

had been successfully tested. Here, the impossibility of importing or creating new 3D models into Factory I/O is identified as a drawback.

A growing number of companies have recently shown an interest in this technology, mostly related with Industry 4.0, as it reduces the time and cost of introducing new products and allows different scenarios to be evaluated and validated with the manufacturing controllers in the virtual environments, prior to the physical commissioning (Riera and Vigário, 2017). VC can be seen as part of the PLM, which is an integrated driven approach to all aspects of a product, from its design inception through its manufacture, deployment, maintenance and end-of-life.

2.5 MACHINE MONITORING

Nowadays, many industrial sectors use some type of direct data collection in the industrial process. However, neither its application is universal, nor the level of adhesion from companies is similar and the depth of existing implementations is uniform. The best examples are in industries with automatic assembly lines, such as electronics or automotive industries, areas in which automation is already very extensive and where process data, therefore, is easily available (Giessbauer et al., 2018).

The molds industry has been a pioneer in automatic manufacturing, namely in CNC technologies, typically in milling, turning and electric discharge machining, and the problems with the connection to the machines has been solved regarding the transmission of the programs from machining software. In the other direction, i.e. in the collection of information about the state of the process or the equipment, the state of the development is poor, being neither universal, nor done with the same depth in all companies.

There is significant research related to CNC machines monitoring techniques, especially with tool condition monitoring. For instance, in the work carried by Downey et al. (Downey, Bombiński, et al., 2015) (Downey, O'Sullivan, et al., 2016), the authors collects and analyses data from three sensor technologies (force, acoustic and vibration) in a real time production environment, to monitor CNC tool wear. Other works, like the one by René de Jesús et al. (René de Jesús et al., 2003) and the one by Stavropoulos et al. (Stavropoulos et al., 2016), uses driver current signal analysis on tool wear and breakage detection.

Concerning the use of OPC UA standard on monitoring CNC machines over a wider range of parameters, there are fewer implementations and results available, of which the two most relevant are described here. The work from Mourtziz et al. (Mourtzis et al., 2018) proposes an OPC UA-based framework for the modelling of milling and lathe CNC machine-tools. They also developed a data acquisition device to allow the integration of legacy machine-tools, without connectivity capabilities, in their holistic framework, presenting a laboratory case study to validate the proposed system. Liu et al. (Liu et al., 2019) propose a Cyber-Physical Machine Tools (CPMT) platform based on OPC UA and MTConnect to enable standardized, interoperable and efficient data communication between machine tools and various types of software applications. To demonstrate the advantages of the proposed CPMT platform, different applications were developed, including an OPC UA client, augmented reality assisted wearable Human-Machine Interface, and a conceptual framework for a CPMT-powered cloud manufacturing environment. These two works have their major developments around data acquisition using external sensors, giving less relevance to the direct data exchange with the CNC machine controllers.

2.6 OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE

The main supporting technology of the developed work is the OPC UA standard. In this section, a brief description and the main characteristics of this standard are presented.

Defined by the international standard IEC 62541, OPC UA is the approach for a communication layer implementation recommended by the German Reference Architecture Model Industrie 4.0 (RAMI 4.0)[7] industrial platform (De Melo and Godoy, 2019). OPC UA is a standard that, due to its characteristics, ensures open connectivity, interoperability, security, scalability and compliance of industrial devices and systems. Based on a multi-layered architecture (see Figure 2), where a single server supplies all the information and services with active security, from a given system, OPC UA is much more than a protocol. It has built-in information models and defines basic rules for information exchange and interfaces.

[6] https://opcfoundation.org/about/opc-technologies/opc-ua/

[7] https://www.plattform-i40.de/

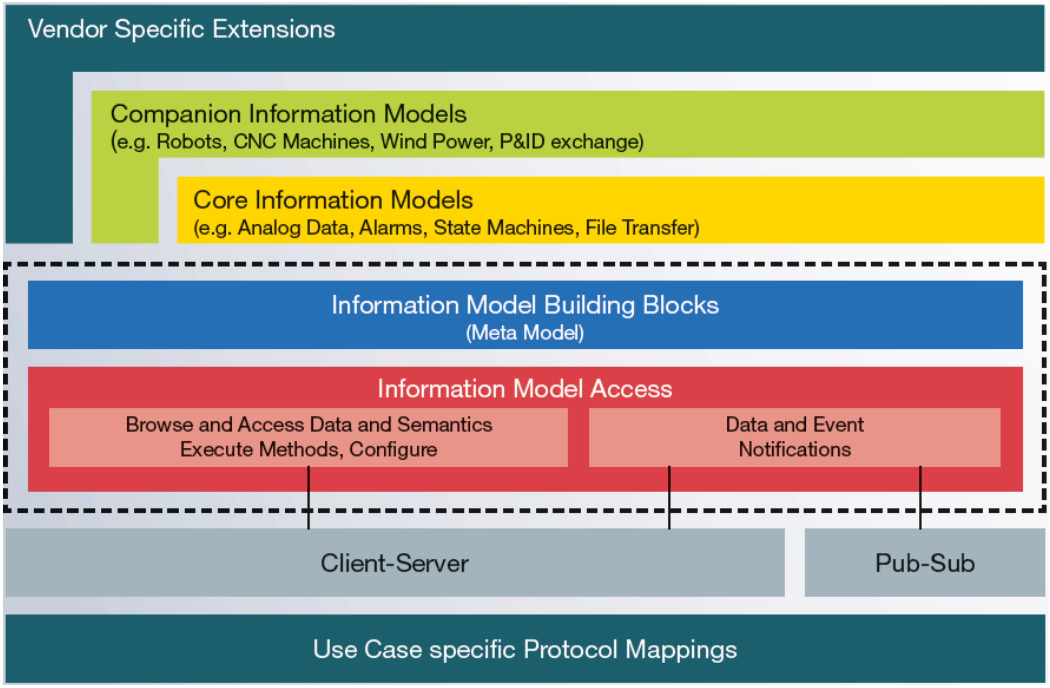

Vendor Specific Extensions
Companion Information Models
(e.g. Robots, CNC Machines, Wind Power, P&ID exchange)
Core Information Models
(e.g. Analog Data, Alarms, State Machines, File Transfer)
Information Model Building Blocks
(Meta Model)
Information Model Access
Browse and Access Data and Semantics
Execute Methods, Configure
Data and Event
Notifications
Client-Server
Pub-Sub
Use Case specific Protocol Mappings

Figure 2: OPC UA Architecture.[6]

2.6.1 *Open connectivity*

OPC UA comes as a successor of OPC Classic[8], but supporting multiple operating systems. OPC Classic can be configured to provide a fair amount of security, but by relying on Microsoft Windows and DCOM/COM functionalities, it is difficult to setup in other operating systems. On the other hand, OPC UA was designed to work in multiple environments and on multiple platforms, with security built into the platform from the start (Hunkar, 2014).

2.6.2 *Interoperability*

In order to allow interoperability, OPC UA uses the Address Space concept to represent all the information generated from a specific system or device. This data is usable and can be interpreted by other systems using the same protocol, with its structure being built on information models, resulting on a standardized way to represent all data and information to be accessible by client systems. OPC UA uses the concept of object to represent system data and behavior. Objects serve as

[8] https://opcfoundation.org/about/opc-technologies/opc-classic/

locations to store variables, events and methods, being also able to connect to each other using references.

The standard information model is designed to allow type definition so that designers can meet their own application needs and, on top of this, information models can also be combined for specific areas, such as CNC systems (VDW and OPC Foundation, 2017), Robotics (VDMA and OPC Foundation, 2019b) and Machine Vision (VDMA and OPC Foundation, 2019a), to name just a few. These specific models are called Companion Specifications (CS) and derive from the standard model, inheriting its features, but can also include modifications, which can also be combined, depending on the system's needs. The released CS can be found on OPC Foundation official GitHub repository[9]. In the development of these CS, some of the reference companies in the automation and robotics fields such as ABB, KUKA, or SIEMENS were actively involved, which underlines the importance of this movement.

2.6.3 *Security*

OPC UA gives a lot of importance to the secure communication between client and server, addressing a broad range of applications, and encompassing different security and timing requirements. The available security modes, shown in Table 1, can enable both digital signature and encryption mechanisms, only digital signature mechanisms, or none of them. An OPC UA client can choose the desired security mechanisms from the ones available by a specific OPC UA server. After a secure channel (and session) has been established between server and client, there is still the need for user credentials represented by another certificate, or by a username/password combination (Cavalieri et al., 2019). Bearing in mind that most of the OPC UA applications are within industrial environments, security issues should not be forgotten, as cyberattacks to this environments may compromise their normal operation. The OPC UA security analysis (Sicherheit in der Informaationstechnik, 2017), commissioned by the German Federal Office for Information Security, presents recommended measures and procedures that should be followed when implementing OPC UA-based solutions.

[9] https://github.com/OPCFoundation/UA-Nodeset

Table 1: OPC UA security modes (Neu et al., 2019).

Mode	Properties
None	No security.
Sign	Encoded with sender's private key. Only certificate owner has the private key. Anyone can verify the identity. Provides authenticity.
SignAndEncrypt	Adds encryption to sign. Encoding with receiver's public key. Anyone can encrypt. Only the certificate owner can read. Authenticity, confidentiality and integrity.

2.6.4 *Scalability and Compliance*

Various OPC UA profiles are defined, as show in Table 2, with different application scenarios in mind, allowing OPC UA to scale down to a chip level, using the Nano Embedded Device profile, while still retaining its prominent features (Imtiaz and Jasperneite, 2013). On the opposite side, when the implementation of complex scenarios is required, a PC-based server using the Standard UA profile could be the adequate solution. The OPC Foundation developed the UA Compliance Test Tool (UACTT)[10], as a way to identify which profiles are supported by our application, and to verify if the designed OPC UA server or client is compliant with the OPC UA specification.

2.6.5 *Communication Mechanisms*

For information exchange, OPC UA offers two main communication mechanisms: Client-Server and Publish-Subscribe (PubSub) (Drahos et al., 2018). In the Client-Server mechanism, the client accesses the information provided by the server through defined services. The PubSub mechanism can be applied in different ways: for messaging over local area networks (LAN), data is published by the OPC UA server (publisher) and consumed by multiple authorized OPC UA clients (subscribers), using Time Sensitive Networking (TSN) to allow real-time low delay communications; for messaging over global networks (WAN/Cloud), OPC UA

[10] https://opcfoundation.org/developer-tools/certification-test-tools/opc-ua-compliance-test-tool-uactt/

Table 2: OPC UA profiles (VDMA and Fraunhofer IOSB-INA, 2017).

Profile	Characteristics
Nano Embedded Device	Limited functionality only, for very small devices, e.g. sensors. Only one connection, but without UA security, no subscriptions and no method calls possible.
Micro Embedded Device	Restricted functionality, at least two parallel connections, additional subscriptions and data monitoring, but no UA security and no method calls.
Embedded UA	Basic functionalities of OPC UA are available plus UA security and method calls.
Standard UA	Includes all functionalities for secure information access including UA security. No alarms and no history. PC-based servers should support at least this profile.

PubSub specification (OPC Foundation, 2018) defines mappings on protocols such as, for instance, MQTT[11].

Regarding the use of OPC UA, Ayatollahi et al. (Ayatollahi et al., 2013) created an information model and a prototype OPC UA Server using it for data acquisition, event generation and remote control of a machine tending industrial robot. At the time, they identified a problem related with the insufficient standard communication interfaces which, even with the introduction of the OPC UA standard, still lacked a generally accepted semantic description of devices used in the manufacturing plant. Since then, some progress was made with the recent releases of several CS by the OPC foundation in collaboration with important industry partners, which introduces standard information models to describe these devices.

The automotive manufacturer, Renault, has also adopted OPC UA as their company standard for machine-to-machine communications. This company will both retrofit existing machines, as well as purchase new OPC UA-capable machines to establish an industrial data management platform based on Google Cloud. This will also allow Renault to standardize the data model describing their operations globally.[12]

[11] http://mqtt.org/

[12] https://youtu.be/WwR_6VHuN4Q

3

CNC MACHINES MONITORING

This chapter describes the methodology developed for the implementation of a CNC monitoring solution based on an OPC UA server. The development of a generic C# class is explained, as well as the chosen approaches for data exchange with the CNC machines. Details of the implementations referred on this chapter are further described on a technical manual (André Martins, Costelha, Neves, and Bento, 2020a) and a user manual (André Martins, Costelha, Neves, and Bento, 2020b) that were created within this work.

