Flexible Work Cell Simulator using Digital Twin Methodology for Highly Complex Systems in Industry 4.0

Pedro Tavares[1,3], Joao André Silva[2], Pedro Costa[1,2], Germano Veiga[1,2], and António Paulo Moreira[1,2]

[1] FEUP - Faculty of Engineering of University of Porto, Portugal
[2] INESC TEC - INESC Technology and Science formerly INESC Porto
Porto, Portugal
[3] SARKKIS robotics, Porto, Portugal

Abstract. The continuous evolution in manufacturing processes has attracted substantial interest from both scientific and research community, as well as from industry. Despite the fact that streamline manufacturing relies on automation systems, most production lines within the industrial environment lack a flexible framework that allows for evaluation and optimisation of the manufacturing process. Consequently, the development of a generic simulators able to mimic any given workflow represent a promising approach within the manufacturing industry. Recently the concept of digital twin methodology has been introduced to mimic the real world through a virtual substitute, such as, a simulator. In this paper, a solution capable of representing any industrial work cell and its properties is presented. Here we describe the key stages of such solution which has enough flexibility to be applied to different working scenarios commonly found in industrial environment.

Keywords: Work Cell Simulator, Digital Twin, Diversify Communication Module, Industry 4.0, Industrial Environment

1 Introduction

The organisational growth has been directly linked to the ever growing evolution of scientific and technological fields. Automation systems have become key solutions to increase efficiency in most industrial environments. Robots present themselves as the top feature resulting from the development of these automation system as they ensure key characteristics for streamlining automated processes. Nowadays, robots' development is based on the ability of robots being autonomously adaptable to several applications. There are a wide range of operations that prove the need for automation in different industrial fields, from shipbuilding to aerospace industries, in which robots also prove to be an high performance enhancer [12, 1, 7].

Automated systems have proven to be efficiency enablers in different industries. Therefore, the importance of developing simulators to cope with the improvements of such systems has risen. The ultimate goal of these simulators is focused on the ability of evaluating real scenarios and hypothesis, consolidating such hypothesis or adjusting them to better fit the underlined problem. Examples of the crucial importance of these software are the medical and surgery training, the aviation testing or even the scalability of technological environments [10, 3].

Robotic solutions present extra complexity due to the high redundancy level of common work cells. As such, a robotic simulator emerges as a key tool to compute all the stages associated with robotic operations ranging from pick and place to transport of equipment. Thus, a software that intends to mimic robotic work cells ideally needs to represent exactly all components and display them accurately while facilitating poses, paths and trajectories testing.

Recently, efforts have been made towards the beginning of new industrial revolution. Industry 4.0 represents a novel concept that includes cyber-physical systems and Internet of Things (IoT). Its properties translate into smart systems and untimely smart industries. Within this industrial trend, notion of "Digital Twins", virtual substitutes of real world objects, has gained considerable importance. Consequently, simulators can be integrated as iterative tools to achieve program optimisation for a given work cell [15, 2].

The main goal of this work is to develop a flexible simulation solution to be used by present and future robotic and automated work cells. The proposed solution fits the requirements enumerated above, while being able to communicate with any external devices (such as Programmable Logic Computers (PLC) or sensors) using generic protocols. In addition, it can also reproduce jobs and tasks of any dynamic work cell designed for the simulator.

The current paper is structured in five main sections. Section 2 aims to provide an overview of the current state of the technology associated with these software simulators. Then, section 3, System Architecture, will describe the fundamentals used for implementation of the proposed simulator. Section 4, 3D visualisation, will provide an explanation of the graphic engine and how the 3D visualisation rendering assures an accurate representation of any given work cell and its properties. Section 5, Developed Features, intends to present the key operation layers of the developed simulator, such as work cell construction, communication channels, motion validation and work cell optimisation. Then, in section 6, Proposed Simulator & Industry 4.0, an overview of the contribution of this simulator to the new concept of industrial environment will be presented, following the premises of Digital Twin Simulation. Section 7, Discussion and Future Perspectives, will summarise the results achieved with the simulator while reviewing the contribution of this work to the scientific community and its future updates.

