Framework Architecture for Manufacturing Systems Emulation

Maximilian Nicolae
Dept. of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
max.nicolae@upb.ro

Stefan Mocanu
Dept. of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
stefan.mocanu@upb.ro

Mihai Craciunescu
Dept. of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
mihai.craciunescu@upb.ro

Radu Dobrescu
Dept. of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
radu.dobrescu@upb.ro

***Abstract*— The paper presents the architecture of an emulation framework and the associated services that provide the computing resources and services for the design of Manufacturing Systems with regards to Industry 4.0 requirements.**

***Keywords*— Virtual Manufacturing, embedded systems, predictive maintenance, Computer Integrated Manufacturing, Hardware in the Loop.**

I. Introduction

Manufacturing systems are closed related to new product development (NPD), therefore it is normal that many steps involved in NPD should capture also the manufacturing stage. This is often referred as Design for Manufacturing (DFM). Besides this, manufacturing has also other concerns related to optimization and maintenance. The last one will be proper addressed in this paper with regards to predictive maintenance.

Nowadays in most of the situations when manufacturing systems are addressed from research perspective the word “virtual” is also present. The Virtual Manufacturing (VM) field, even though it was introduced over 20 years ago, is in a continuous growth as there are many benefits to consider. The main point is that through VM the manufacturing technology, which normally has a large inertia, can keep up with the faster development of the information technology and communication field. This problem is multi-disciplinary and extremely complex as VM integrates new technologies and concepts such as embedded systems, grid computing, model-driven concepts, cloud computing and sensor networks. The specialists’ opinion is that VM technologies will radically change the current approach on manufacturing and will have a great deal of impact [1]. As a most simple demonstrations, a straight forward indicator for a new product performance is *time to market*. This is one of the key factors in determining the level of market performance of a certain product or service, and is one of the main sections of investment by companies, mainly to take advantage of time frame opportunities ([2], [3], [4]). In this manner, the time it takes from designing a product or service to actually launching the product will decrease dramatically by using VM. Considering the available IT technologies, the main problem in developing virtual manufacturing systems doesn't come from the user interface design, user-friendliness, or ergonomics. The main concerns are linked to the precision of the models and the simulation (even in comparison with simulation times) and mostly in enveloping the real life new technologies in the virtual environment. These problems also derive from fact that a piece of fabrication equipment is a complex system, to development of which took part a diverse team of specialists. In the real system scenario, the components interface is rigidly defined and in accordance with standards. Each of the composing (real) modules has its own time base, so that the parallel functioning of modules becomes natural. The moment when the actual system is simulated, things change considerably. There are continuous time systems that are simulated on discrete time platforms. There is a combination of modules, some continuous others discrete, that must be synchronized in simulation.

The benefits of VM can lead also to aspects like designing and simulation of safety and ergonomically operations for workers at their workplace [13]. Additionally, VM can be combined with Virtual Reality (VR) and used for Augmented Reality in the site [14] or for training the personnel [15].

Among current fields of research that address also the VM there can be enumerated Cyber Physical Systems (CPS), Industry 4.0 and Virtual Commissioning.

In the aforementioned context, this paper presents an ongoing project that aims to develop a framework and a software platform with regards to object-oriented programming, capable of integrate the advancements in

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

modeling and simulation of manufacturing equipment together with their real counterparts. The idea of integrating the real devices as Hardware in the Loop (HiL) adds the emulation functionality. The architecture of this framework is the foundation of the success for the entire project. It is less probable that when the framework will be ready, it will become a standard in VM, but with inspiration from hobby platform like Arduino and Raspberry Pi, our proposed framework could become a stage in the final goal of total integration.

The paper is structured as follow, after the introduction with the overall goals, similar approaches are investigated, then the proposed architecture is described fallowed by a short feasibility study of the methodology. In the end some conclusions are presented.