With the current market diversity of CNC controllers, and given that some of this equipment is not open to third-party applications, the direct interaction with the controller becomes extremely difficult. On the other hand, more recent equipment often has an available communication protocol, and some even have an SDK, an Application Programming Interface (API), or simply the definition of a group of messages that can be exchanged with the equipment to this end. Alternatively to the acquisition methods using the information provided by the CNC controllers, direct access to the sensors already integrated, or externally added to the CNC, can also contribute for the collection of performance metrics, which in turn will allow the prediction of some machine states and/or events.

3.1 GENERIC C# CLASS

In order to accommodate all this variety of paradigms, a generic C# abstract class (CNCBase) with abstract methods was developed to help modeling a generic interface for any CNC machine, allowing the development of a monitoring application which can work with different CNC machines. With this type of implementation, each CNC machine specific brand then overrides the generic class with its specific functions according to its specific communication interfaces, when needed, but exchanging the information available using the same information model (see Figure 3).

This kind of implementation makes the future incorporation of other CNC machine models more straightforward. Another relevant component of our application is

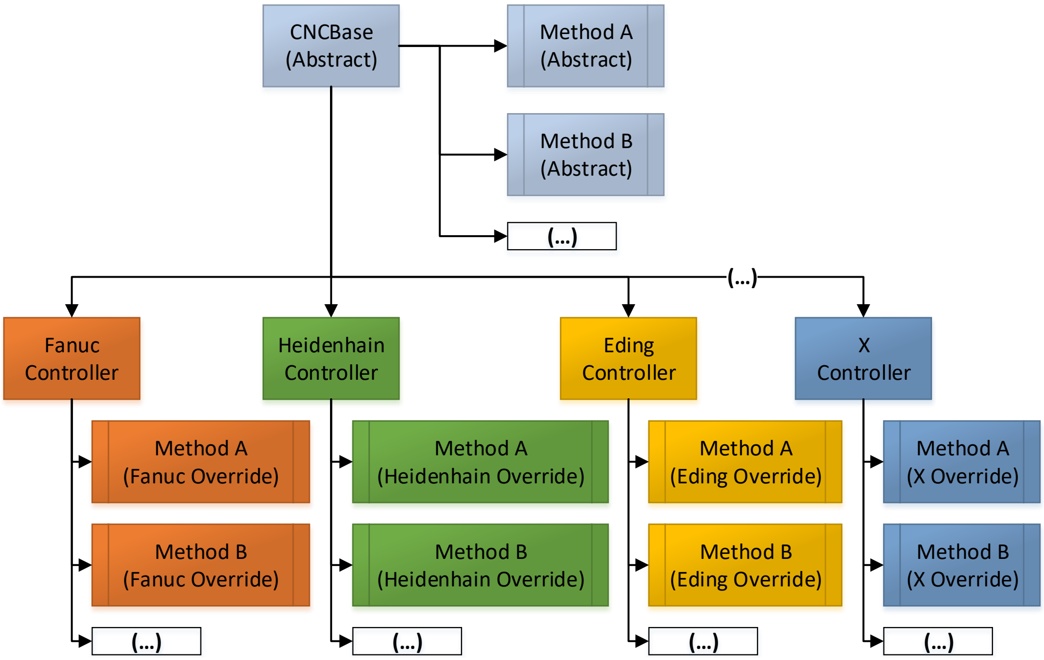


Figure 3: Generic C# abstract class and methods.

a configuration file, which allows the definition of some application parameters, such as the CNC machine model/brand to be monitored, the communication address definition to be used, as well as the machine parameters and frequency to be monitored. The main goal of this file is the definition of run-time function parameters, i.e., without the need to recompile the server application for specific machines, or, for instance, whenever different data is to be acquired, or is to be acquired at different sampling rates.

Entries in this file respect the following syntax:

`<add key="Name" value="NodeID,Enabled,Time"/>` where

- `Name` – Method name;
- `NodeID` – Node number from address space where the value is updated;
- `Enabled` – Parameter that enables this entry.
- `Time` – Value that represents time interval between variable update.

As an example, the following code presents one possible entry from the XML configuration file, where the "`Get_x_dir_actpos`" method is indicated to be called to update the value of the node with the id of `6008`, every `1` second.

```
<add key="Get_x_dir_actpos" value="6008,false,1" />
```

Another feature of this configuration file is its first part, where it is possible to specify a set of configuration parameters for each application, such as, for instance:

```
<add key="CNC_CONTROLLER" value="FANUC" />
<add key="CNC_IP" value="192.168.1.4" />
<add key="CNC_PORT" value="8193" />
<add key="POWER_METER_IP" value="192.168.1.2" />
<add key="POWER_METER_PORT" value="502" />
<add key="VIDEO_IN_PATH" value="http://24.181.120.98:8080/" />
<add key="VIDEO_OUT_PATH" value="../../../videos/" />
<add key="VIDEO_FPS" value="30" />
<add key="VIDEO_DURATION" value="10" />
<add key="INFLUXDB" value="true" />
```

where,

- CNC_CONTROLLER – Allows the definition of the CNC controller model, use internally to choose the correct methods for each specific CNC brand and model;
- CNC_IP – Allows the definition of the IP address which the developed application will use to connect to the CNC controller;
- CNC_PORT – Allows the definition of the port which the developed application will use to connect to CNC controller;
- POWER_METER_IP – Allows the definition of the IP address which the developed application will use to connect to an energy meter;
- POWER_METER_PORT – Allow the definition of the port which the developed application will use to connect to an energy meter;
- VIDEO_IN_PATH – Allows the definition of the path to a video stream source;
- VIDEO_OUT_PATH – Allows the definition of the path to save the generated videos;
- VIDEO_FPS – Allows the definition of the generated videos frames per second;
- VIDEO_DURATION – Allows the definition of the video duration (in seconds);
- INFLUXDB – Allow to set the use of a specific database, in this case InfluxDB.

This generic class was initially developed without any relation to the OPC UA, so it can still be used if there is a paradigm shift in terms of its usage in these applications. As an example, the first version of the CNCMonitor application, was developed with the sole objective of collecting data from a CNC, directly to an InfluxDB TSDB (Time Series Database).

3.2 USING A SOFTWARE TO DESIGN OPC UA INFORMATION MODELS

OPC UA is the next generation solution compared to OPC Classic. One of the most important improvements in OPC UA is a powerful Information Model concept: the Address Space. OPC UA allows exposing real-time process data and underlying infrastructure as a consistent information model built up with nodes. The process model is represented by nodes, attributes, and their mutual relationships. Therefore, using OPC UA this powerful concept allows one to expose not only raw process data but also entire consistent information sets about the process state and behavior. The flexibility of OPC UA ensures that no existing or future systems are too complex to be exposed via OPC UA. Of course, such flexibility leads to difficulties during design, development, and deployment, thus the importance of using specific software that was developed to help with information model design, such as:

- **UAModeler**[1] from Unified Automation – Requires an account for software download, with the free version not allowing C# code generation and support for `<UANodeSet>` (official format for files containing sets of OPC UA nodes);
- **UA Model eXcelerator Professional (UMX Pro)**[2] from Beeond – This tool enables developers to graphically configure an OPC UA compliant information model, produce, merge and manipulate `<UANodeSet>` files and generate SDK independent code;
- **OPC-UA-Modeler**[3] from Fraunhofer IOSB – Web-based graphical tool to develop information models;
- **Free OPC UA Modeler**[4] – Free and open source tool, that operates and outputs `<UANodeSet>` files types;
- **Siemens OPC UA Modeling Editor (SiOME)**[5] – Free tool, created by Siemens, to define OPC UA information models or mapping existing companion specifications on a SIMATIC PLC. This tool allows to import

[1] https://www.unified-automation.com/products/development-tools/uamodeler.html
[2] https://beeond.net/umxpro/
[3] https://www.iosb.fraunhofer.de/servlet/is/35891/
[4] https://github.com/FreeOpcUa/opcua-modeler
[5] https://support.industry.siemens.com/cs/document/109755133/siemens-opc-ua-modeling-editor-(siome)-for-implementing-opc-ua-companion-specifications?dti=0&lc=en-WW

and edit information models as XML files, or to generate and to export individualized models;

- **Object Oriented-Internet Address Space Model Designer (ASMD)**[6] from CAS – Initially a commercial product of CAS Lodz Poland, this open source information model design tool with embedded ModelCompiler[7] from OPC Foundation allows working with `<ModelDesign>` file types to generate `<UANodeset>` files, C# and ANSI C source code.

At the time this research started, this kind of software either was not open source or the trial versions had very few features enabled. Therefore, it was decided to develop the generic C# abstract class (CNCBase) described in the Section 3.1 with abstract methods, to help modeling the interface for a generic CNC machine, allowing the development of monitoring application which work with different CNC machines. With this type of implementation, each CNC machine specific brand then overrides the generic class with its specific functions from its specific communication interfaces, but exchanging the information available using the same information model.

Recently, since the founder and Executive Director of CAS, Lodz Poland, decided to open-source the ASDM software, this software was explored in this work and used to both modify the information model, and generate the files needed to build the address space of the built OPC UA servers. Based on the CNC CS, a new CNC machine information model was built, replacing this functionality from CNCBase. In what concerns the machine to sensor data connection the C# CNCBase class is still being used. In future developments of this work, it will be interesting to explore the data bindings feature of ASMD (leveraging machine to sensor connectivity) coupling the Address Space variables with the process (sensors, actuators, etc.) making an in-memory process replica.

3.3 OPC UA SERVER

With the goal of making the collected information available using a standardized approach, the OPC UA protocol was chosen. OPC Foundation provides several implementations in different programming languages. We chose to use the .NET stack version that is available as an open-source project[8], currently with a license

[6] https://github.com/mpostol/ASMD
[7] https://github.com/OPCFoundation/UA-ModelCompiler
[8] https://github.com/OPCFoundation/UA-.NETStandard

that allows developments within education and research fields. The .NET version of OPC UA is the one with most support and available implementations backed by the OPC Foundation. Furthermore, the use of the .NET Framework approach allows the developed OPC UA servers to run on Windows, Linux and macOS (using .NET Core[9]), as well as in embedded systems applications (using nanoFramework[10]). The development of the application in this work is based on the Reference Server[11], given that it was certified to be used in industry by UA CTT[12].

In order to define the Address Space of our OPC UA Server, the CNC CS (VDW and OPC Foundation, 2017) was considered, specifically following the OPC UA information model example of a 3-axis machine tool, as shown in Figure 4 (which can easily be adapted or extended to a different number of axis). After a first version built using UAModeler and the CNCBase class, the more recent version of the information model that defines CNC machines in this case, was developed using the ASMD software, mentioned on Section 3.2. This tool provides a graphical and hierarchical representation of the designed model, following the OPC UA notation and syntax.

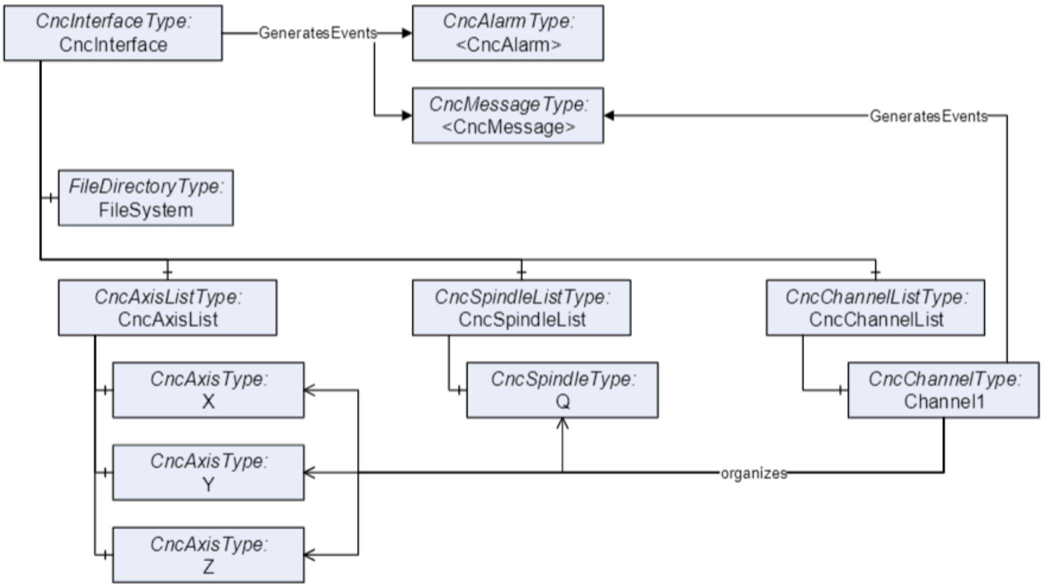


Figure 4: 3-Axis machine tool OPC UA information model (VDW and OPC Foundation, 2017).

