Sensors & Transducers, Vol. 261, Issue 2, June 2023, pp. 61-66

Sensors & Transducers
Published by IFSA Publishing, S. L., 2023
http://www.sensorsportal.com

Digitalization of Aeronautic Painting Shop Floors for Improved Commissioning Activities

[1, *] **J. H. Lugo,** [2,*] **A. Abassi, R. Balderas and N. Lassabe**
Capgemini Engineering, Research and Innovation Department, Toulouse, France
E-mail: jesus-hiram.lugo-calles@capgemini.com, ahlam.abassi@capgemini.com

Received: 24 April 2023 Accepted: 23 June 2023 Published: 26 June 2023

Abstract: Industrial commissioning plays a critical role in ensuring the safe and efficient operation of facilities and minimizes downtime and maintenance costs over their lifetime. To extend and adjust commissioning capabilities, Virtual Commissioning uses digital models of devices and processes to verify, validate, and optimize code programming, and component selection. To perform the validation process, a simulation involving control devices and process digital twins is required, leading to inherent computational complexity. Distributed simulation approach allows for simulation of complex systems by breaking down a large simulation into smaller, manageable parts that can be run simultaneously on separate processors, while still preserving the overall behavior and interactions of the system being simulated. This paper presents a distributed Virtual Commissioning solution for a spray paint process presented in UAV painting shop floor. The methodology for developing the implementation is described in detail: greenfield scenario generation, automation process, software toolchain development, selection of communication protocols, re-use of digital twins for extended applications, and complexity analysis. A set of 3d scenarios is used to demonstrate the result's performance.

Keywords: Virtual Commissioning, Automation, Toolchain, Distributed simulation.

---

1. Introduction

Virtual commissioning (VC) is a method that allows the user to propose a commissioning solution using digital models of industrial systems and devices to accelerate and improve traditional processes. [1-5] The main benefits obtained from VC are:

- Early simulation, and verification and validation (V&V) of machine code by using different simulation approaches Hardware-in-the-Loop, Software-in-the-Loop, and Model-in-the-Loop (HiL, SiL, MiL) without impacting production, see Fig. 1.
- Higher quality SW with optimized controls architecture and programming;
- Faster changes implementation while maintaining quality and reducing risks;
- Shortened overall commissioning time;
- Decrease manufacturing system lead time.

VC solution methodologies have been widely studied, and standardization efforts regarding the complexity of the involved digital twin specification and V&V process used during the VC activities have been developed. [6] It is worth noting that the more specifications and realistic details are considered for developing the solution, the more accurate the representation of the system to be commissioned would be. However, the simulation of highly complex scenarios could lead to poor computer performance, hindering and sometimes, inhibiting a reliable validation process. To improve this, the VC developer could evaluate simulation complexity [7] to identify elevated consuming processes and objects in the scenario.

The aeronautic industry is characterized by its highest quality manufacturing and safety standards, leading to a constant demand of precise and accurate automated systems. This situation opens the door to the proposal of multiple Industry 4.0 green-field scenarios where robots can collaborate with humans to improve manufacturing processes.

This article is an extension of [1] as it details the use case development and the approach used to model the system and design the architecture for the virtual environment.

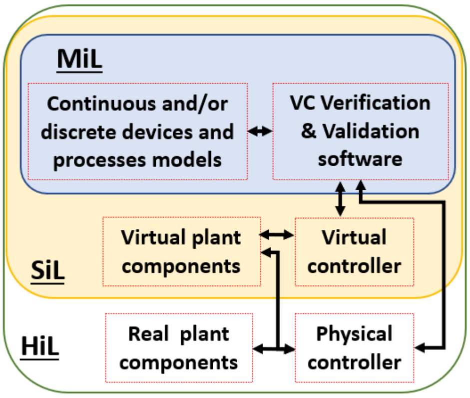

Fig. 1. Simulation approaches used for V&V purposes and their interactions. This graph demonstrates how MiL can be extended to more complex approaches like SiL and HiL.