II. Similar Approaches

There are major advancements in all of the components of VM but there is no environment that seamless integrates all of the necessary components that will lead to a concurrent engineering approach. The idea of concurrent engineering (for product development) resides in objectives like waste elimination, DFM, first time right, zero defects, etc. This approach can be easily followed on Electronic Design Automation (EDA) tools where the new product design stages are overlapping. One change made in the schematic (design of the operation principle) will instantaneously provide the corresponding changes at the following stages (related to physical parameters and manufacturing considerations). By having an emulation and simulation environment the real conditions of fabrications can be created. In Fig. 1 (adapted from [5]) one can see how, while designing the schematic hardware, the code can be developed and tested. In the same time the layout is developed in accordance with fabrication requirements and signals integrity. Possible mistakes are signaled at this stage by automatic validation provided by the tool (Altium Designer). Then, the assembly is simulated and if some problems are identified, the correction costs will be very low comparative with the previous scenario (rework etc.). Based on the layout and assembly specifications there are generated outputs for fabrications equipment (Computer Integrated Manufacturing – CIM). And if any of the events from the above scenario would happen, then a change in a previous stage will generate automatic updates in the following stages thus the cost of redesign will be extremely low comparative with the first (classical) design methodology.

The allover desired agile methodology which needs to identify and eliminate redundant information and operations (referred as waste) but in the context of innovation process triggered the need for specialized and unified frameworks for new product and service development (NPSD). An interesting approach in this direction is proposed in [6] where the idea of artifact is introduced representing an element of information carrier.

Resuming strictly to manufacturing process, probably the domain that provides most of the solutions for the objectives

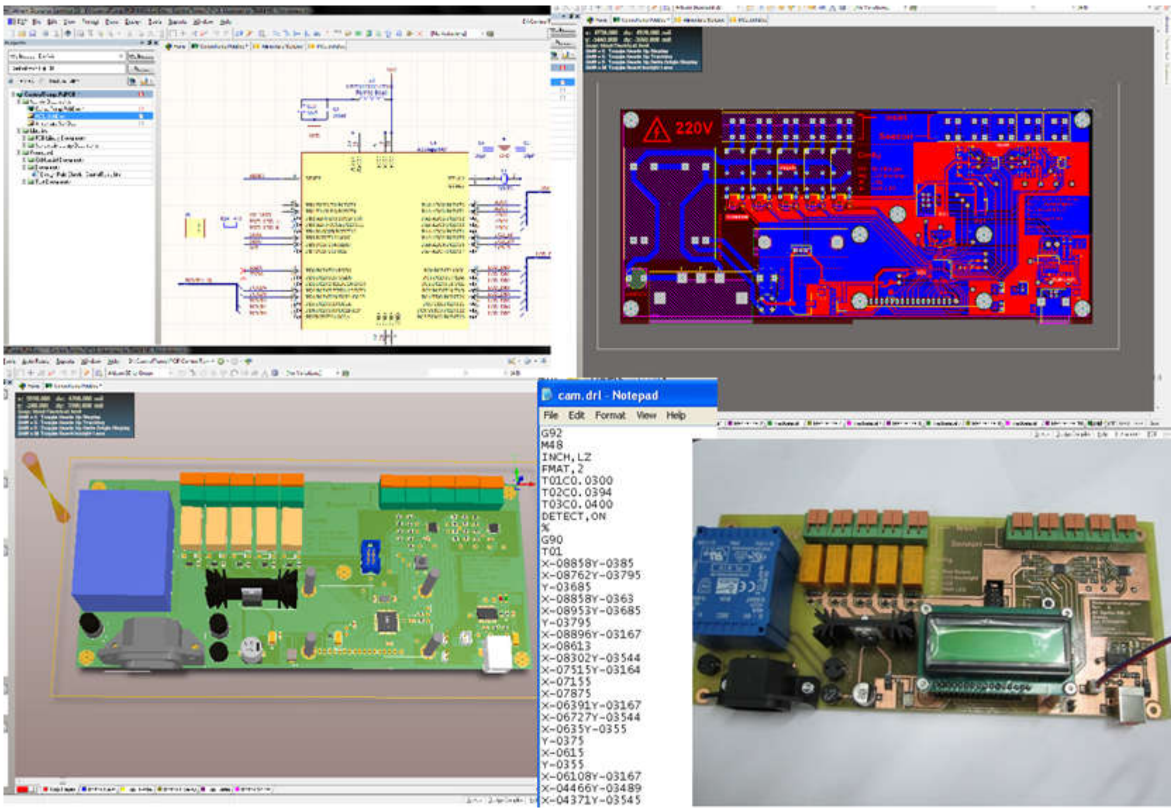

Fig. 1. Concurrent engineering in Altium Designer EDA (source: [5])

related to concurrent engineering is Virtual Commissioning (VM). Paper [7] provides a good introduction of VC as a tool for accelerating the system integration in manufacturing presenting also a suitable case study. VC promotes as advantages indicators like product quality improvement, reduction in the system ramp-up time, and a short time to market. More advantages from economical point of view can be found in [8]. Even widely promoted VC is still under development with many things remaining to address [9]. Adding to this the fact that software applications that provides VC functionality are very expensive (even the academic licenses) we considered that the need for the framework we target is well motivated.