2 Related Work

Simulation software is key in today's research and industrial community. There have been studies that validate the usage of simulator tools to provide a dynamic view over a real world work cell or individualised system without incurring in danger of damaging such system or components surrounding it. Furthermore, these software simulators accurately mimic the real world system behaviour. Applications of these software solutions range from control models testing, to semiconductor manufacturing or even vehicles dynamic study [13, 4, 6].

Robot-based solutions have demonstrated tremendous utility in the industrial world. However, its design, representation and validation is a complex problem. The efforts in generating new frameworks and tool kits to help bring robotics simulation to a one-to-one relationship with real-world interaction have resulted in a set of simulators that provide a better insight in a real work cell system, thus saving time and financial efforts [9, 11].

An important feature to consider when developing a complex robotic work cell is related to the extended study of its main components. In that regard, there are other minor software such as RoboDK, RoKiSim or COSIMIR which allow to study small stages such as robot kinematics or work cell positioning. Furthermore, recently, the RoboDK organization developed simplistic approaches to automatic work cell calibration tasks [14, 5].

Furthermore, a key stage of automated or robotic work cells is related to the interface with external devices or software. Modern work cells intend to use external knowledge acquired from sensors (vision systems, force sensors, lasers) or from input signals coming from PLCs and external consoles. This integration is well documented and validated within the current state of the art [17, 16].

A recent and fast growing solution that promotes modular implementations has been proposed based on the robotic operative system (ROS) and its associated simulator Gazebo, as well as Visual Components. These allow for the simplification of most problems by developing multiple simple algorithms that, when integrated, assure a modular solution to a complex problem while adding flexibility to the overall system.

Although all the strategies and applications described above have been implemented in the industrial world there are still some limitations to be resolved.

3 Simulator Architecture

An ideal simulator should be defined as a multi layer software capable of recreating a real work cell while accepting new inputs and adjusting accordingly to any given feedback.

In order to implement a well structured simulator, the proposed architecture follows the paradigm of MVVM (Model-View-View Model). This architecture is divided into three major layers: (1) the View, which is the graphical structure that serves all user interfaces (UI), (2) the View Model, which is the programmable section of software that manages properties and commands to be displayed or raised by bindings with the View, and (3) the Model that represents the real content of a defined object (see figure 1).

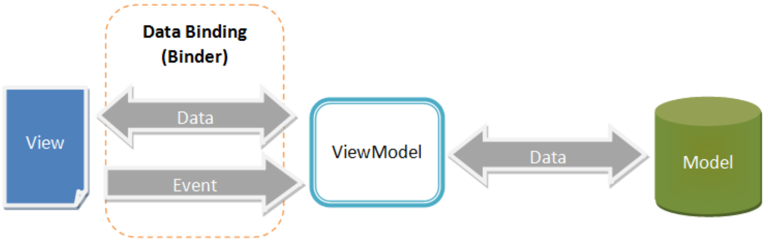


Fig. 1. MVVM Architecture Schematic

Moreover, based on this architecture, the proposed simulator has five different tiers. Each tier of the proposed simulator is built following a modular methodology (see Table 1).

Table 1. Tiers in the Proposed Simulator

Tier	Tier Specification	Application Examples
1	Modular Work Cell Construction	Inserting components: External Axis, Robots, Machinery, Tools. Creating Components Chains.
2	Communication Channels	Modbus RTU. Modbus TCP. Ethernet TCP/IP. Communication with External Devices.
3	Robot / Machinery / System Motion.	Poses Evaluation. Joint or Linear Movements. Drag and Drop. Path Planning Add-in.
4	Task / Job Loading and Testing.	Load external files and decrypt them to generic languages. Run movements lists.
5	Optimisation.	Poses, Layout or Components Optimisation.