[9] https://docs.microsoft.com/en-us/dotnet/core/
[10] https://nanoframework.net/
[11] https://github.com/OPCFoundation/UA-.NETStandard/blob/master/SampleApplications/Workshop/Reference/README.md
[12] https://opcfoundation.github.io/UA-.NETStandard/help/overview_referenceserver.htm

3.4 DATA EXCHANGE WITH THE CNC CONTROLLER

The implementation built within this work prioritizes the CNC machine interaction by means of a communication protocol, SDK or API, when available. In addition or alternatively to this, when the above methods are not supported, but when a log system is still available, a group of messages is defined and implemented within the machining code, which can be sent from the equipment during the machining processes. This can be automated, by adding, for instance, in the FANUC case, DPRNT instructions to the ISO code in the CNC machine post-processor phase.

In addition to the acquisition methods referred above, or when the CNC machine controller does not support any type of communication, one can use a direct (hardware-based) access to the sensors already integrated or externally added to the CNC machine, in order to acquire the needed data directly. In these cases, the chosen approach relies on devices that have wireless and/or wired network support, allowing to take advantages of both wired and wireless-based solutions (Underberg et al., 2018), having in mind the tight requirements often imposed by industrial environments (Cena et al., 2008).

The next subsections will provide further details about the methodology for developing OPC UA servers for CNC machines, by describing the application to specific scenarios, considering three different CNC controller vendors.

The specific case studies were selected based on the main equipment brands and operations used by molds industry companies in the Marinha Grande cluster, in Portugal, and within our research lab. As shown on Figure 5, two types of CNC controllers used in the industry were chosen, namely the FANUC 31i-A and the Heidenhain iTNC530 CNC controllers, and, as part of a parallel project related with a CNC retrofitting process, an Eding CNC controller.

3.4.1 *FANUC CNC machine*

In what concerns data exchange with the 31i-A FANUC controller, the FOCAS (FANUC Open CNC API Specifications) library is used. This library allows data access from the CNC and PMC (Programmable Machine Control) controllers via Ethernet or HSSB (High Speed Serial Bus) (Fanuc and Inventcom, 2018). The FOCAS library, in both the 1 and 2 versions, contains numerous functions[13] that

[13] https://www.inventcom.net/fanuc-focas-library/general/general

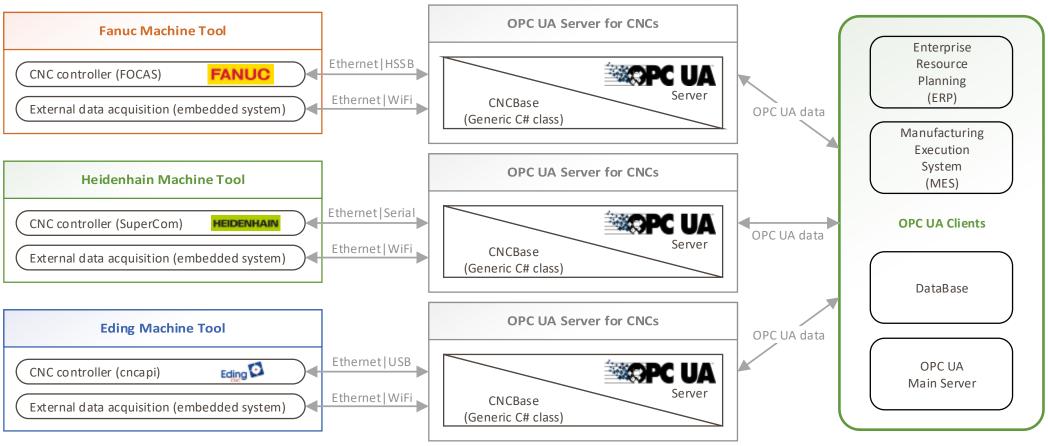


Figure 5: Functional diagram and interaction between systems.

enable data exchange with FANUC CNC machines (Atluru and Deshpande, 2009). However, this library is not supported by all the FANUC controller families, being restricted to machines from the i series. This library can be used for development of Windows and Linux-based applications, as well as for Android and iOS mobile platforms.

Furthermore, FANUC controllers provide several commands that can output variable values and various characters through the RS232 or embedded Ethernet port to externally connected devices. These commands are called External Output Commands, and there are four of them: POPEN (Port Open) and PCLOS (Port Closed) are used to "connect" and "disconnect" to the output port, respectively; between these two instructions one can use BPRNT and DPRNT commands to output data in binary or ASCII formats, respectively (Smid, 2005). These commands can be added in the post-processor phase, allowing the acquisition of some machine parameters when the FOCAS library is not supported by the specific controller, or when the FOCAS library does not provide access to the needed information.

3.4.2 *Heidenhain CNC machine*

The solution used to exchange data with the Heidenhain iTNC530 controllers was based on the SuperCom Heidenhain communication library[14]. This Adontec library allows communication with Heidenhain TNC controllers through the serial or Ethernet (TCP/IP) ports, similarly to FOCAS for FANUC controllers. This library contains functions to build data connections to one or more Heidenhain controllers. SuperCom is event-driven, and enables file transfer to and from the Heindenhain

[14] https://adontec.com/

TNC, listing, creating and deleting folders, renaming and deleting files, reading TNC configuration data, retrieving machine status, machine data and process data, as well as reading and writing memory registers, among other functionalities. The amount of information available may vary between the different controller series. The library also contains direct access functions that can be used to retrieve or modify data directly from the controller connected PLC memory. This library can be used for developing both Windows and Linux-based applications, although a specific license is needed for each platform.

3.4.3 *Eding CNC machine*

Our research lab as been recently involved in a CNC retrofitting project (Lucas, 2019), using an Eding controller board[15]. In this case, data is exchanged with the controller through the USB or Ethernet (TCP/IP) interface using the cncapi[16] communication library. This library allows sending and receiving data by sharing its internal functions, which are also used by the graphical user interface that allows viewing and controlling the Eding CNC controller. An OPC UA server was implemented using the methodology described above, in order to work in parallel and communicate with the Eding software. It is possible to exchange data with all the Eding controllers using this library, independently of the device to be controlled. Note, however, that the library only works on Windows. The Eding library is not available in C#, but only in C++. As such, an existing C++-based code wrapper for C# was used[17], so that data could be accessed from the developed OPC UA server, built using C#.

3.5 EXTERNAL SENSORIZATION - INDUSTRIAL EQUIPMENT

In what concerns the incorporation of industrial devices in the developed OPC UA server, two examples where added to the system: a power meter used to monitor the power usage of the CNC machine, and a camera to monitor and record the machining process in case of alarm events from the CNC machines.

[15] https://www.edingcnc.com/products.php
[16] The cncapi and its documentation is only available after installation of the Eding CNC Software.
[17] https://www.oosterhof-design.com/cncapi-netframework/

3.5.1 *Power Meter*

In this specific case, the industrial device integrated in the developed OPC UA server is an EMpro EEM-MA370[18] energy analyzer, from Phoenix Contact. This equipment make its information available through the Modbus[19] TCP protocol (a version of the protocol intended for use over a TCP/IP network).

The Modbus protocol does not have any cybersecurity feature incorporated in its specification (users, passwords, certificates, or others). On the other hand, the developed OPC UA server has the ability to provide these same security services, necessary for the communication with an external client.

As such, the device that communicates via Modbus and the computer with the OPC UA server need to be located on a private network, protected by a firewall, which prevents unauthorized access from outside (see Figure 6), or to use a point-to-point connection.

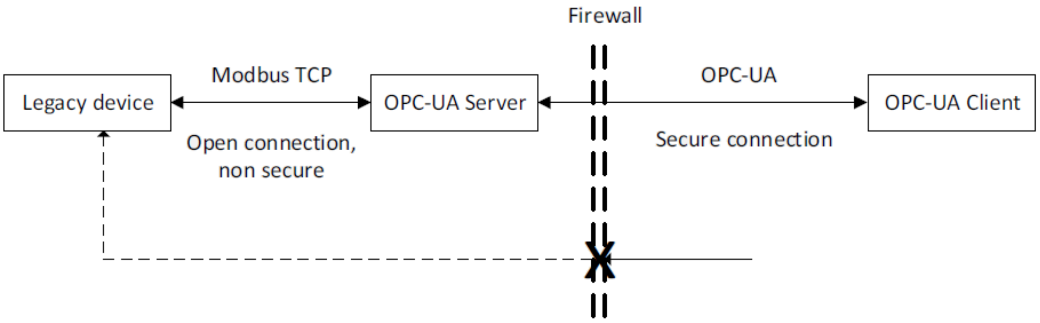


Figure 6: Diagram representing the communication security between devices.

The developed OPC UA application is a Modbus TCP client in terms of communication with the energy analyzer. For external clients, the application is an OPC UA server that interfaces with the energy analyzer securely.

There are several Modbus libraries available. In this case, the open-source EasyModbus[20] library was used. This communication protocol is widely disseminated in the industry, making this implementation particularly useful in every cases in which the industrial device to be integrated has this communication protocol available.

[18] https://www.phoenixcontact.com/online/portal/us?uri=pxc-oc-itemdetail:pid=2907983
[19] https://www.ni.com/pt-pt/innovations/white-papers/14/the-modbus-protocol-in-depth.html
[20] https://sourceforge.net/projects/easymodbustcp/files/latest/download

3.5.2 *Camera*

Another example of an industrial equipment integration is the use of a camera. Currently the main goal of using the camera is to acquire video footage (with lower quality and only for less detailed remote monitoring) of the machine's operation, generating videos in each event created by OPC UA related with CNC alarms.

In order to achieve this, a FIFO (First-In, First-Out) system was implemented, using a C# linked list that contains the last frames of the CNC event. Each time a CNC alarm event occurs, a method is called and the last frames (the current frames on the FIFO system) that contains the recollection of what happened in the time just before, and up until, the event was captured, are saved to a video file and stored in a shared folder. This method also returns the path to the saved video, making it accessible to the operation manager and client through the OPC UA server.

The cameras tested with the developed system were a Genie Nano C1920[21] (industrial camera) from DALSA Teledyne and a generic public IP network camera available through an online internet stream. However, almost any camera supporting RTSP (Real Time Streaming Protocol) or UVC, among other protocols for accessing the video stream, are also supported by the developed system.

Since industrial cameras are also supported by this system, in future developments of this work, machine vision techniques can be applied to achieve image analysis in some situations and, as a result, other types of information can be gathered.

3.6 EXTERNAL SENSORIZATION - DEVELOPMENT OF DEDICATED HARDWARE

To acquire additional data, the IoT development board ESP32-PoE[22], from Olimex was used. Built around the ESP32-WROOM-32 module (Espressif, 2019), it supports WiFi, BLE and 100Mb Ethernet with Power-Over-Ethernet (PoE). It further includes a LiPo battery connector, a MicroSD card slot, GPIO (General Purpose Input Output) headers and an UEXT (Universal EXTension) connector that allows the connection of any device using I2C, SPI or RS232 communication. This board is

[21] https://www.teledynedalsa.com/en/products/imaging/cameras/genie-nano-1gige/

[22] https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware

also available in a galvanic isolated version for the PoE power and in an industrial version[23], supporting an extended temperature range from -40°C to 85°C.