2. Distributed Simulation Approach

Distributed simulation architectures provide better performance of industrial system representations. Several projects have successfully implemented this simulation scheme in the robotics field with different goals in mind: robot specific behaviors, either individually, as a swarm or within an organism [8]; distributed simulation system to validate the control of the real robot through the connection of GUI and digital twins [9]. It is worth noting that this approach implies issues related to the hardware specifications of each computer in terms of available memory (RAM and ROM) as well as execution time, affecting data synchronization [10], and the additional costs of using extra hardware and the energy to power it. According to the results obtained from the previous studies, the following outcomes are expected:

- Improvement of current toolchain simulation parameters (i.e., time slices, cycle times, mathematical solvers, etc.).
- Selection of alternative SW/HW tools.
- Division of complex simulation into simpler ones.
- Communication protocol parameters selection and adjustment.

3. Use case Description and Development

The use case, which is an aeronautic painting shopfloor taken from our previous work presented in [11], targets to spray paint a fairing a fairing used as a motor and propeller support of a UAV taxi, see Fig. 2. The industrial cell consists of four painting robots (equipped with smart painting nozzles), one or two smart conveyors (depends on the implementation as we will show later), and a safety system (composed of beam laser sensors), as shown in Fig. 3, and Table 1. With the development of this scenario, we try to address the need of hybrid automated environments, where robots can coexist and human workers in a safe manner; and robotize legacy industrial processes. The painting process is described as follows:

1. A fairing is placed on conveyor C1
2. C1 moves the fairing until it reaches the position sensor P1, stopping C1 and activating the spray-painting process for robots RL1 and RR1.
3. After the painting activity of RL1 and RR1 is completed, C1 moves the fairing to conveyor C2 (or remains on C1 if the cell only uses one conveyor).
4. C2 moves the fairing until it reaches the position sensor P2, stopping C2 and activating the spray-painting p After the painting activity of RL2 and RR2 is completed, C2 moves the fairing to its farthest end.
5. The safety system is always running in a parallel thread, allowing it to be activated at any time and provide an emergency stop reaction for a single device or the whole cell.

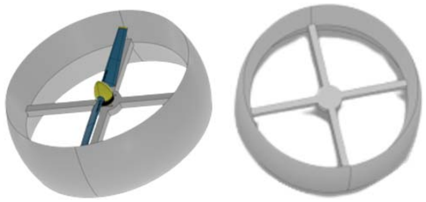

Fig. 2. Faring 3D model developed in DS CATIA. Left: propeller assembly composed of fairing, electric motor, and propeller. Right: propeller fairing before being machined.

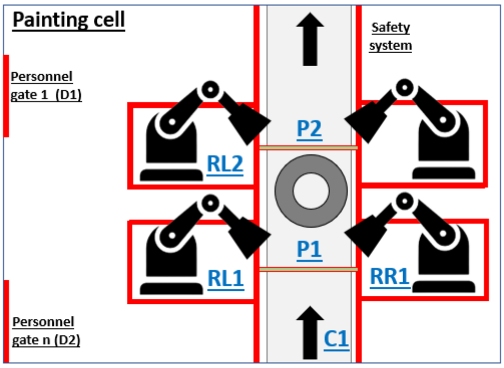

Fig. 3. Painting process schematic diagram.

To ease the modeling of the cell, we follow a Model-Based Systems Engineering approach in which, we analyze the systems from a top-down perspective, refer to Fig. 4. This method provides a practical view of the systems [12], their composing subsystems, and the communication between them, targeting reusability of the models for later projects.

Table 1. BOM for each simulation scenario.

Component	Description	DELMIA	CoppeliaSim
Industrial robotic manipulator	6 DoF spray paint robotic arm	ABB IRB 6400R/2.5-150 Quantity: 4	UR10 : Universal Robotics Arms, Quantity: 4
Paint nozzle	Customized simulated spray paint nozzle	Quantity: 4	Quantity:4
Conveyor	Customized simulated conveyor	(11m x 0.3m) Quantity: 1	(4m x 3m) Quantity: 2
Position sensors	Simulated laser beam position sensor	Quantity:2 Diameter:10 mm	Quantity:2 Diameter:10 mm
Safety sensors	Simulated laser beam safety sensor	Quantity:18	Quantity:15
Worker mannikin	Programmable dummy to represent	Quantity:1	Quantity:1
Gates	Entry levels to the cell	Quantity:4	Quantity:1