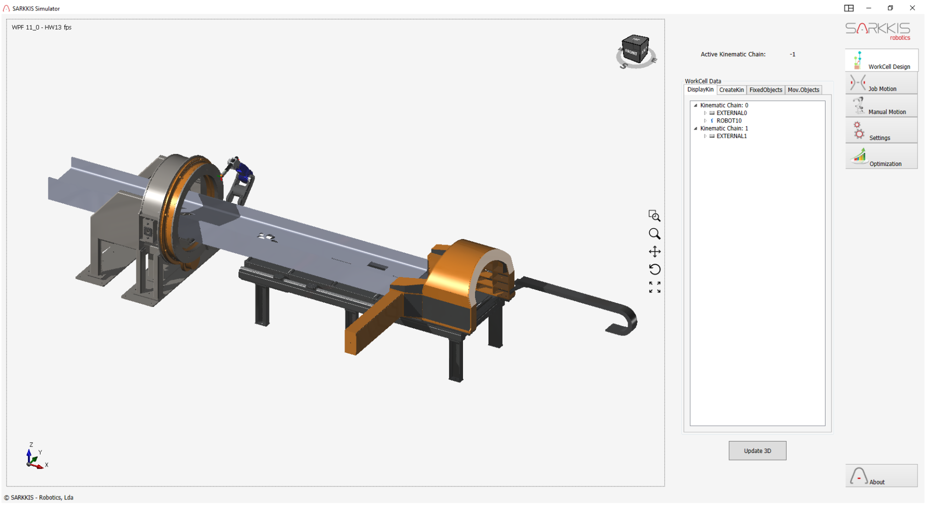
4 3D Visualization

An important aspect of any simulator is its user interface. Thus, a 3D engine able to render the entities of structures to be displayed was selected for this work, and the software Eyeshot Ultimate 10 was used [8].

This Eyeshot Ultimate 10 software is based on the construction of geometries (e.g. Mesh, surfaces, solids, lines or just profile extrusions) which are associated to major blocks and recursively inserted within other blocks in order to recreate a geometrical chain. This methodology is easily integrated with automated and robotic work cells where there may exist a tool attached to a robot placed on top of an external axis, or simply a machine structure with tools placed in key positions. For a detailed recreation of the work place, it is possible to place separate blocks containing walls, tables or even human operators geometry details, thus, completing the work environment.

Another crucial aspect of a 3D engine built with Eyeshot Ultimate 10 is the ability to animate all attached blocks given a normalised axis transformation. Thus, it is possible to recreate robots/parts movements by crossing the axis information at each joint with this animation feature. Additionally, similar to all 3D engines, Eyeshot has the rotate/translate view features that allow a better grasp of the recreated cell, as well as, a unique feature of dragging and dropping components (Object Manipulator) that provides an inexperienced user with a wide range of validation tools (from repositioning parts to visualising better displays for such parts).

An example of a high redundancy robot-based system developed on Eyeshot can be found below (see figure 2).



Active Kinematic Chain: -1

Kinematic Chain: 0
EXTERNAL0
ROBOT10
Kinematic Chain: 1
EXTERNAL1

Fig. 2. Example of a robotic work cell - SARKKIS robotics simulator

5 Developed Features

In addition to the potential of 3D visualization and animation, the proposed simulator has five main features as stated in Table 1. The following sub chapters will synthesise each of the main tiers.

5.1 Modular Work Cell Construction

The modular work cell construction allows for selection of individual components from a database or using acceptable files and then a assembly of these to form chains.

In order to insert a component, the system requires a generic language document that follows a combination of URDF (Unified Robot Description File) and SDF (Simulation Description File) standards with some minor adjustments. SDF files require a global knowledge of the work chain, while URDF files only represent a single component of such chain. The file type defined as generic input of this simulator serves both worlds, as it allows to create from single objects description files to complete work chains. It is accomplished by defining components of interest for each file, its relations and possible future transformations points that may link the chain described on one file with the following ones.

Then, to better organise the insertion and allocation process, five major types of components were selected:

1. External Structures: comprehending any structure that can be linked to other components such as robotic external axis or CNC machinery.
2. Robots: a large database of robots is associated to the software and the possibility of linking others by creating an URDF based file is presented to the user (example on figure 3).