Outside the domain of VC there are other "custom" implementations of VM. In [10] the Hardware in the Loop (HiL) method is generalized in what is called Hybrid Process Simulation (HPS) that combines both simulated and real components. The authors also identify the need for an HPS ontology and provides a formalized method for its applications together with three examples of applications. A similar framework is presented in [11] where so called "clever components" are used for automatic generation of control logic (PLC programs) and model plants. A comprehensive framework for CPS is presented in [12] which refers to inhouse tools developed by the Automation System Group in the Warwick Manufacturing Group (Univ. Warwick).

Further we looked to similar approaches addressing only one machine tool from the entire factory, considering that the methods, if scalable, could be extended to a set of machineries. The key word in these implementations is hardware in the loop (HiL). In this direction, the work presented in [16] reiterates the need for fast development of machine tools and demonstrates how HiL could be used to optimize the design process.

Another field that have developed in the last period is the robot simulation environment. Besides its main intent of visualizing the behavior of mobile robots by testing their commands, this field could provide also the ability to integrate in a broader framework for virtual manufacturing together with running in parallel with the real process, thus, contributing to the implementation of new concept like digital tween. Some examples of such implementation but in other domains are [17], which extends the driver simulator capabilities by integrating HiL testbeds, and [18] which help patients with their rehabilitation exercises.

In the end but not less important, another driver towards the desired VM is the revolution that 3D printers brought in the landscape of rapid prototyping and hobby activities. In this context, [19] discusses the additive manufacturing (AM) from the simulation perspective.

All the above approaches motivate the need for an integrated methodology that will lead to use the advancements in domains like VC, agile manufacturing, HiL, AM, EDA and CIM but on more natural bases, with open source software and copying the model of going viral platforms like Arduino and Raspberry Pi. The final goal would be to "force" the vendors of machinery tools and manufacturing factories to provide open models for their equipment.

III. Considerations Regarding the Architecture of the Framework

One of the important thing when someone has two design an architecture is the tradeoff between centralized and decentralized architecture. The idea of decentralizing seems to get us close to the natural model of things. On the other hand, the decentralization comes with the price of increasing the overhead and the complexity of the implementation. As well, the communication between the parts of the decentralized structure poses another set of problems. Even the trend is to decentralize, most of the applications that provides VC have a centralized architecture. Indeed, some of their components like the physics engine can run on parallel platforms (virtual machines, GPUs etc), the component itself is centralized. From our experience, the main reason why such component is thought centralized is because of the strong coupled systems that it should simulate.

On the other hand, nowadays, when designing systems, it become normal to get inspired from biological counterparts, considering that the evolution had its time to optimize those "systems". It is difficult for engineers to fully do that and even for mix teams of neuroscientist, biologists, physicist and engineers. For far as we know there are no such big breakthroughs in how the brain functions. Therefore, for the moment (and applicable to this architecture) we used only engineering methods. Things will probably change in the near future when some machine learning algorithm will be able to give better results than an optimum and robust designed control algorithm that operates in a noisy environment. By optimum here we mean in respect with an approximation model of the process.

Having the above in the consideration, our architecture will be a decentralized one, that will try to mimic the real parts involved in the manufacturing systems by dividing the physical process (concerning and responsible for each "physical component") to a virtual component (that we will further call node). One node, function of its complexity and degree of coupling with other nodes, can operate either on one application task, or one operating system process, going upper to a virtual machine or dedicated hardware. This method is not new and resembles the way how some of the actual models from simulation platforms interact. There are objects which communicate to each other. The fact that is not something new comes with the benefit that there can be used a lot of things that already exists. The novelty is the proposal to use one hardware for each virtual component or group of components as mentioned above. As an idea of the size of the intended hardware, someone could think to small single-board computers like Raspberry Pi, Intel Edison, various System on Chip (SoC) and those system that will follow them in the future of integration. Despite some disadvantages of this approach that are discussed later, there are also two major benefits:

1. The system will increase its scalability without the need of investing in expensive hardware like high performance computers (HPC). Being cheap it will be adopted fast by many engineers (or students) and it

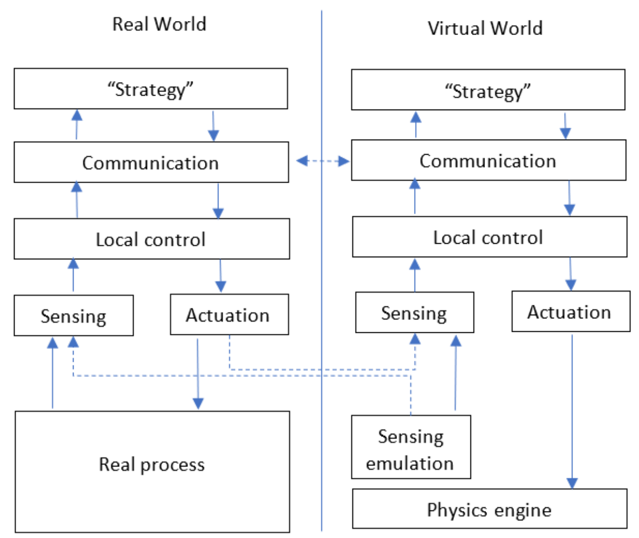

Fig. 2. Layered structure of the manufacturing components

will facilitate work in team by simply connecting the nodes (even remotely located).

2. The hardware module that runs the virtual nodes can be used to run in parallel with the real component (HiL) and provide information for predictive maintenance.

The main disadvantage that should be overcome is that generated by the systems that are strongly coupled which, by breaking them apart, will put a lot of pressure on the communication. By considering the physical location of the components of a manufacturing line, the components are not to be strongly coupled as the interface among them is clear defined even from their design stage. Even the process they influence should not be near close to what is happening in complex life form. Therefore, the process involved in manufacturing systems are loosely coupled and permits the simulation and emulation on a decentralized architecture. Practically, the single strong couple system that the actual architectures have is the physical engine. Our approach is to break the physical engine in more engines and find a way to synchronize them. In this way, the system will get close to its real nature and the hardware implementation will be less demandable of high performance resources.

Each bounded component of the manufacturing system (e.g. CNC, robot joints, etc.) is interfaced with a controller through sensor and actuators. The controller is implemented on a specific hardware. The analogy between the real (physical) world and the virtual world is illustrated in Fig. 2 where a layered structure is presented. In order to implement methods like HiL, the real process has to be decoupled from some part of the actuation and sensing and a clear interface should be defined.

If considering only the real world, the manufacturing tool is composed of sensors (the sensing layer), actuators (actuation layer), the local logics (local control layer), communication modules that interface with upper layer (e.g. strategy layer). The dashed lines show the information stream for HiL. Thus, the real device can be connected the virtual process (simulated by the physical engine) through the virtual sensing layer, virtual actuation layer and sensing emulator.

To achieve the purposed objectives, the virtual counterpart should resemble the same layers and emulate the interface between them. Moreover, for concurrent engineering goals, some parts of the code that control the logics should be usable without modification. This requires other level of abstraction that are often met in new software and embedded architectures. Among these levels at least the hardware abstraction layer (HAL) should be defined together with some adaptation layers. These will ensure the possibility to write code (and reuse code) that are hardware independent.

The proposed framework that could implement this architecture is composed of the following components:

- Gazebo simulator or MORSE (more details on comparison of several robot simulators and the reason behind this choice can be found in [20][21][22])
- The Robot Operating System (ROS) which is a framework of its own that promote the development of software for robots in a hardware independent manner. An example of how to interface ROS with Gazebo is described in [23].
- The hardware intended to emulate the controller and the process (the virtual component) could be composed of small single-board computers running Linux as a suitable solution both from the speed of development and ease of integration. One problem with Linux is that it is not a real time operating system intended for automation processes. The Real Time Application Interface for Linux (RTAI) adds the required services for Linux to behave like an industrial real time operating system.