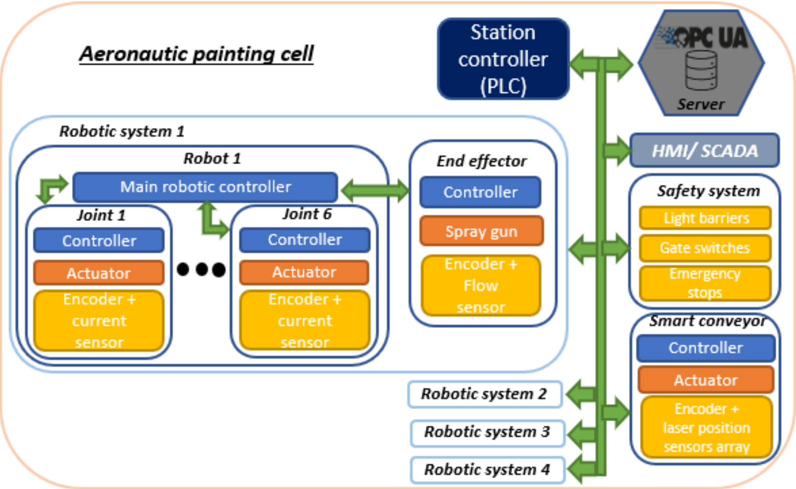

Fig. 4. Top-down view of painting cell systems. This diagram displays the elements that compose each element of the cell and how they communicate to each other.

3.1. Toolchain Design and Development

To design an efficient toolchain for VC purposes, we use the layered network shown in Fig. 5 as a template, where the different layers encompass the IT and OT according to their use and purpose. It is worth noting that the semantic layer is being represented by an OPC-UA server, which provides plenty of flexibility on the communications with the different technological levels of devices.

The scenario was developed simultaneously on two different CAM software: DELMIA and CoppeliaSim, as shown in Figs. 6 and 7. Both software are meant for similar purposes but target different applications: industrial manufacturing for the first one, and robotics programming for the second. To cope with the interoperability between our modules, we implemented an OPC-UA network, consisting of a server running on SIMIT SP, and using the client capabilities of the rest of the software to attach to it. OPC-UA supports soft real-time data exchange, event handling, and historical data access, making it a suitable choice for industrial automation, control systems, and other applications that require reliable data exchange and contain self-controlled and internal safety parameters reactions and behaviors. A PLC is used to synchronize the tasks of all smart devices and control the safety system. The PLC can share data using OPC-UA, allowing a simple data mapping between control and semantic layers.

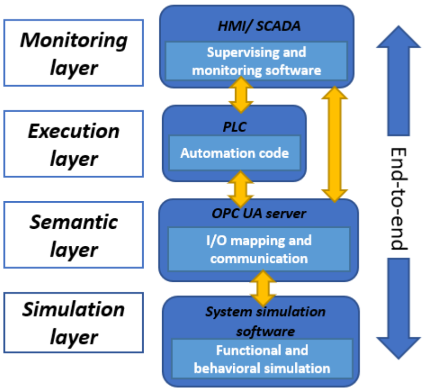

Fig. 5. Mapping and network connection between SiL layers.

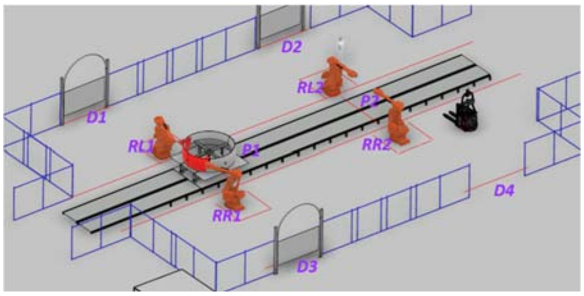

Fig. 6. Aeronautic painting cell 1 developed with Dassault Systemes tools: DELMIA 3Dexperience and ControlBuild. This cell is characterized using a single conveyor and four doors used by the shopfloor workers and forklifts.

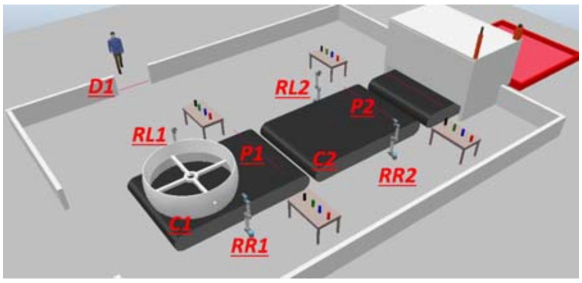

Fig. 7. Aeronautic painting cell 2 developed with CoppeliaSim EDU version and Python code. This scenario has a third conveyor that takes the painted fairing to an electric oven (white box) to accelerate the drying process.