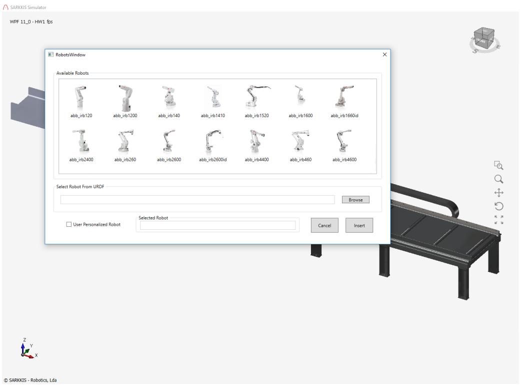


Fig. 3. Example of Robot Choosing Window

3. Tools: comprehending lasers, grippers, visual attachments, other sensors or a plain combination of all the above.
4. Fixed Items: components that simply give a more detailed aspect to the simulation environment and may serve as frame referential or components' support.
5. Moving Items: parts that may move separately from the rest of the chain or that may insert some motion on attached components without moving themselves, such as conveyors.

Once all components are inserted in the work cell, it is possible to remove, replace or re-position them. Moreover, the linkage between components is defined using context menus where it is indicated whether or not a component should be inserted within a block of a previous component.

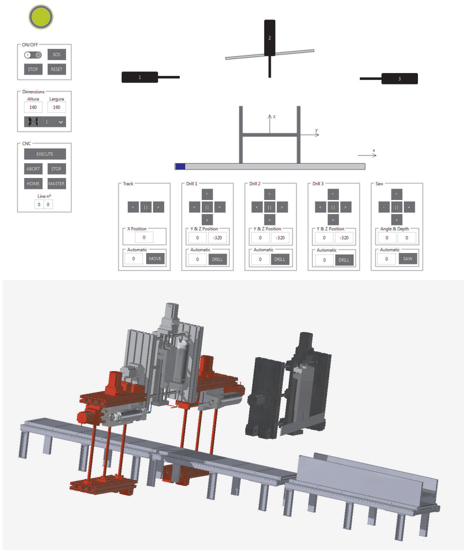
5.2 Communication Channels

In order to integrate external devices with the simulator,two major connection channels were created: Modbus TCP/IP and Ethernet IP.

The Ethernet IP is a standard protocol that uses the potential of Ethernet communication to send UDP packages that contain basic Input/Output data.

The Modbus TCP/IP consists in Modbus RTU (Remote Terminal Unit) protocol with a TCP interface running on Ethernet. Modbus RTU is an application protocol, as it defines rules for organising and interpreting data, but consists in a messaging structure, independent of the underlying physical layer. Essentially, Modbus TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (Modbus as the application protocol).

To validate the communication between controllers and the simulator, it was parametrized a soft PLC using the development system 'Codesys', compliant with the IEC 61131-3 standard. The main goal of it is to control all the operations regarding the simulation of the beam drilling and sawing machine shown along the controller's HMI on figure 4.



Dimensions: Altura 160, Largura 160

Line nº 0 0

Track: X Position 0; Automatic 0

Drill 1: Y & Z Position 0 -320; Automatic 0

Drill 2: Y & Z Position 0 -320; Automatic 0

Drill 3: Y & Z Position 0 -320; Automatic 0

Saw: Angle & Depth 0 0; Automatic 0

Fig. 4. Beam drilling and sawing machine simulation & the respective HMI

With that goal in mind, the PLC can not only command and control all the manual and semi automatic operations activated using the features presented on the HMI, but also has the ability to execute CNC programs generated by a CAM (Computer Aided Manufacturing) software for the production of beams, MetroID BeamCUT, developed by SARKKIS Robotics. This software imports IFC and DSTV files and generates text files containing G Code that are subsequently read and interpreted by the controller. After interpreting the G Code, the controller determines which axes are to be moved and controls their movement with position related feedback.

In order to simulate the control of a beam drilling and sawing machine, the PLC communicates the actual position of the motors with the simulator using Modbus TCP/IP (Transmission Control Protocol and Internet Protocol).