Combining this hardware with the open62541 server, (open source version of OPC UA in C) it is possible to create OPC UA servers for devices with more limited resources, which provide data from a given sensor, or set of sensors, to the main server. The development of these OPC UA servers, uses as starting point the project from the opcua-esp32[24] repository. This project uses the open62541 library to build an OPC UA server to be used on a ESP32, with an information model that contains a temperature sensor (DHT22), a callback method to call the ESP32 onboard LED and two relays, which can be remotely controlled as OPC UA objects.

Two implementations were developed: one that uses an LM35 temperature sensor, connected to an analog input, and another that uses an MPU6050 module, connected through the I2C interface, which includes an accelerometer and 3-axis gyroscope, both with the objective of making the sensor data available through OPC UA to the main server. Using additive manufacturing technology, a case was built to protect the electronics (see Figure 7), as well as allowing easier installation on a DIN rail in the machine's control cabinet.

Despite the specific use of LM35 and MPU6050 sensors, this device can be developed to use any sensor that can be read using an ADC (Analog-to-Digital Converter), I2C interface or any other protocol type available in the ESP32 (this changes in the development might require some signal conditioning circuit depending on the sensor specific output). Additionally, the information model used by the OPC UA server to expose the sensor should be changed, according to the specific sensor characteristics.

In order to explore this project and start changing it to our needs, the ESP-IDF environment needs to be configured, enabling project creation, building and flashing to the ESP32, as well as application debugging. In the technical manual (André Martins, Costelha, Neves, and Bento, 2020b), a result from this project, the instructions to set up this development environment are defined.

[23] https://www.olimex.com/Products/IoT/ESP32/ESP32-POE-ISO/open-source-hardware
[24] https://github.com/cmbahadir/opcua-esp32

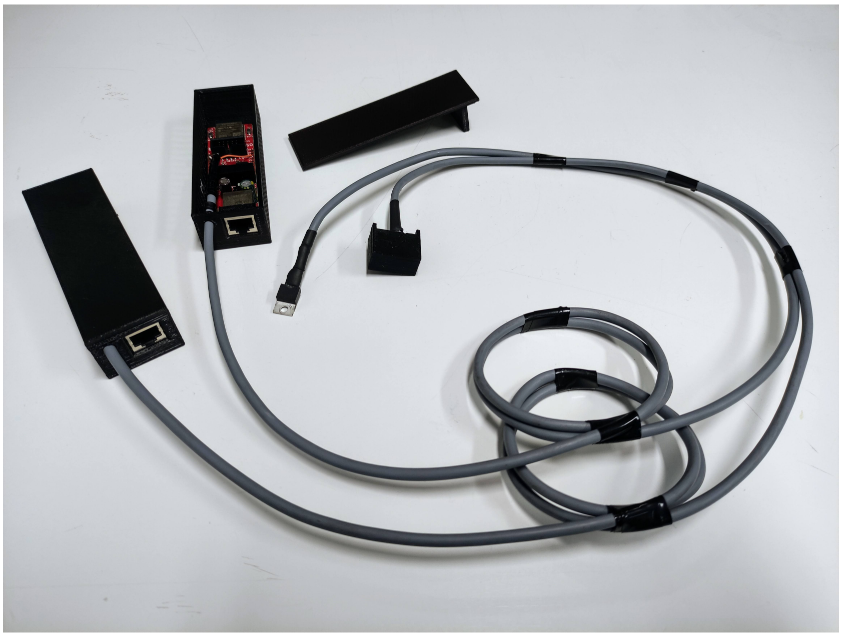


Figure 7: Developed external sensorization for machine monitoring.

4

DIGITAL TWIN TO SUPPORT PLM

This chapter describes the methodology developed and used to build a virtual model, the DT, and to integrate industrial equipment in a smart manufacturing environment, using the concepts described in the Chapter 2.

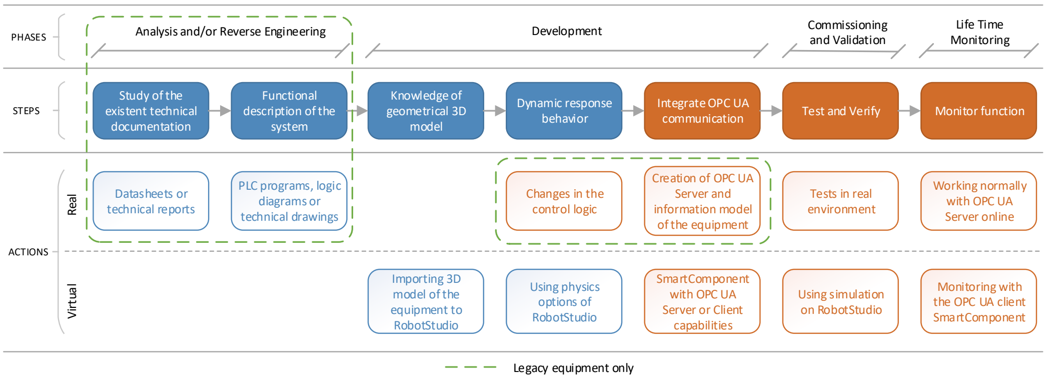

PHASES
Analysis and/or Reverse Engineering
Development
Commissioning and Validation
Life Time Monitoring
STEPS
Study of the existent technical documentation
Functional description of the system
Knowledge of geometrical 3D model
Dynamic response behavior
Integrate OPC UA communication
Test and Verify
Monitor function
ACTIONS
Real
Datasheets or technical reports
PLC programs, logic diagrams or technical drawings
Changes in the control logic
Creation of OPC UA Server and information model of the equipment
Tests in real environment
Working normally with OPC UA Server online
Virtual
Importing 3D model of the equipment to RobotStudio
Using physics options of RobotStudio
SmartComponent with OPC UA Server or Client capabilities
Using simulation on RobotStudio
Monitoring with the OPC UA client SmartComponent
Legacy equipment only

Figure 8: Methodology to support the design, commissioning and supervision of smart factory components through their DT.

4.0.1 *Integrating legacy industrial equipment and developing OPC UA-enabled equipment*

Figure 8 illustrates the methodology developed for design, commissioning and supervision of smart factory legacy equipment, including its DT development and integration. After identifying the legacy equipment that shall be put through an *Industry 4.0 retrofit process*, an analysis, with possibly some reverse engineering, needs to be done. Unlike in the integration of new industrial equipment in which the simulation model is created with the help of complete documentation, in legacy equipment, technical documentation may not be easily accessible. As such, it gets harder to create a robust and accurate simulation model that can serve as a virtual twin. In the development phase, a 3D model with geometrical information of the device must be created, if one does not exist, and then imported to RS. The next step is to add dynamic and responsive behavior, which can be achieved through the available RS SDK, a well documented tool. In the third step, the fusion of the real and virtual/digital world is achieved by feeding virtual models with field data as well as conveying information to the real device, using an OPC UA interface. This OPC UA interface in the real equipment should always be implemented as an OPC UA server. The DT should act like an OPC UA server (in simulate mode) and like an OPC UA client (in monitor mode). When dealing with the integration of both legacy and recent equipment from different manufacturers some possible scenarios could exist. Bellow you find a description of these scenarios and the proposed solutions:

- **The equipment is controlled by a PLC with Classic OPC or OPC UA?** If the automated system is controlled by a PLC, the brand and model should be investigated to find if it has the possibility of creating an OPC UA or a Classic OPC out of the box. When the equipment to integrate natively supports the OPC UA, this communication standard should be preferred for the reasons mentioned in this report, and in this case the integration of such equipment should require less effort compared with the following options. If the option of Classic OPC is available an OPC UA COM Server Wrapper from OPC UA .Net Standard stack could then be implemented to map the information. This option enables to migrate to OPC UA-based systems, while still being able to access information from old existing OPC Classic systems. When the industrial equipment is in a retrofit process or in the development of new devices the use of OPC UA-enabled PLCs, such as Phoenix Contact PLCNext[1], Siemens S7-1500[2] and Omron NJ/NX[3], should be considered.
- **The equipment is controlled by a PLC without Classic OPC or OPC UA?** If the above features are not available in the PLC, other communication protocols should be used to develop the OPC UA interface. For example, using a Modbus server running directly on the PLC and then developing an OPC UA server as an external application running on a PC. Using the OPC UA SDKs from the OPC Foundation that acts like a Modbus client from the PLC point-of-view and, externally, act like an OPC UA server, making the interface with PLC possible. There are some Modbus libraries available such as, for instance, the open-source EasyModbus[4].
- **The equipment is controlled by a PC?** If the automated system is running on a PC, or has a PC-based controller, combining the OPC UA SDKs with APIs from the automated equipment should be the way to build the server.

[1] https://www.plcnext-community.net/en/

[2] https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html

[3] https://www.ia.omron.com/products/family/3705/

[4] https://sourceforge.net/projects/easymodbustcp/

The OPC Foundation makes the stack available in different programming languages, however the one currently with most support and being tested and certified is the .NET Standard[5].

- **The equipment is controlled by embedded systems?** If the automated system is based on embedded systems with low resources available, using open62541[6] is probably the best option. This low weight open-source C implementation of OPC UA is the appropriate option to use in embedded systems and it also supports the PubSub communication, thought its use is not yet certified due to the lack of official test cases and testing tools. The use of OPC UA in PubSub approach allows reaching multiple data consumers (subscribers) with a single message, the publisher is not slowed down by the subscriber processing delays, thus providing higher efficiency than session-based data exchange (client-server) (Gogolev et al., 2020).

In this step, some tests and verification should be done in the commissioning phase to ensure the intended operation of the devices and their components is achieved, and that there is consistency between real and virtual environments. Finally, after the commissioning of the equipment, a constant monitoring is possible trough the OPC UA interface created previously.

4.0.2 *Integrating OPC UA-enabled industrial equipment*

Given the growing acceptance of OPC UA within industry, and with the continued release of standard information models in many industrial equipment sectors, integrating new devices in this virtual environments is becoming easier, particularly in terms of communication capabilities. Furthermore, modern industrial equipment is usually provided with its 3D model and detailed, often digital, technical documentation. As shown in Figure 8, the methodology for integrating new equipment shares some stages with the one described for legacy equipment, excluding the need for OPC UA server development on the device's side, as well as for the initial phase of analysis and reverse engineering on the existing equipment, rendering the task easier and swifter.

[5] https://github.com/OPCFoundation/UA-.NETStandard
[6] https://open62541.org/

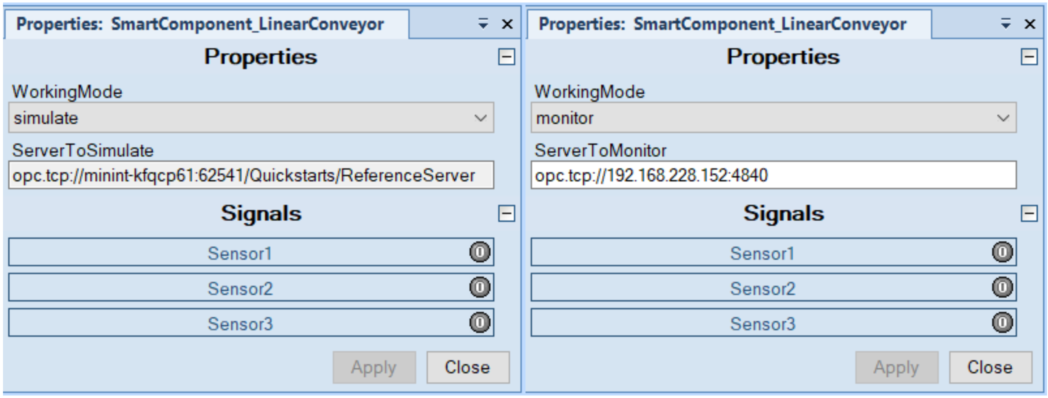
4.0.3 *Digital Twin modes of operation*

Using ABB RS, and its SDKs a generic SmartComponent (SC) was developed, that can be associated to an industrial equipment to implement its DT. This SC provides two working modes for the device it is connected to: *simulate* and *monitor*, as shown in the properties box in Figure 9.

When the RS simulation starts, and the SC *WorkingMode* property is set to *simulate*, an OPC UA server will be launched, simulating all the aspects of the real industrial devices. The address of the created server will show up in *ServerToSimulate* property box.