The safety system was equally implemented on both scenarios, simple detection of position sensors (laser beams) around the robots and at the entry gates of each scenario in addition to the emergency stop button. It is worth noting that safety behaviors are programmed in every smart device to provide a fast response to events locally and send the alarm and safety signals to the cell controller for synchronization and manufacturing cell reaction.

The painting process is defined in different manner according to the CAM used: surface modification using the robot surface simulation application in the first scenario and particles projection using a simple LUA code for the second one.

Both scenarios were designed and tested individually using the toolchain shown in Fig. 8, where the digital twin of the process is connected via Control Build as a client to an OPC-UA server for the first scenario and via python in the second digital twin, working as a database, that receives and distributes information to other clients: a S7-1500 PLC (running on Siemens TIA Portal V17 and PLCSim Advanced 4.0) and a supervisory system consisting of a HMI system (Siemens WinCC) and SCADA platform (Fernhill).

To obtain better performance, we decided to divide the main system into subsystems. Thanks to OPC-UA technology, we could implement a stable industrial network between different computers allowing the interconnection of multiple simulations, see Fig. 9. The proposed toolchain consists of two workstations for the heavy computing processes and a remote PC to display system variables.

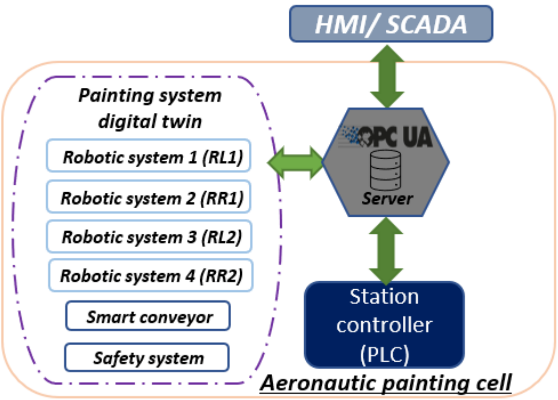

Fig. 8. Single scenario toolchain. Components of the painting system are surrounded by the purple dotted line.

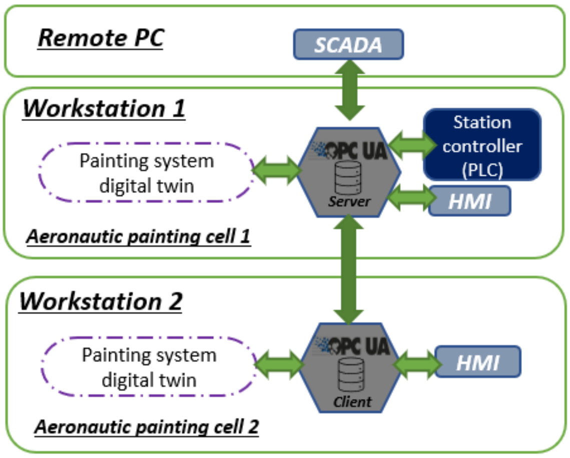

Fig. 9. Extended toolchain implemented for distributed simulation in multiple PC platforms.

3.2. Complexity Analysis

The proposed distributed approach, based on the one shown in [7], considers the modelled objects as sets of attributes and a total of 11 measures that needs to be determined for each system or toolchain. The proposed model takes several inputs into account (3D model, Programming parameters, time …). The proposed measured parameters are listed next.

- M1: the number of the modelled objects.
- M2: the number of the connections among the modelled objects.
- M3: number of attributes of all the objects (Mass, volume, gravity).
- M4: Number of manually changed attributes (more complex), like detectability.
- M5: Number of attributes that are not inherited from parent object.

- M6: Summarizing the cyclomatic complexity of all the program blocks.
- M7: Total number of lines in a program.
- M8: Required time duration
- M9: Number of running software in the toolchain
- M10: Real time estimation (create a weight parameter that effect the complexity)
    - 0: if system is not real-time
    - 1: if system soft real-time
    - 2: if system is hard real-time
- M11: Number of cycles needed to complete the simulation.