As a result it was created a Modbus TCP/IP server on the soft PLC and a client on the simulator. On the server side, by using 'Codesys' for the controller's development there is only the need for setting the parameters needed for the communication to work, as 'Codesys' has already the Modbus TCP/IP protocol entirely implemented. On the client side, the parameters were set so the communications works seamlessly.

Overall, this system allows the user to test the machine's controller and visualise its behaviour and sequence of commands, helping on its implementation on a later phase.

5.3 Robot / Machinery / System Motion

Another key aspect of a simulator is its animation, i.e. what kind of movement it is able to recreate. Taking into consideration that the proposed simulator intends to serve all industrial work cells, a generic language to represent differential motion events, such as, linear or joint movements for robotic cells, position shifts for conveyor's parts as poses transformations for objects associated with common machinery (drill, saws, torches attached to linear axis...) was created.

A numeric algorithm (similar to IKFast from OpenRAVE) was also developed in order to fully describe the chain kinematics whether it may be a robotic work chain or a common work chain. Complementary to this algorithm, the simulator suggest a wide range of path planner options, both complete and optimal, such as the A* family, or probabilistic. As stated in the literature each has its advantages and disadvantages, and, thus, considerations should be had on which to be used for a given case.

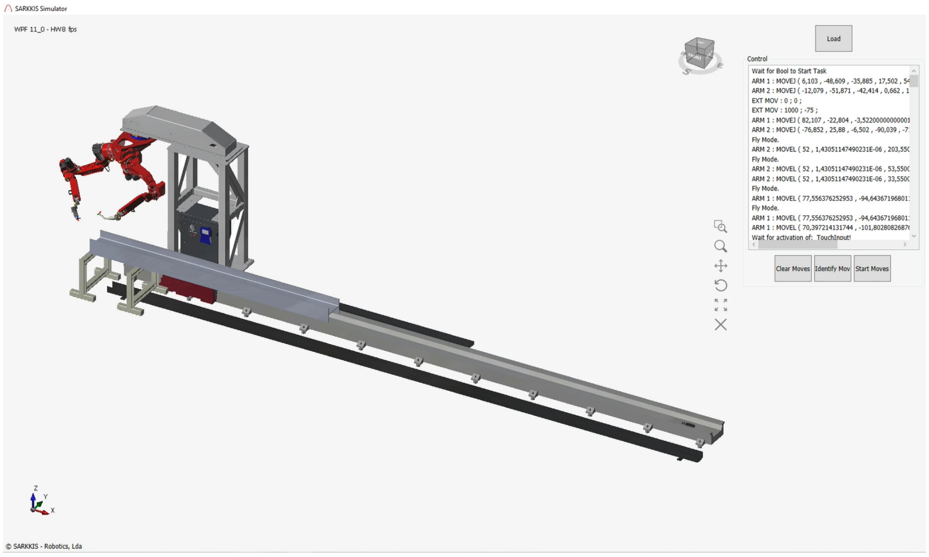
Essentially, the simulator grants the user the option to test and validate poses and movements manually or by using predefined planners.

5.4 Task / Job Loading and Testing

Manufacturing processes typically start with a design project of a part to be produced. Based on that project some information may be withdrawn in order to sequencing a set of points and operations to be had in an initial part in order to convert it to the projected one.

The idea behind this simulator is to accept multiple project formats, decrypt them and identify what positions and operations are coded within that project. Currently it is accepted both cutting and welding tasks or jobs (set of tasks).