The other *WorkingMode* option available is *monitor*. When the RS simulation starts in this mode, an OPC UA client will be launched and, if the security requirements are fulfilled, it will connect to the OPC UA server address inserted in the *ServerToMonitor* SC property.

A variety of uses can be given to this OPC UA SC: firstly, during the design phase, the complete simulation can be used to validate design decisions. In a second stage, it may be used to test individual (real) device integration with the rest of the (simulated) environment and to validate its real operation within the industrial process before commissioning the entire line. Finally, after commissioning, and during regular operation, this implementation can be used for the continuous monitoring of the industrial equipment and process based on data analytics techniques, or just as a simple mimicking and monitoring of the real equipment behavior in the RS virtual environment.



Properties: SmartComponent_LinearConveyor

WorkingMode: simulate

ServerToSimulate: opc.tcp://minint-kfqcp61:62541/Quickstarts/ReferenceServer

Properties: SmartComponent_LinearConveyor

WorkingMode: monitor

ServerToMonitor: opc.tcp://192.168.228.152:4840

Figure 9: Developed OPC UA SC properties.

5

TESTS AND RESULTS

This chapter presents several tests and results obtained from the application of the developed methodologies described in this report.

5.1 OPC UA SERVER FOR CNC MACHINES

The developed system was tested with three different CNC machine controllers both in laboratory and industrial environment: Fanuc, Heidenhain and Eding.

Concerning Fanuc, the first tests were first carried in a laboratory environment, using the NCGuide[1] CNC simulator. Some FOCAS functions are not available on this simulator and, because of this limitation, the second stage of tests, were developed in an industrial environment. Using the CNCMonitor application, referred on the Section 3.1, working as a datalogger. Data was continuously collected for two months from a CNC milling machine equipped with a Fanuc 31i-A controller, for experimental results. It should be noted that, to transition from the simulator to the real machine, it was only necessary to change the IP of the controller and the model in the configuration file, as the application worked flawlessly.

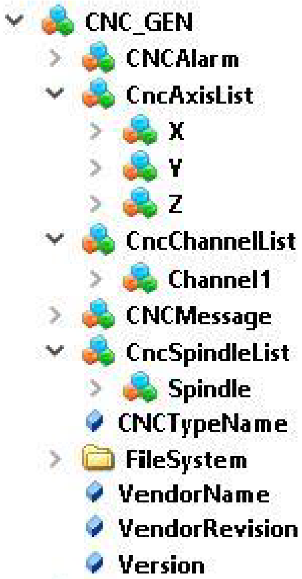
With Eding, tests were made only in the CNC simulator from Eding, given that our retrofitting process was not yet finished. However, since the same software both for the Eding CNC simulator and the real CNC controller board, we do not expect perceivable differences.

Regarding Heidenhain, tests were only made with the iTNC530 CNC simulator so far, with the main limitation of the trial version of the simulator (which we used) being the maximum program size of 100 lines of machine instructions. Due to this limitation, in this test, unlike the others, the CNC axis are moved manually, reflecting these changes in cutter position in the OPC UA server variables.

In what concerns the information model, an hierarchically structured address space of the OPC UA server of one of the CNC machines is shown in Figure 10 as an example. Server address spaces can be accessed with appropriate OPC UA

[1] https://www.fanucamerica.com/products/cnc/software/cnc-guide

generic clients such as, for instance, the UAExpert[2] from United Automation, which was used in this case.



- CNC_GEN
  - CNCAlarm
  - CncAxisList
    - X
    - Y
    - Z
  - CncChannelList
    - Channel1
  - CNCMessage
  - CncSpindleList
    - Spindle
  - CNCTypeName
  - FileSystem
  - VendorName
  - VendorRevision
  - Version

Figure 10: CNC machine OPC UA server address space structure.

A video obtained during these tests is available in the following link: https://youtu.be/VpfkCkDUswU.

5.2 DIGITAL TWIN TO SUPPORT PLM

This section first describes the developed scenario and the manufacturing process, followed by the operation modes of the generic SC, built to embody the application of methodologies described in Section 4. It then presents several tests of their application and the corresponding results.

5.2.1 *Description of the scenario*

The developed scenario consists of a system created in the robotics laboratory of the Polytechnic of Leiria, mainly used for research, development and education, composed by several devices (see Figure 11):

- R1 - Scorbot ER IX robotic arm;
- R2 - Scorbot ER IX robotic arm;

[2] https://www.unified-automation.com/products/development-tools/uaexpert.html

- R3 - ABB IRB1200 industrial robotic arm;
- R4 - KUKA KR6 ARC industrial robotic arm;
- R5 - ABB IRB2400 industrial robotic arm;
- C1 - Intelitek pallet conveyor;
- C2 - Belt conveyor (in-house development);
- V1 - BVS-1280M-INS Teledyne Dalsa camera;
- V2 - UI-5240CP-C-HQ IDS camera;
- CNC - ProLIGHT 1000 milling machine;
- AW - Automatic warehouse (in-house development).

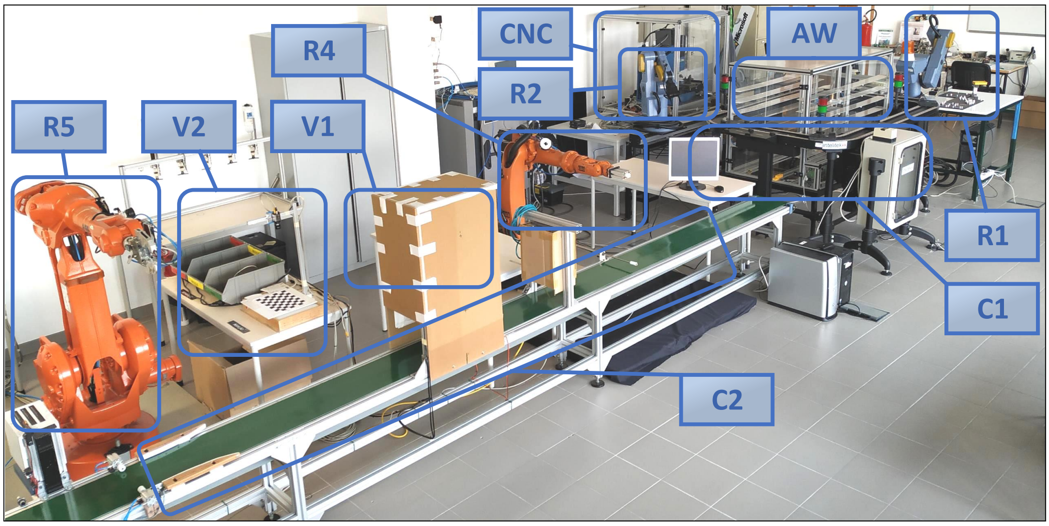


Figure 11: Robotics laboratory of Polytechnic of Leiria.

5.2.2 *The Manufacturing Process*

The functional block diagram, including device physical interactions, is described in Figure 12. This diagram summarizes each device communication and/or I/O capabilities, while also showing a high-level description of the developed manufacturing process scenario.

The manufacturing process begins in the *Parts Supplier* storage cell, where a plastic part is supplied by a specific mechanism, one at a time. When the part is available, the R1 arm picks it up from the supplier mechanism and places it on one of the pallets available on the workstation. Then, this pallet is moved to the shuttle located on the C1 conveyor (the shuttles, which are in the conveyor at all times,

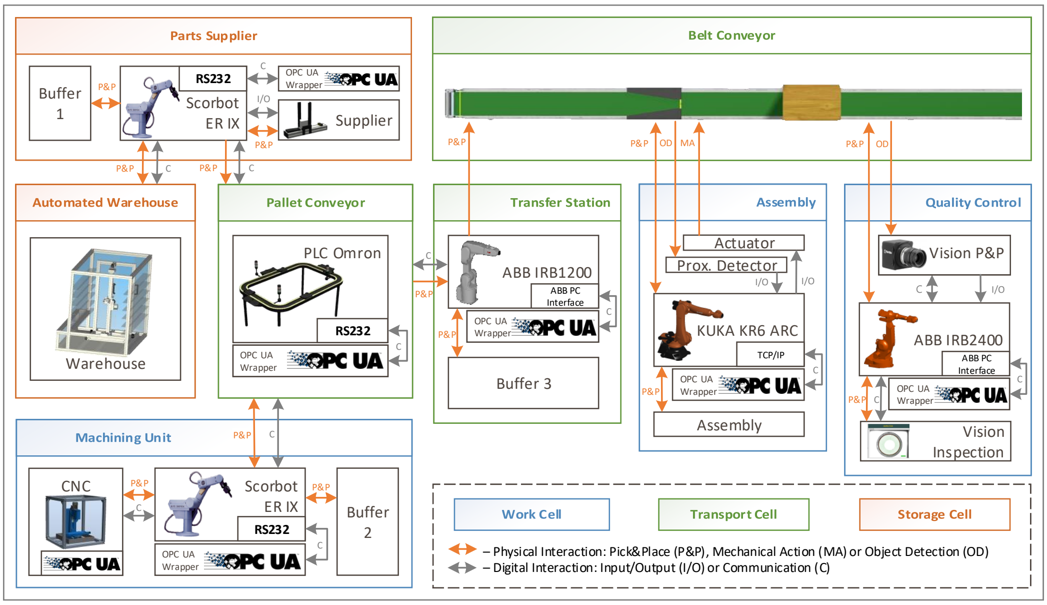


Figure 12: Functional diagram and interaction between systems.

are used to transport pallets between the workstations served by the conveyor). As soon as the palette is placed on the shuttle, the R1 controller signals C1 to release the shuttle from the station, starting the transportation.

When the part is delivered to the machining station, the R2 controller sends a message to C1 to retain the shuttle, while the palette is moved to the CNC area buffer. When this movement is done, the R2 controller signals the conveyor to release the shuttle, effectively finishing the transport operation. The plastic part is then picked by R2 and placed in the CNC machine, in order to perform the needed operations. As soon as the CNC signals the completion of the part, R2 picks it up, places it again on the palette and requests a new transportation from C1. As soon as the shuttle is in place, the palette is returned to the conveyor and the conveyor signaled to carry on.

The process follows by transferring parts, either coming from the automated warehouse of the machining unit, or to the C2 conveyor. R3 (which, at present, only exists in the virtualized model), within the *Transfer Station* transport cell, is responsible for this task, picking machined parts from C1 and placing them on C2. The C2 conveyor is always running and will transport the parts, first to the *Assembly* work cell, and then to the *Quality Control* work cell.

Regarding the *Assembly* work cell, when an OD (Object Detection) event occurs, R4 sends a signal to the MA (Mechanical Actuator) to stop the part in the assembly area. Then, the R4 arm picks the part and places it on the assembly table in order to join it with another part (previously placed on the same table). When this process is complete, the assembled set is placed on C2 by the same R4 arm.

In the *Quality Control* work cell, when an OD event occurs, the R5 controller sends a signal to the MA in order to stop the part on the appropriate place within the quality control area. A machine vision system (V1) will then locate the part on the belt conveyor, pick it and place it on the inspection table. This inspection is done by another machine vision system (V2) located above the inspection table. Depending on the results of the quality tests, the part goes to the packaging process or is placed in a defective part box by the R5 arm.

5.2.3 *Industrial robots integration*

As seen before, this scenario includes five robotic arms from different manufacturers. These have different specifications both in terms of physical capabilities and communication possibilities, which creates the "multi-vendor" environment for mimicking a real case scenario in which both new and old (legacy) machines need to be integrated in an Industry 4.0 process.

The ABB robots controllers that support the PC Interface option can be directly connected to an OPC Classic server application developed by ABB, allowing the access to the controller data organized in two main nodes, called IOSYSTEM and RAPID. These expose IO signals and RAPID data (limited to persistent declared variables) from controllers connected on the same network.