IV. Evaluation of the Feasibility

We started this framework architecture based on previous experience with Virtual Manufacturing Line (VML). We had to integrate a Multiphysics Software Application (HYPAS) in a VML platform (Fig. 3 [24]). The HYPAS was design during ’80s and is very fast, therefore, comparing the PC in that period with actual small single-board computers like Raspberry Pi, Orange Pi or Intel Edison, the last ones win in performance by far. Therefore, a simple architecture for HiL system came in mind, by taking the models form HYPAS which is Windows based together with Matlab/Simulink modules and port them to Octave in Linux we can simulate complex machinery tools used in manufacturing lines in order to provide predictive maintenance. One problem, as mentioned in the previous section was the real time constrain. To overcome this problem, we integrated the hardware peripheral interrupts where possible. Later, we considered the aspects presented in the introduction section and now we come up with this proposal.

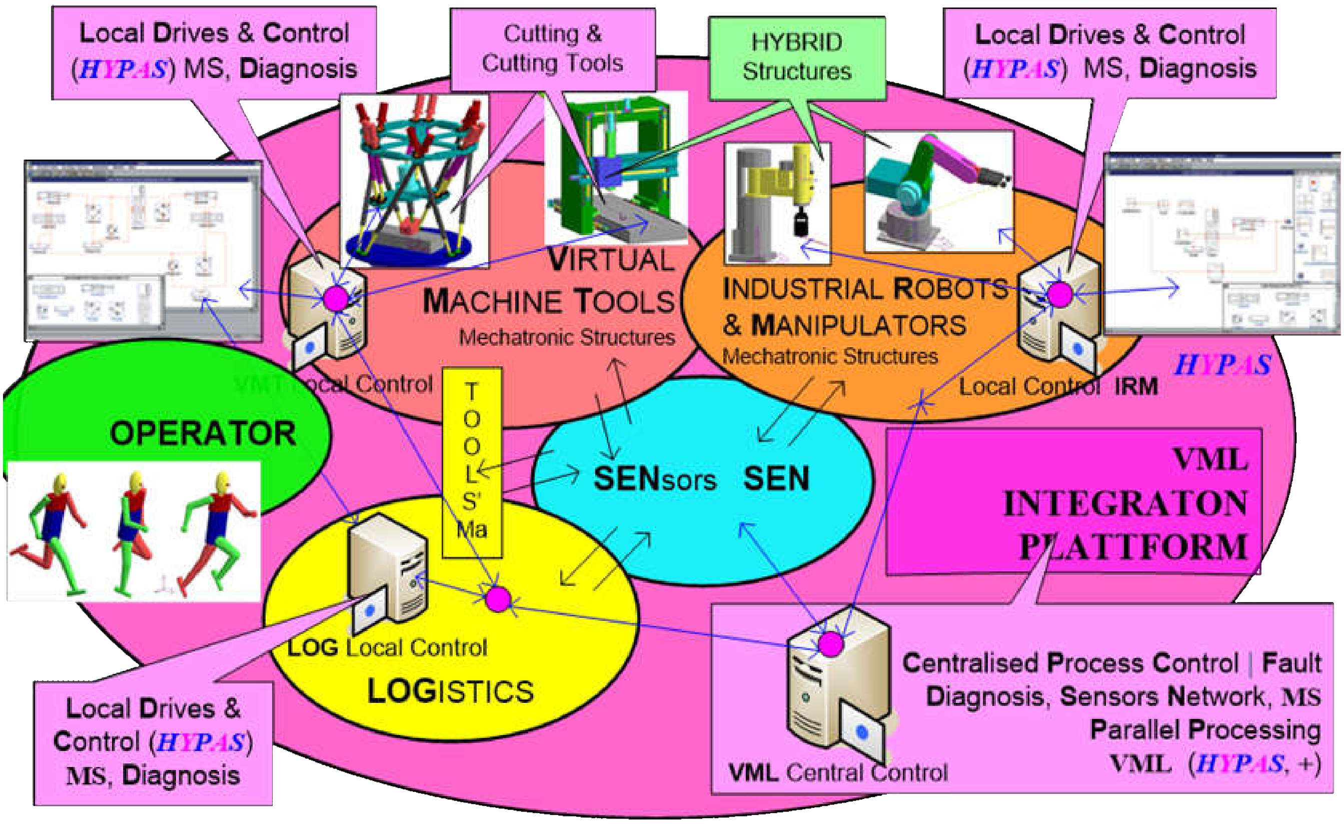

Fig. 3. Integrated environment for VML using HYPAS application (source: [24])

As mentioned, implementations of HiL systems requires real time constrains. Papers like [25] and [26] demonstrates the possibility to implement complex systems on regular hardware using Linux and RTAI. In [27], one simulation is implemented on Raspberry Pi to be used to control the steam turbine model with shaft and generator, but it can be used on wide range of complex physical models as the authors state.