Then, the decrypted files create a list of instruction to be graphically reproduced on the simulator. A example can be seen below in figure 5.



```
Wait for Bool to Start Task
ARM 1 : MOVEJ ( 6,103 , -48,609 , -35,885 , 17,502 , 54
ARM 2 : MOVEJ ( -12,079 , -51,871 , -42,414 , 0,662 , 1
EXT MOV : 0 ; 0 ;
EXT MOV : 1000 ; -75 ;
ARM 1 : MOVEJ ( 82,107 , -22,804 , -3,52200000000001
ARM 2 : MOVEJ ( -76,852 , 25,88 , -6,502 , -90,039 , -7
Fly Mode.
ARM 2 : MOVEL ( 52 , 1,43051147490231E-06 , 203,550
Fly Mode.
ARM 2 : MOVEL ( 52 , 1,43051147490231E-06 , 53,5500
ARM 2 : MOVEL ( 52 , 1,43051147490231E-06 , 33,5500
Fly Mode.
ARM 1 : MOVEL ( 77,556376252953 , -94,64367196801
Fly Mode.
ARM 1 : MOVEL ( 77,556376252953 , -94,64367196801
ARM 1 : MOVEL ( 70,397214131744 , -101,80280826876
Wait for activation of:  TouchInput!
```

Fig. 5. Example of loaded and decrypted Job.

5.5 Optimization

By supporting an accurate representation of the work cell, the proposed simulator provides the user with insight and validation tools on how to manufacture and assemble components. Additionally, the proposed simulator has a direct connection to a built-in add-in which performs optimisation in two major areas: components global position to maximise the operation range and components relative position to reach different poses via optimal trajectory.

This add-in allows a trained user to detect faults on previously defined robot or machinery code and it allows one to define new manufacturing sequences based on the optimised solution.

This feature is novel within the field of industry simulators and has the potential to revolutionize the way parts are manufactured and assembled.

6 Proposed Simulator & Industry 4.0

Recent industrial development and requirements lead to the implementation of the proposed simulator. The paradigm of digital twin simulation is crucial to define efficient robotic sequences that will untimely increase manufacturing productivity in real world work cells. This simulator was used as a virtual validator of two highly complex robotic cells. The first robotic cell was developed under the CLARiSSA European Project (see figure 6).

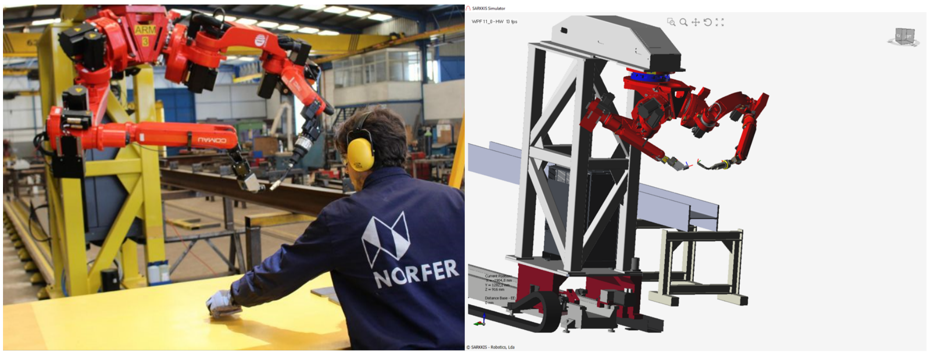


Fig. 6. Example of loaded and decrypted Job.

The simulator is of crucial importance since it allows for the robotic code to be interpreted by the simulator leading to a sequence of actions (as portrayed in figure 5). Each action was validated with the simulator platform and, once finished, a new robotic code was generated in order to be applied to the robotic work cell.

Another work cell that applies the digital twin methodology was used in the CoopWeld National Project (still ongoing), in which, the simulator optimisation tool provided insight on components' placement and ideal configuration of components to reach pre-defined poses.

Thus, this simulator is able provide the user with new and flexible state-of-the-art tools, such as optimisation techniques. It represents a key framework to be consider for future robotic applications.

7 Discussion and Future Perspectives

Throughout the project, it was developed a solution with enough flexibility to be valuable to both scientific research and industrial communities.

The proposed simulator presents the possibility of having a single framework capable of recreating specific work cells accurately. This is accomplished by using standardised resources which enable a easier adaptation to the software.

Another important aspect to consider when developing a software of this kind is time and computational effort. Although the software itself requires some hard drive space, due to the model database of multiple robots, external axis, conveyors, fixed parts and tools, it has low computation effort while assuring a fast response to a user request.

Furthermore, the five tiers of this simulator allow to mimic the work cell while validating orders, poses or even communication protocols safely and with reduced time effort.