OPC Foundation includes in its open source OPC UA .Net Standard Stack, an OPC UA COM Server Wrapper that is designed to map OPC Classic server information to OPC UA. It enables vendors to migrate to OPC UA-based systems while still being able to access information from old existing OPC Classic systems. This wrapper is used here to convert the information coming from the ABB OPC Classic server to our OPC UA server, since ABB only support the incorporation of OPC UA servers in robots that have their IRC5 controllers with this optional feature enabled[3]. This is a quick solution, since it takes advantage of the existing package, while allowing the validation of the communication with the robot controllers within the simulation/virtualization environment at an early stage of the development. However, it implies the use of an unnecessary translation layer (Robot - OPC Classic - OPC UA), and groups all robot controllers within the same OPC classic server (instead of a distributed architecture with a server per device/workstation within the

[3] https://new.abb.com/products/robotics/controllers/irc5/irc5-options/opc-ua

shop floor network, be it real or simulated). To overcome these limitations, another solution was developed, where an OPC UA server was created for each robot. These servers were developed using the OPC Foundation .NET stack to implement OPC UA functionality and the ABB PC SDK[4] development tool to communicate with the robot controllers, by using their PC Interface optional feature.

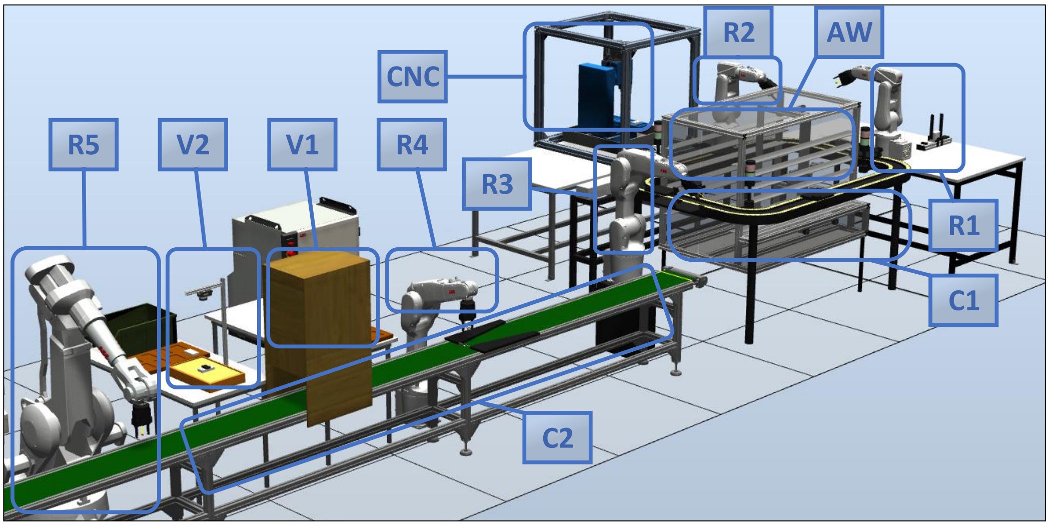


Figure 13: Robotics laboratory virtualization in RobotStudio.

Since ABB RS is a proprietary solution, it does not include the simulation of robots from other vendors. The flexibility of the software could be used for the construction of other robots as RS SC, but that implied an unnecessary deviation from the focus of this project. As such, and without loss of generality and applicability, robots from other manufacturers were virtualized using equivalent ABB robots, as shown in Figure 13. The KUKA KR6 ARC robot is virtualized using an ABB IRB1200, while the two Scorbot ER IX are virtualized using two ABB IRB120.

5.2.4 *Intelitek pallet conveyor*

This element is a rounded corner rectangle, continuous-loop, pallet conveyor that belongs to an earlier CIM (Computer Integrated Manufacturing) system from Intelitek (see Figure 14).

Our Intelitek pallet conveyor and its stopping stations are controlled by an Omron CQM1H-CPU51 PLC that was initially connected to OpenCIM software through RS232 serial communication. Some reverse engineering was applied here to understand how the PLC exchanged data with the OpenCIM, resulting in a list of

[4] https://developercenter.robotstudio.com/pc-sdk

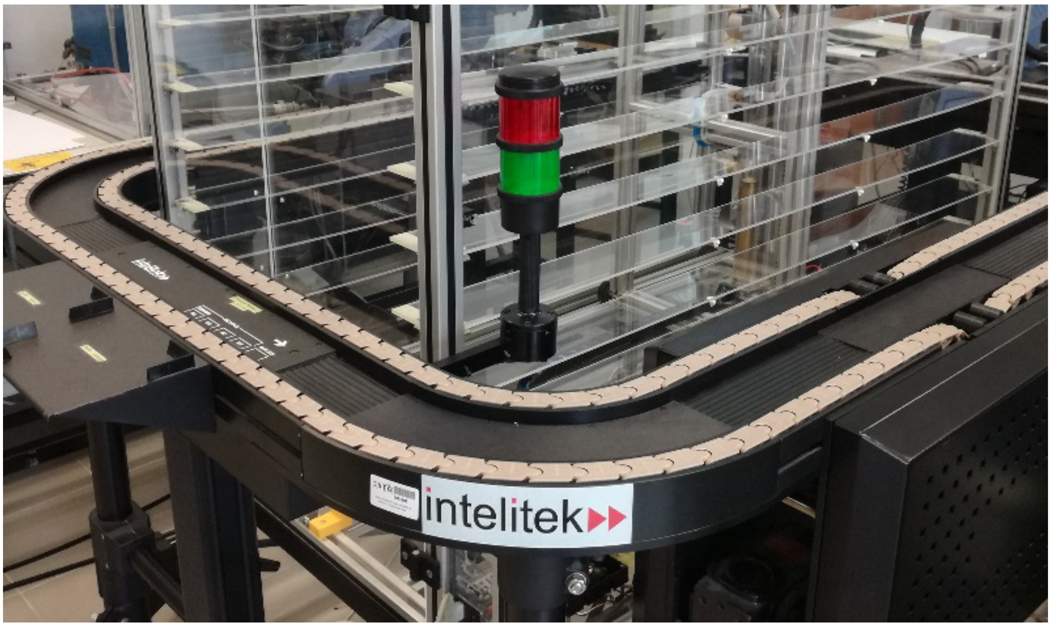


Figure 14: Intelitek pallet conveyor.

available commands that were created to implement the same communication in our virtualized conveyor.

Since RS only provides virtual models of ABB products, a new RS component was designed by creating a 3D model of the conveyor elements using Autodesk Inventor (see Figure 15).

Using the SC features of RS, including physics simulation capabilities, it was possible to implement a model that behaves very similarly to the real conveyor. The communications were virtualized by adding a TCP/IP server to exchange messages with the other elements. This SC functionality was built using the ABB RS SDK, which enables a programmer to develop different kind of custom applications or Add-Ins that can be easily added as a new features to the simulation.

This conveyor is viewed as one of the legacy devices being integrated in this environment, from the Industry 4.0 point of view. For this integration in the virtual system, a TCP/IP to OPC UA adapter was built, which works as a TCP/IP client in terms of communication with the conveyor. Externally, this adapter is an OPC UA server encapsulating the conveyor control and monitoring.

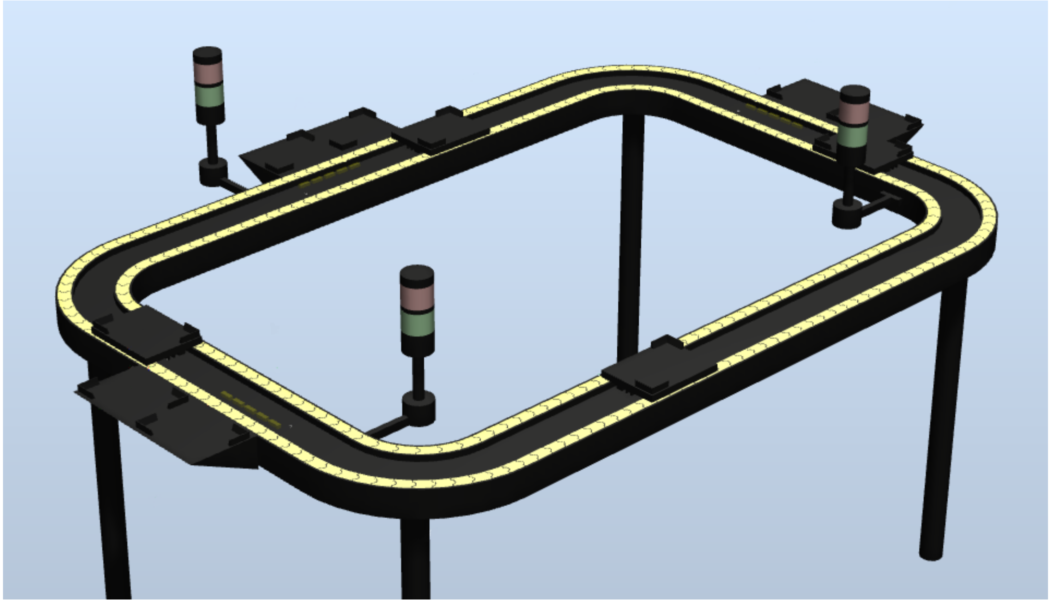


Figure 15: Intelitek pallet conveyor RobotStudio virtualization.

5.2.5 *Machine vision systems*

Machine vision systems are used in two applications in this setup. In the first case, the V1 system is connected to the R5 controller and they work together with the purpose of picking the parts from an exact location on the belt conveyor, and place them on the *Quality Control* work cell. In the second case, the V2 system is also connected with the R5 controller, and the main goal is to perform the quality control inspection of the finished parts.

The integrated vision system that comes with RS and is supported on ABB systems (based on Cognex vision systems) supports mainly the interaction with real cameras, with limited support for emulated cameras. Since the goal here is to acquire images from the simulation process within the simulated environment, a SC-based vision system was created by Hugo Costelha using the RS SDK. When triggered, this SC acquires a simulated view of the camera, saves it to a file, then triggers the image processing application. These vision systems should work with any image processing software, having been tested with both a commercial package (Sherlock[5]) and an open source library (OpenCV[6]), as shown in Figure 17. As in the real operation, the virtual camera needs a calibration process using, for instance, a checkerboard, taking into account the working plane, as shown in Figure 16 (more

[5] https://www.teledynedalsa.com/en/products/imaging/vision-software/sherlock/
[6] https://opencv.org/

complex effects, such as lens distortion, could be added in the future in order to obtain more realistic camera simulations).

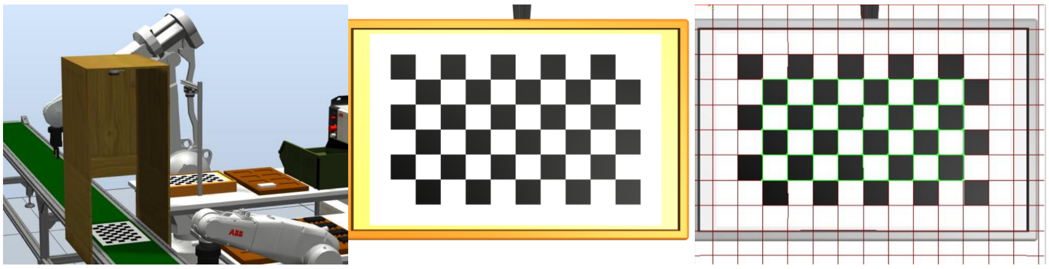
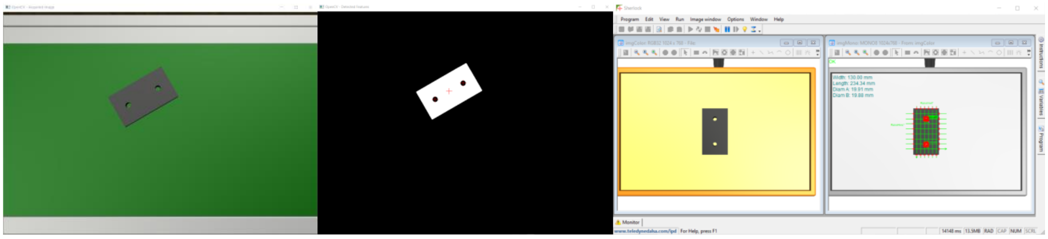


Figure 16: Vision calibration system.