We applied this scheme to each software used in the toolchain to have a better vision of the complexity of the desired system, see Table 2.

Table 2. Complexity analysis of the extended toolchain.

ET	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
DELMIA	112	203	100	88	188	19	175	450
CB	0	0	0	0	0	0	33	450
CoppeliaS	135	81	236	127	363	24	1569	470
Python	0	0	0	0	0	2	192	470
SIMIT SP	98	49	0	0	0	2	0	470
PLCSim Adv	0	0	0	0	0	0	0	470
TIA PORTAL	55	13	0	0	0	2	0	470
WINCC	52	50	0	0	0	0	0	470
TOTAL	400	346	336	215	551	47	1969	470

It is noticeable that Delmia toolchain is faster than CoppeliaSim toolchain since the physics engines and the digital models programming are different, even if they share the same solution architecture. A comparison between each individual toolchain complexity is shown in Fig. 10, while Fig. 11 displays the complexity of each software used for the extended toolchain. It is worth noting that PLCSIM Advanced and SIMIT SP are the only software considered as soft real time. The simulation only requires one cycle to complete the painting tasks, taking 470 seconds in M11.

4. Simulation Analysis and Results

To test the efficiency of the safety system, an operator dummy was added to both scenarios: it was programmed to behave as a human worker to verify the reduction of smart components' speed. The results were acceptable since the signals are correctly sent along all the modules: Delmia to Control Build (OPC-UA Client) directly connected to the OPC-UA server, while on the other hand, a second toolchain going from CoppeliaSim scenario to the python OPC-UA client, then, connecting to OPC-UA server, which shares the information to the PLC OPC-UA client, containing a simulated HMI for control and monitoring.

Data can also be sent in the other way, i.e., the user can send an emergency stop signal for one of the robots from the virtual HMI and see that the system in the 3d scenario fully stops. To improve simulation performance, the software toolchain was divided into three different workstations connected to a local network: one for simulation layer and OPC-UA server, another one for execution layer and HMI, and the last one for SCADA client configurator and operator interface.

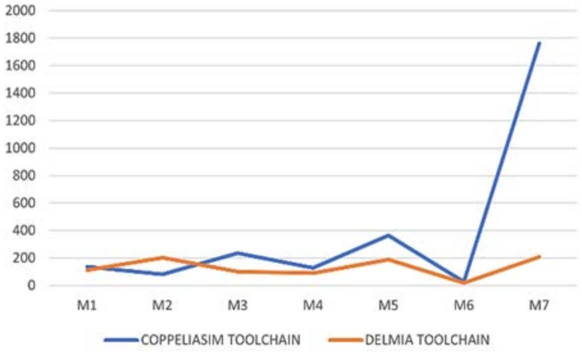

Fig. 10. Complexity comparison of individual toolchains.

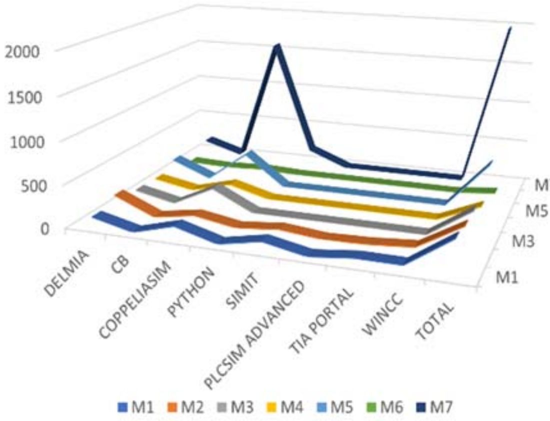

Fig. 11. Complexity analysis of the extended toolchain using the indicators from M1 to M7.

5. Conclusion

A distributed simulation approach can be applied whenever a system or process is highly complex and high computer resource consumer. For VC applications, the distribution of scenarios on multiple computers allows the developer to achieve high performance during validation and verification of simulations, and even physically test communication protocols. Some of the advantages obtained from this approach are listed below:

1. Resource utilization. By distributing the simulation load among multiple entities, each entity only needs to focus on simulating a smaller portion of the overall system, allowing for more efficient use of available computing resources.
2. Realism. By modeling the interactions between different parts of a system in parallel.
3. Scalability and flexibility: dynamic addition or removal of simulation entities, making it possible to adapt to changing simulation requirements in real-time.
4. Improved accuracy: By breaking down the simulation into smaller parts that can be run in parallel, the overall simulation can be run faster and with greater accuracy, since the errors in each part are independent.