Regarding the Gazebo simulator and ROS framework, the literature abounds in papers claiming suitable results in accordance with our requirements related these platforms. Some examples are [28] where a five degree of freedom (DOF) visual servoing robot is presented with eye-in-hand configuration. In [29] is evaluated the error that Gazebo and ROS have in comparation with the real robot in a trajectory tracking application. Again, the results are very encouraging. [22] addresses also the multi-robot system and find some limitations, but nothing to be concerned as mentioned before, we intended to divide the physics engine and find a way to synchronize the parts. In this way, the upper limit of the number of robots that a single physics engine could support will not be an issue anymore.

Adding to all of this the performance and functionality of modules like *numpi* and *scipi* found in Python scripting, we could conclude that this approach has all the chances to succeed in its objectives to provide a framework for manufacturing system emulation, ensuring accurate simulation and emulation based on real time HiL, and, in the end, accelerate the prototyping of products and addresses student laboratories with low resource requirements.

Let note also that this framework allows an intelligent approach for managing the manufacturing process assets in order to replace the classic maintenance of the assets with one more advanced. If the classic maintenance generally means replacing components after a period of running, nowadays are developed the concepts of proactive, predictive, prescriptive maintenance which involves a continuous monitoring of the assets for tacking the right actions in order to improve the equipment lifecycle. The continuous monitoring means real time data collection and, at the end, facing the Big Data concept results analytics and virtual transducers.

V. Conclusions and Further Development

This work proposes a framework for manufacturing system design and emulation able to integrate the state of the art technological advancements for the development of new applications with novel innovative directions that combine IoT technologies, new sensors, transducers, measurement tools, artificial intelligence (AI) techniques, big data analytics and uncertainty management. It prepares a next step to the development of a virtual production system that can implement machine learning technics that could compete with human creativity. This last part could change the classical optimization paradigm based on mathematical models to a paradigm of optimization based on learning and creativity that could lead to what is now referred as Virtual Production Intelligence.

ACKNOWLEDGMENT

This work was partially supported by the Romanian Ministry of Education and Research under grant 78PCCDI/2018-CIDSACTEH.

REFERENCES

[1] Xi Junjie, "Research on Virtual Manufacturing and System Structure of Complex Products," Information Management, Innovation Management and Industrial Engineering (ICIII), 2010 International Conference on , vol.4, no., pp.616,619, 26-28 Nov. 2010.

[2] Y-J.Chen, Y-M.Chen, and H-C.Chu, "Enabling collaborative product design through distributed engineering knowledge management", Computers in Industry, ELSEVIER, 2008, (59), pp.395-409

[3] Y. Shen, S.K. Ong, A.Y.C. Nee, "Product information visualization and augmentation in collaborative design", Computer-Aided Design, ELSEVIER, 2008, (40), pp.963- 974.

[4] H. Keming, W. Dahu, Z. Tong, Z. Jiaolong, "Research on a virtual manufacturing system," Mechanic Automation and Control Engineering (MACE), 2010 International Conference on , vol., no., pp.3057,3060

[5] M. Nicolae, "Service Operations Management from System Engineering Persepctive", Ed. Printech 2015, pp. 38

[6] A. I. Böhmer, M. Meinzinger, R. Hostettler, A. Knoll and U. Lindemann, "Towards a framework for agile development of physical products influence of artifacts and methods," 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Funchal, 2017, pp. 237-245

[7] H. Vermaak and J. Niemann, "Virtual commissioning: A tool to ensure effective system integration," 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), Donostia-San Sebastian, 2017, pp. 1-6.

[8] N. Shahim and C. Møller, "Economic justification of Virtual Commissioning in automation industry," 2016 Winter Simulation Conference (WSC), Washington, DC, 2016, pp. 2430-2441.

[9] S. Süß et al., "Test methodology for virtual commissioning based on behaviour simulation of production systems," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, 2016, pp. 1-9.

[10] W. S. Harrison, D. M. Tilbury and C. Yuan, "From Hardware-in-the-Loop to Hybrid Process Simulation: An Ontology for the Implementation Phase of a Manufacturing System," in IEEE Transactions on Automation Science and Engineering, vol. 9, no. 1, pp. 96-109, Jan. 2012.