Width: 130.00 mm
Length: 234.34 mm
Diam A: 19.91 mm
Diam B: 19.88 mm

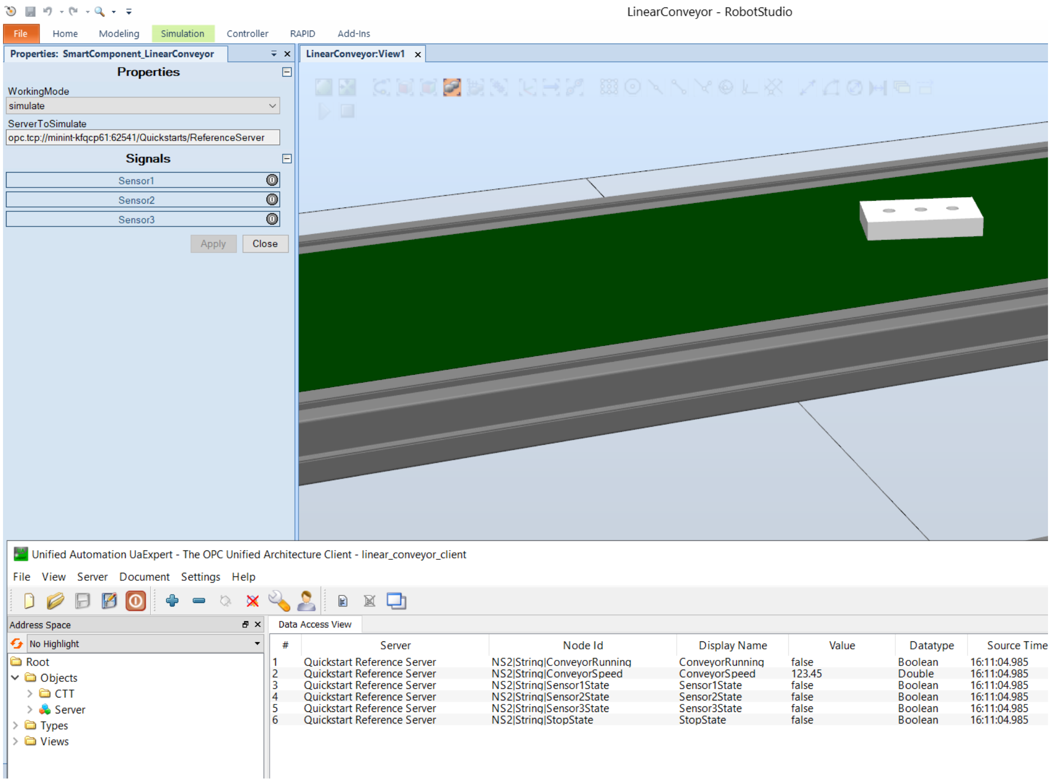
Figure 17: Part detection using vision with OpenCV (left) and Sherlock (right).

5.2.6 *CNC Machine integration*

The CNC milling machine, is also a legacy device, currently undergoing a retrofitting process aiming to replace the earlier CNC controller with a new system based on an EdingCNC controller board. This new controller provides Ethernet connectivity to the CNC along with an SDK for creating customized applications. This application (together with our OPC UA server) allows structured data exchange between the CNC and other equipment, using the CNC CS from OPC Foundation. In the developed case scenario, this CNC is part of the *Machining Unit* work cell and works together with the R2 arm. The machining process is not modeled in our virtual environment, however, the simulated machine SC simulates the output of the milling process, which can be configured to take the amount of time associated with the real milling process.

5.2.7 *SmartComponent*

When the SC is in the *simulate WorkingMode*, a simulation of the industrial equipment behaviors, including its communications capabilities, is executed. Figure 18 shows the RS simulation environment, as well as the UAExpert client that was connected to the simulated OPC UA server, showing the address space created by the SC which describes the belt conveyor. The set of tests made in the *simulate* mode went according with the expected behaviour, obeying to the external orders given by the OPC UA client and, at the same time, returning feedback from the sensors on the DT version of the belt conveyor.



WorkingMode: simulate

ServerToSimulate: opc.tcp://minint-kfqcp61:62541/Quickstarts/ReferenceServer

Sensor1 0

Sensor2 0

Sensor3 0

#	Server	Node Id	Display Name	Value	Datatype	Source Time
1	Quickstart Reference Server	NS2|String|ConveyorRunning	ConveyorRunning	false	Boolean	16:11:04.985
2	Quickstart Reference Server	NS2|String|ConveyorSpeed	ConveyorSpeed	123.45	Double	16:11:04.985
3	Quickstart Reference Server	NS2|String|Sensor1State	Sensor1State	false	Boolean	16:11:04.985
4	Quickstart Reference Server	NS2|String|Sensor2State	Sensor2State	false	Boolean	16:11:04.985
5	Quickstart Reference Server	NS2|String|Sensor3State	Sensor3State	false	Boolean	16:11:04.985
6	Quickstart Reference Server	NS2|String|StopState	StopState	false	Boolean	16:11:04.985

Figure 18: Tests in simulate working mode.

When selected, and upon simulation start, the *monitor WorkingMode* creates an OPC UA client that connects to the real device's OPC UA server. Among the tests made in *monitor* mode, the one from Figure 19, where a plastic part is carried by the belt conveyor, is described here. The part was placed near the end of the belt conveyor with equal position in the real equipment and in its DT, and then the real conveyor motion was started. During that motion the conveyor belt speed was changed several times and, without any direct input feed, the DT followed the speed variations according with the information of the rotary encoder made available through the OPC UA server by the real conveyor. A video show-casing these tests is available in the following link: https://youtu.be/n2sw55RcvHI.

A set of tests was developed to determine the performance of the DT, and to quantify the simulation computational needs for the given scenario, where five

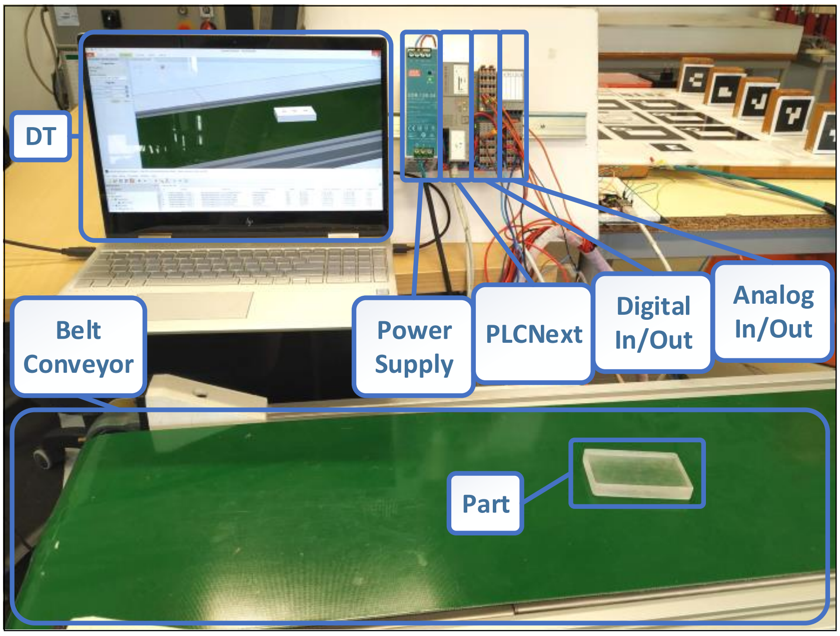


Figure 19: Tests in monitor working mode.

industrial robotic arms, two conveyors, two vision systems, a CNC milling machine and an automatic warehouse are working together to simulate a typical production process. This evaluation was performed on a PC with Intel(R) Core(TM) i7-8550U processor with four 1.8GHz CPU cores. Graphics are provided by a dedicated nVIDIA GeForce MX150 GPU with 1GB of memory. The PC has 12 GB of RAM, a 512 GB SSD, and is running an up to date Windows 10 64-bit.

Using the resource monitor while running the simulation process in RS in real time, it was verified that all processes associated with RS have a small footprint on the CPU cores, with the largest component using up to 20% of a single core. A video obtained during these simulations is available in the following link: https://youtu.be/wLeMhhgLS5A.

6 CONCLUSION AND FUTURE WORK

In this work, a virtual model of a typical production process with broader use through the complete lifecycle of an automated system was implemented, integrating both recent and legacy equipment. That virtualization scenario shows the feasibility of the DT of an industrial process using an existing simulation environment such as RS. In fact, RS was demonstrated to be a feature rich simulator, while supporting user-developed extensions through its SDK. From a subjective point of view, and in the author opinion, the learning curve and the programming effort using this software are comparably shorter than other popular simulators in the robotics field, such as Gazebo and Tecnomatix Process Simulate. Additionally, the resources consumed by the simulation, including the core simulation process and the additional user-defined features running alongside, are perfectly suited to medium performance PCs.

For VC to become an adopted industry standard, the creation of reusable virtual models is a must. This work demonstrates how such models can be created within the RS environment, although the developed methodology could be applied to other simulation environments. Even knowing that RS was developed mainly for offline programming of ABB robots, this work shows that this software, together with its SDKs, can be used in the different stages of the lifecycle of industrial equipment. By merging the capability of the ABB SDKs with the features of the OPC UA stack from OPC Foundation, together with the developed methodologies, a complete lifecycle system was developed, integrating project, simulation and monitoring of an industrial automation system, with real and simulation results being provided. It provides methodologies for integrating not just OPC UA enabled equipment, but also both newly developed and legacy devices.

As this method of creating and implementing SCs relies on the RS SDK, developed by ABB, which is closed-source, there is some risk that future updates might break the API backward compatibility although, historically, that has not been a problem. Another potential difficulty with taking this route of combining RS SC and OPC UA is the need for a different set of skills for automation engineers, ranging from robot and PLC programming, to other programming languages and paradigms not traditionally used in the field.

Regarding the integration of CNC machines, a methodology for the implementation of an all-in-one monitoring solution was developed. Based on OPC UA, prioritizing the equipment interaction by means of a communication protocol, SDK, an API or simply the definition of a group of messages that can be exchanged with the equipment. Additionally, the developed system can access the sensors already integrated, or externally added to the CNC machine, making the future incorporation of other CNC machine models straightforward.

Given the variety of monitoring systems and communication protocols, multi-vendor solutions must take an approach where the various systems available on the market are combined on a single system, allowing the data to be acquired from a wider range of equipment in the industry. With the described approach, it was possible to fully integrate OPC UA within this simulation environment, benefiting from OPC UA's structured approach, open platform characteristics and built-in security. Given the growing acceptance within industry for OPC UA, and the continued release of standard information models (CS) in many industrial equipment sectors, integrating additional equipment in the future should be easier. The main problem of OPC UA is the lack of well documented open source implementations. In fact, even the OPC Foundation's own SDK, used in this development, which is viewed as a reference implementation, is scarcely documented, resulting in a steep learning curve and increased development time.

The natural evolution of the laboratory test scenario is the inclusion of more devices, both in the real world as well as the DT, and to improve the simulation to work seamlessly with real or simulated devices. We are also running experiments with more real world CNC equipment and will have further data in the future.

Future work in this field will include data mining and machine learning techniques in order to automatically detect and predict faults in order to perform predictive maintenance on CNC machines.

Within the wider scope where this work is included, the interconnection of MES and ERP software with shop floor data supplied from our OPC UA Servers, is also being developed, using KPIs (Key Performance Indicators) and OEE (Overall Equipment Effectiveness) metrics to determine efficiency and productivity of the machining processes. Here, the single OPC UA-based server interface plays an important role in guaranteeing a common communication and data model for CNC machines within a given company.

At the time of writing of this report the PubSub feature of OPC UA is not yet certified due to the lack of official test cases and testing tools. Some advantages on using this type of PubSub approaches can be stated compared with the traditional server-client used approach, namely the unnecessary traditional coupled server-client paradigm and better scalability. In future developments of this work, after a fully tested and certified version of this OPC UA feature is made available, including it in the developed work might provide some benefits.

Recently, the Universal Machine Tool Interface (UMATI)[1] working group was established, in which OPC Foundation, the German machine tool builders association (VDW) and other industrial partners, are developing the OPC UA CS for Machine Tools. The aim is to develop an OPC UA information model as a universal communication interface for machine tools to "external" communication partners such as MES, ERP, automation systems or the cloud, planned to be released in the second half of this year[2]. Our developed methodology combines well with those developments. By adapting our interface with the resulting developments of this work group, we will guarantee that the systems based on our solution will work and support the integration, side-by-side, with future products that support this yet to be defined standard.