It is worth noting that the use of several computers increases the cost of VC solutions due to the use of extra hardware.

Another method to improve the distributed simulation approach is the use of hard real-time SW and HW, creating a fully realistic representation of the scenarios and validating HiL simulations.

Acknowledgements

This work was conducted and funded with the support of the Research Tax Credit (CIR in French). A special thanks to all the R&I Virtual Commissioning non-permanent collaborators that help us to develop the necessary tools, libraries, and models, allowing us to turn ideas into projects.

References

[1]. A. Abassi, J. Lugo-Calles, R. Balderas-Hill, and N. Lassabe, Distributed Virtual Commissioning implementation of an aeronautic shopfloor. in *Proceedings of the 3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0 / 5.0 (ARCI' 2023)*, 2023, pp. 145-149.

[2]. R. Balderas Hill, J. Lugo Calles, J. Tsague, T. Master and N. Lassabe, Improving interoperability of Virtual Commissioning toolchains by using OPC-UA-based technologies, in *Proceedings of the 26th IEEE International Conference on Emerging Technologies and Factory Automation*, September 2021, pp. 1-7.

[3]. R. Balderas Hill, J. Lugo Calles, J. Tsague, T. Master and N. Lassabe, Virtual Commissioning Implementation of Side Panel Assembly Process for Automotive Shop Floor, *Sensors & Transducers Journal*, Vol. 257, Issue 3, May 2022, pp. 12-19.

[4]. J. Lugo Calles, A. Abassi, R. Balderas Hill, and N. Lassabe, Zero to hero: a generalized virtual commissioning implementation method, in *Proceedings of the Meeting for the Dissemination of Research in the Study of Complex Systems and their Applications (EDIESCA2022)*, September 2022, pp. 1-13.

[5]. J. Lugo Calles, A. Abassi, R. Balderas Hill, and N. Lassabe, Virtual Commissioning Implementation of Industrial Aeronautic Assembly Process, in *Proceedings of the Meeting for the Dissemination of Research in the Study of Complex Systems and their Applications (EDIESCA2022)*, September 2022, pp. 1-16.

[6]. Albo, Anton & Falkman, Petter, A standardization approach to Virtual Commissioning strategies in complex production environments, *Procedia Manufacturing*, 51, 2020, pp. 1251-1258.

[7]. Gergely Popovics, László Monostori, An approach to determine simulation model complexity, *Procedia CIRP*, 52, December 2016, pp. 257-261.

[8]. Lutz Winkler, Heinz Wörn, Symbricator3D – A Distributed Simulation Environment for Modular Robots, in *Proceedings of the Second International Conference on Intelligent Robotics and Applications (ICIRA 2009)*, Singapore, December 16-18, 2009, pp. 1266–1277.

[9]. Chen Peihua, Cao Qixin, Yang Bixin, Leng Chuntao, An RTM based Distributed Simulation System for Guide Robot, in *Proceedings of the 4th International Conference on Digital Manufacturing and Automation (ICDMA)*, 2013, pp. 185 – 189.

[10]. Paul Breugnot, Bénédicte Herrmann, Christophe Lang, Laurent Philippe, Data Synchronization in Distributed Simulation of Multi-Agent Systems, Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation, in *Proceedings of the 20th International Conference PAAMS 2022*, L'Aquila, Italy, July 13–15, 2022, pp. 50–62.

[11]. A. Abassi, J. Lugo-Calles, R. Balderas-Hill, and N. Lassabe, Virtual Commissioning Implementation of Industrial Painting Process, in *Proceedings of the 2nd International Conference on Computers and Automation (CompAuto' 2022)*, France, August 18-20, 2022, pp. 65-68.

[12]. Vitolo, F., Rega, A., Di Marino, C., Pasquariello, A., Zanella, A., Patalano, S., Mobile Robots and Cobots Integration: A Preliminary Design of a Mechatronic Interface by Using MBSE Approach, *Appl. Sci.*, 2022, 12, 419.