[11] A. Khan, P. Falkman and M. Fabian, "Virtual engineering framework for automatic generation of control logic including safety," 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, 2017, pp. 648-653.

[12] R. Harrison, D. Vera and B. Ahmad, "Engineering Methods and Tools for Cyber–Physical Automation Systems," in Proceedings of the IEEE, vol. 104, no. 5, pp. 973-985, May 2016.

[13] H. Xin, H. Lu, W. Luo and H. Shao, "Research on assembly modeling process based on virtual manufacturing interactive application technology," 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, 2017, pp. 363-367.

[14] C. J. Turner, W. Hutabarat, J. Oyekan and A. Tiwari, "Discrete Event Simulation and Virtual Reality Use in Industry: New Opportunities and Future Trends," in IEEE Transactions on Human-Machine Systems, vol. 46, no. 6, pp. 882-894, Dec. 2016.

[15] F. Pürzel, P. Klimant, V. Wittstock, M. Kuhl, "Real NC Control Unit and Virtual Machine to Improve Operator Training", 2013 International Conference on Virtual and Augmented Reality in Education, Procedia Computer Science 25 (2013) 98 – 107, Elsevier.

[16] M. Witt, P. Klimant, "Hardware-in-the-Loop Machine Simulation for Modular Machine Tools", The 15th CIRP Conference on Modelling of Machining Operations - Procedia CIRP 31 (2015) 76 – 81, Elsevier

[17] K. S. Swanson, A. A. Brown, S. N. Brennan and C. M. LaJambe, "Extending driving simulator capabilities toward Hardware-in-the-Loop testbeds and remote vehicle interfaces," 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, 2013, pp. 122-127.

[18] Z. Du, Y. Sun, Y. Su and W. Dong, "A ROS/Gazebo based method in developing virtual training scene for upper limb rehabilitation," 2014 IEEE International Conference on Progress in Informatics and Computing, Shanghai, 2014, pp. 307-311.

[19] G. Q. Zhang, A. Spaak, C. Martinez, D. T. Lasko, Biao Zhang and T. A. Fuhlbrigge, "Robotic additive manufacturing process simulation - towards design and analysis with building parameter in consideration," 2016 IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, 2016, pp. 609-613.

[20] P. Castillo-Pizarro, T. V. Arredondo and M. Torres-Torriti, "Introductory Survey to Open-Source Mobile Robot Simulation Software," 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting, Sao Bernardo do Campo, 2010, pp. 150-155.

[21] G. Echeverria, N. Lassabe, A. Degroote and S. Lemaignan, "Modular open robots simulation engine: MORSE," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 46-51.

[22] F. M. Noori, D. Portugal, R. P. Rocha and M. S. Couceiro, "On 3D simulators for multi-robot systems in ROS: MORSE or Gazebo?," 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, 2017, pp. 19-24.

[23] K. Takaya, T. Asai, V. Kroumov and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2016, pp. 96-101.

[24] Fl. Ionescu, C. Constantin, M. Nicolae, Vl. Kotev – Object Oriented Virtual Manufacturing Lines – a Most Efficient and Flexible Tool for RDE and Process Control – Proceedings of The 5th International Conference on Manufacturing Engineering 2014, pp.119-132.

[25] L. Lou and K. Kühnlenz, "Hardware-in-the-loop development and real-time testing for precision motion control under RTAI," 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 2014, pp. 312-316.

[26] J. M. Serna, C. Fory, A. M. González-Vargas and A. Soria-López, "Real-time control platform based on free software and USB communication," 2016 IEEE Conference on Computer Aided Control System Design (CACSD), Buenos Aires, 2016, pp. 845-849.

[27] J. Reitinger, P. Balda and M. Schlegel, "Steam turbine hardware in the loop simulation," 2017 21st International Conference on Process Control (PC), Strbske Pleso, 2017, pp. 380-385.

[28] P. M. Khiabani, B. S. Aghdam, J. Ramezanzadeh and H. D. Taghirad, "Visual servoing simulator by using ROS and Gazebo," 2016 4th International Conference on Robotics and Mechatronics (ICROM), Tehran, 2016, pp. 308-312.

[29] A. Renawi, M. A. Jaradat and M. Abdel-Hafez, "ROS validation for non-holonomic differential robot modeling and control: Case study: Kobuki robot trajectory tracking controller," 2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Sharjah, 2017, pp. 1-5.