[1] https://opcfoundation.org/markets-collaboration/umati/
[2] https://opcua.vdma.org/en/viewer/-/v2article/render/47927388

BIBLIOGRAPHY

Arjoni, Diego Hernandez et al. (2018). "Manufacture equipment retrofit to allow usage in the industry 4.0". In: *Proceedings - 2017 2nd International Conference on Cybernetics, Robotics and Control, CRC 2017*. ISBN: 9781538606773. DOI: 10.1109/CRC.2017.46.

Atluru, Sri H and A Deshpande (2009). "Data to Information: can MTconnect deliver the promise". In: *37th Annual North American Manufacturing Research Conference (NAMRC 37)*. ISSN: 10473025.

Ayani, M., M. Ganebäck, and Amos H.C. Ng (2018). "Digital Twin: Applying emulation for machine reconditioning". In: *Procedia CIRP*. Vol. 72. Elsevier B.V., pp. 243–248. DOI: 10.1016/j.procir.2018.03.139.

Ayatollahi, Iman et al. (2013). "Prototype OPC UA Server for Remote Control of Machine Tools". In: *IN-TECH*.

Bartodziej, Christoph Jan (2017). *The Concept Industry 4.0 - An Empirical Analysis of Technologies and Applications in Production Logistics*. Springer Fachmedien Wiesbaden. DOI: 10.1007/978-3-658-16502-4.

Cavalieri, Salvatore, Marco Giuseppe Salafia, and Marco Stefano Scroppo (Jan. 2019). "Integrating OPC UA with web technologies to enhance interoperability". In: *Computer Standards and Interfaces* 61, pp. 45–64. ISSN: 09205489. DOI: 10.1016/j.csi.2018.04.004.

Cena, Gianluca, Adriano Valenzano, and Stefano Vitturi (Mar. 2008). "Hybrid wired/wireless networks for real-time communications". In: *IEEE Industrial Electronics Magazine* 2.1, pp. 8–20. ISSN: 19324529. DOI: 10.1109/MIE.2008.917155.

De Melo, Pablo Felipe Soares and Eduardo Paciencia Godoy (June 2019). "Controller Interface for Industry 4.0 based on RAMI 4.0 and OPC UA". In: *2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 229–234. ISBN: 9781728104294. DOI: 10.1109/METROI4.2019.8792837.

Downey, Jonathan, Sebastian Bombiński, et al. (2015). "Automatic multiple sensor data acquisition system in a real-time production environment". In: *Procedia CIRP*. Vol. 33. DOI: 10.1016/j.procir.2015.06.039.

Downey, Jonathan, Denis O'Sullivan, et al. (2016). "Real Time Monitoring of the CNC Process in a Production Environment- the Data Collection & Analysis Phase". In: *Procedia CIRP*. Vol. 41. DOI: 10.1016/j.procir.2015.12.008.

Drahos, Peter et al. (Apr. 2018). "Trends in industrial communication and OPC UA". In: *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*. Vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., pp. 1–5. ISBN: 9781538644218. DOI: 10.1109/CYBERI.2018.8337560.

Espressif (2019). *ESP32-WROOM-32 Datasheet*. Tech. rep. Espressif.

Fanuc and Inventcom (2018). *FANUC Open CNC API Specifications*.

Fernández, Igor Azkarate, Mikel Ayani Eguía, and Luka Eciolaza Echeverría (Sept. 2019). "Virtual commissioning of a robotic cell: An educational case study". In: *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. Vol. 2019-September. Institute of Electrical and Electronics Engineers Inc., pp. 820–825. ISBN: 9781728103037. DOI: 10.1109/ETFA.2019.8869373.

Forgó, Zoltán, Alfred Hypki, and Bernd Kuhlenkotter (2018). "Gesture based robot programming using ROS platform". In: pp. 455–461.

Giessbauer, Reinhard et al. (2018). *Global Digital Operations Study 2018 - How industry leaders build integrated operations ecosystems to deliver end-to-end customer solutions*. PwC, pp. 1–64. ISBN: 3315657866.

Gogolev, Alexander, Roland Braun, and Philipp Bauer (Jan. 2020). "TSN Traffic Shaping for OPC UA Field Devices". In: *IEEE 17th International Conference on Industrial Informatics (INDIN)*. Institute of Electrical and Electronics Engineers (IEEE), pp. 951–956. DOI: 10.1109/indin41052.2019.8972252.

Grieves, Michael and John Vickers (Jan. 2016). "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems". In: *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Springer International Publishing, pp. 85–113. ISBN: 9783319387567. DOI: 10.1007/978-3-319-38756-7{_}4.

Harrison, William S. and Frederick Proctor (2015). "Virtual Fusion: State of the Art in Component Simulation/Emulation for Manufacturing". In: *Procedia Manufacturing*. DOI: 10.1016/j.promfg.2015.09.069.

Hunkar, Paul (2014). *OPC UA vs OPC Classic*. Tech. rep. DSInteroperability.

Imtiaz, Jahanzaib and Jurgen Jasperneite (2013). "Scalability of OPC-UA down to the chip level enables "internet of Things"". In: *IEEE International Conference on Industrial Informatics (INDIN)*. ISBN: 9781479907526. DOI: 10.1109/INDIN.2013.6622935.

Jbair, Mohammad et al. (May 2019). “Automatic PLC code generation based on virtual engineering model”. In: *Proceedings - 2019 IEEE International Conference on Industrial Cyber Physical Systems, ICPS 2019*. Institute of Electrical and Electronics Engineers Inc., pp. 675–680. ISBN: 9781538685006. DOI: 10.1109/ICPHYS.2019.8780213.

Khan, Adnan et al. (Dec. 2018). “Digital Twin for Legacy Systems: Simulation Model Testing and Validation”. In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2018-August. IEEE Computer Society, pp. 421–426. ISBN: 9781538635933. DOI: 10.1109/COASE.2018.8560338.

Koenig, N. and A. Howard (2005). “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In: ISBN: 0-7803-8463-6. DOI: 10.1109/iros.2004.1389727.

Konstantinov, Sergey et al. (2017). “The Cyber-physical E-machine Manufacturing System: Virtual Engineering for Complete Lifecycle Support”. In: *Procedia CIRP*. Vol. 63. Elsevier B.V., pp. 119–124. DOI: 10.1016/j.procir.2017.02.035.

Kritzinger, Werner et al. (Jan. 2018). “Digital Twin in manufacturing: A categorical literature review and classification”. In: *IFAC-PapersOnLine* 51.11, pp. 1016–1022. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.08.474.

Liu, Chao et al. (Apr. 2019). “A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect”. In: *Journal of Manufacturing Systems* 51, pp. 61–74. ISSN: 02786125. DOI: 10.1016/j.jmsy.2019.04.006.

Lucas, João (2019). *Desenvolvimento de servidor OPC UA para sistema CNC*. Leiria.

Martins, Andre, Hugo Costelha, and Carlos Neves (Apr. 2020). “Supporting the design, commissioning and supervision of smart factory components through their digital twin”. In: *2020 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 114–119. ISBN: 9781728170787. DOI: 10.1109/ICARSC49921.2020.9096072.

Martins, André, Hugo Costelha, and Carlos Neves (Apr. 2019). “Shop Floor Virtualization and Industry 4.0”. In: *19th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2019*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781728135588. DOI: 10.1109/ICARSC.2019.8733657.

Martins, André, Hugo Costelha, Carlos Neves, and Luís Conde Bento (Oct. 2020a). *CNCServer - Technical Manual*. Tech. rep. Leiria: Polytechnic of Leiria.

— (Oct. 2020b). *CNCServer - User Manual*. Tech. rep. Leiria: Polytechnic of Leiria.

Martins, André, João Lucas, et al. (Nov. 2020). “Developing an OPC UA Server for CNC Machines”. In: *2020 International Conference on Industry 4.0 and Smart Manufacturing*.

Mourtzis, Dimitris, Nikolaos Milas, and Nikolaos Athinaios (Jan. 2018). “Towards Machine Shop 4.0: A General Machine Model for CNC machine-tools through OPC-UA”. In: *Procedia CIRP*. Vol. 78. Elsevier B.V., pp. 301–306. ISBN: 9781510875692. DOI: 10.1016/j.procir.2018.09.045.

Neu, Charles Varlei, Ina Schiering, and Avelino Zorzo (Nov. 2019). “Simulating and detecting attacks of untrusted clients in OPC UA networks”. In: *ACM International Conference Proceeding Series*. New York, New York, USA: Association for Computing Machinery, pp. 1–6. ISBN: 9781450372961. DOI: 10.1145/3360664.3360675.

Neves, Carlos, Hugo Costelha, and Luís Bento (2018). “Indústria 4.0 - A quarta revolução industrial”. In: *O Molde nº118*.

OPC Foundation (2018). *OPC UA Specification Part 14 - PubSub 1.04*.

Pitonakova, Lenka et al. (July 2018). “Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators”. In: Springer, Cham, pp. 357–368. DOI: 10.1007/978-3-319-96728-8{_}30.

René de Jesús, Romero Troncoso et al. (2003). “Driver current analysis for sensorless tool breakage monitoring of CNC milling machines”. In: *International Journal of Machine Tools and Manufacture* 43.15. ISSN: 08906955. DOI: 10.1016/j.ijmachtools.2003.08.004.

Riera, B. and B. Vigário (July 2017). “HOME I/O and FACTORY I/O: a virtual house and a virtual plant for control education”. In: *IFAC-PapersOnLine* 50.1, pp. 9144–9149. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.1719.

Sicherheit in der Informaationstechnik, Bundesamt für (2017). *OPC UA Security Analysis*. Tech. rep. für Sicherheit in der Informaationstechnik, Bundesamt.

Smid, P (2005). *Fanuc CNC Custom Macros: Programming Resources for Fanuc Custom Macro B Users*. Industrial Press, pp. 267–274.

Staranowicz, Aaron and Gian Luca Mariottini (2011). “A survey and comparison of commercial and open-source robotic simulator software”. In: *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA ’11*. New York, New York, USA: ACM Press, p. 1. ISBN: 9781450307727. DOI: 10.1145/2141622.2141689.

Stavropoulos, P. et al. (2016). “Tool wear predictability estimation in milling based on multi-sensorial data”. In: *International Journal of Advanced Manufacturing Technology* 82.1-4. ISSN: 14333015. DOI: 10.1007/s00170-015-7317-6.

Straus, Patrick et al. (Dec. 2018). "Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 1474–1483. ISBN: 978-1-5386-5035-6. DOI: 10.1109/BigData.2018.8622076.

Tao, Fei, Qinglin Qi, et al. (Aug. 2019). "Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison". In: *Engineering* 5.4, pp. 653–661. ISSN: 20958099. DOI: 10.1016/j.eng.2019.01.014.

Tao, Fei and Meng Zhang (2017). "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing". In: *IEEE Access* 5, pp. 20418–20427. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2756069.

Underberg, Lisa et al. (June 2018). "Towards hybrid wired-wireless networks in industrial applications". In: *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*. Institute of Electrical and Electronics Engineers Inc., pp. 768–773. ISBN: 9781538665312. DOI: 10.1109/ICPHYS.2018.8390804.

VDMA and OPC Foundation (2019a). *OPC UA Companion Specification for Machine Vision*. Tech. rep. VDMA and OPC Foundation.

— (2019b). *OPC UA Companion Specification for Robotics*. Tech. rep. VDMA and OPC Foundation.

VDMA and Fraunhofer IOSB-INA (2017). *Industrie 4.0 Communication Guideline Based on OPC UA*. Tech. rep. VDMA, Fraunhofer IOSB-INA.

VDW and OPC Foundation (2017). *OPC UA Companion Specification for CNC Systems*. Tech. rep. VDW and OPC Foundation.

Vrabič, Rok et al. (2018). "Digital twins: Understanding the added value of integrated models for through-life engineering services". In: *Procedia Manufacturing*. ISSN: 23519789. DOI: 10.1016/j.promfg.2018.10.167.

Zhong, Ray Y. et al. (2017). "Intelligent Manufacturing in the Context of Industry 4.0: A Review". In: *Engineering*. ISSN: 20958099. DOI: 10.1016/J.ENG.2017.05.015.