For that reason, ideally the proposed software tool can be defined as a guarantee of function for a given process. The modularity and adaptability of such software also allows to be used externally as an add-in of a custom developed application.

Although the proposed simulator shows a wide range of applications and is efficient while performing them, there is still room for improvement. Despite that the simulator has an incorporated generic language to code movements and communication events, currently it only processes Cutting and Welding Jobs files. In future work, it would be interesting to extend the number to file types accepted and decrypted. Notwithstanding, a feature to insert those generic instructions manually is already available.

Parallel to the development of this simulation software, an optimisation tool is being developed. This development will prove to be beneficial as an important additional feature since that it will guarantee not only the validation of tasks but also the efficiency of those same tasks.

In conclusion, the work presented here depicts a simulation framework to be used in all industrial environments. This simulator can be valuable in several fields of automation, such as PLCs' controlled machinery (CNCs, Drilling, Cutting, Sawing, Painting machines) and in robotics due to its flexibility and modularity. This software was already tested in some major projects such as the European CLARiSSA Project and on the national based project, CoopWeld. It has also been used successfully to validate movements in complex work cells, some of them presented throughout the paper.

Acknowledgements

The research leading to these results has received funding from the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «POCI-01-0145-FEDER-006961». A special word to SARKKIS' robotics for their commitment in research and development of revolutionary state-of-the-art algorithms and for their contribution regarding software tools and engineering hours availability.

References

1. Angerer, A., Ehinger, C., Hoffmann, A., Reif, W., Reinhart, G.: Design of an automation system for preforming processes in aerospace industries. pp. 557–562 (2011)
2. Boschert, S., Rosen, R.: Digital twin-the simulation aspect (2016)
3. Buyya, R., Ranjan, R., Calheiros, R.: Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. pp. 1–11 (2009)
4. Collins, D., Lakshman, V., D'Arcy Collins, L.: Dynamic simulator for wip analysis in semiconductor manufacturing. pp. 71–74 (2001)
5. COSIMIR: Robots - robots for industry. (2011), http://industrial-robotics.co.uk/index.html
6. Craighead, J., Murphy, R., Burke, J., Goldiez, B.: A survey of commercial & open source unmanned vehicle simulators. pp. 852–857 (2007)
7. Ehinger, C., Reinhart, G.: Robot-based automation system for the flexible preforming of single-layer cut-outs in composite industry 8(5), 559–565 (2014)
8. Eyeshot: Overview: Why eyeshot (2017), https://www.devdept.com/Products/Eyeshot
9. Harris, A., Conrad, J.: Survey of popular robotics simulators, frameworks, and toolkits. pp. 243–249 (2011)
10. Issenberg, S., McGaghie, W., Hart, I., Mayer, J., Felner, J., Petrusa, E., Waugh, R., Brown, D., Safford, R., Gessner, I., Gordon, D., Ewy, G.: Simulation technology for health care professional skills training and assessment 282(9), 861–866 (1999)
11. Kumar, K., Reel, P.: Analysis of contemporary robotics simulators. pp. 661–665 (2011)
12. Min, K.S.: Automation and control systems technology in korean shipbuilding industry: The state of the art. vol. 17 (2008)
13. Petrovic, D., Roy, R., Petrovic, R.: Modelling and simulation of a supply chain in an uncertain environment 109(2), 299–309 (1998)
14. RoboDK: Simulate robot applications - robodk saves you time from design to production. (2015), https://robodk.com/
15. Schluse, M., Rossmann, J.: From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems (2016)
16. Valera, A., Gomez-Moreno, J., Sanchez, A., Ricolfe-Viala, C., Zotovic, R., Valles, M.: Industrial Robot Programming and UPnP Services Orchestration for the Automation of Factories. INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS 9 (OCT 11 2012)
17. Veiga, G., Pires, J.N., Nilsson, K.: Experiments with service-oriented architectures for industrial robotic cells programming. ROBOTICS AND COMPUTER-INTEGRATED MANUFACTURING 25(4-5), 746–755 (AUG-OCT 2009)