

VALIDATING A RECONFIGURABLE ASSEMBLY SYSTEM UTILIZING VIRTUAL COMMISSIONING

H.J. VERMAAK & J.A. NIEMANN

CENTRAL UNIVERSITY OF TECHNOLOGY, FREE STATE

Abstract

South African manufacturing companies today need to be more sophisticated technologically to compete for global markets. The latest trend in automation and manufacturing emerges in the form of reconfigurable systems. The aim of this paper is to show the development of a reconfigurable assembly system and using virtual commissioning to plan, validate and optimize it. To achieve this “DELMIA” software was used to create a virtual simulation environment to verify an assembly cell from such a system as a case study. Simulations were conducted to verify software functions, device movements and operations, and the control software of the system. As a result, it was found that virtual commissioning is an excellent tool for predicting how the system will function, verifying system code early, and rectifying design flaws. This will enable manufacturing companies to be more competitive, ensure increased productivity, save time and ensure them an advantage over their competition.

Keywords: DELMIA, Digital Manufacturing, Reconfigurable Assembly Systems, Virtual Commissioning

1. INTRODUCTION

By definition, virtual commissioning is the simulation of a virtual system within a virtual environment without needing to develop a physical system beforehand. By utilizing virtual commissioning, design flaws can be rectified early in the design stage, space reservation can be allocated for the machinery used in the system and controller software verified well in advance, before building the physical system (Krause, 2007). Virtual commissioning also allows for easy reconfiguration of an existing system, where process, software or hardware changes can be made in the digital model of the system, then analysed to see how these changes influence the system, and then based on the analysis results the physical system can be modified, preventing costly downtime of the physical system. Furthermore, in industry it enables manufacturers to streamline an assembly line, where planning is done more efficiently and through-put can be predicted due to the visualising of the assembly line (Papakostas et al., 2011). In short, virtual commissioning is established when a virtual factory is controlled by a physical programmable logic controller (PLC) to emulate the behaviour of the physical system (Salamon and Heidari, 2012, Carlsson et al., 2012, Liu et al., 2012, Hollander and Sappei, 2011).

2. PRELIMINARY STEPS TO OBTAIN VIRTUAL COMMISSIONING

To obtain virtual commissioning, some preparations must be done ahead of time. These preparations include that the user must acquire prior knowledge of the simulation software environment under discussion, the devices to be validated along with knowledge of their behaviour and kinematics must be known, and a server (Carlsson et al., 2012) to communicate between the control logic and these devices must be installed. Afterwards, the geometry used to represent these devices must be designed, assembled into smart-devices and then connected to the control logic to validate the system behaviour and control software via an execution environment (Niemann, 2013).

Subsequently follows how geometry is obtained, how it fit together, how behaviour is assigned to it, and finally how it is used to obtain virtual commissioning and validating a system.

2.1 Product Hierarchy

Any geometry which consists of multiple parts, or other products, is known as a product. A product is the root element of a hierarchy and contains multiple sub-elements to represent the branches of the hierarchy tree. Fig. 1 shows clearly that a collection of parts are grouped together to form an assembly. Here, constraints and tasks can be allocated to the assembly, then be connected to internal device logic to form a smart-device. In a similar way, multiple smart-devices can be used together to form work cells and ultimately an entire system (Park et al., 2006, Park, 2005).

2.2 Parts, Assemblies, Mechanisms and Tasks

As seen in the previous section, smart-devices consist of geometry and internal logic (IL). The geometry of a smart-device consists of parts, assemblies and mechanisms (joints). Parts are the most fundamental elements of any geometry, and there are two methods of attaining them, downloading the parts from a vendors' website and alternatively, parts can be

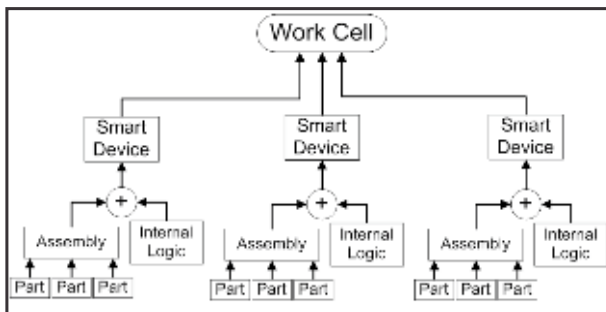


Fig. 1: Hierarchy of a work cell

designed and created by using the suitable computer-aided design (CAD) software (Salamon and Heidari, 2012). This hierarchy of a work cell is shown in Fig. 1. In contrast, an assembly is a collection of parts, linked together by means of specified constraints (see Fig. 2). These assemblies represent a mechanical assortment, which contain at least one fixed part and various moving parts. An example is a piston that is driven by a crank. Moreover, the steps taken to construct these assemblies can better be explained by referring to Fig. 2. Firstly, the parts are imported into the environment (shown at number 1 in Fig. 2). Next the parts must be assigned constraints, which include alignment, orientation of parts and surface contact constraints (shown at number 2 and 3 in Fig. 2). After constraint allocation, the geometry can be updated, causing the parts to rearrange into their intended positions (shown at number 4 in Fig. 2). In addition to constraints, these assemblies can be allocated kinematical commands, by specifying the physical limits, direction of movement, speed, and acceleration of the assembly. This is known as creating a joint or mechanism.

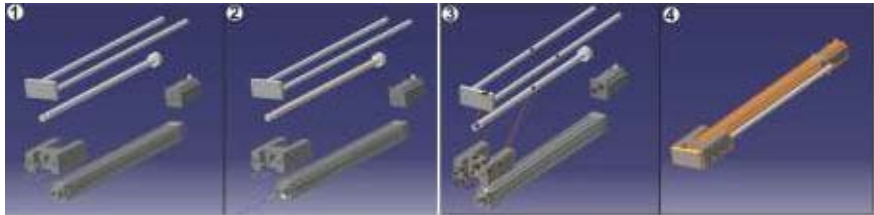


Fig. 2: Assembly of parts

Additionally, these mechanisms can be given behaviour, by allocating tasks and operations to it. To differentiate, an operation depicts the movement of a mechanism or several mechanisms when executed, whereas a task executes a series of consecutive operations and functions (like closing a gripper). To allocate a task, the sequence of possible activities are identified and taught to the mechanism. This entails that the mechanism is jogged or moved into place, where after the set of coordinates are recorded in an operation table which will perform the movements sequentially. This will give mechanisms the needed behaviour to be used as smart-devices.

In a similar fashion to joint creation, different smart-devices can be assembled together to form a larger, more complex assembly or system (Krause, 2007). Fig. 3 shows how separate smart-devices are first imported (shown at number 1), then aligned into position (shown at number 2) and finally attached. These attachments can be seen in the figure at number 3. The smart-devices are attached in a parent-child manner (shown at number 3), which causes smart-devices to move respective to others. The example in the figure shows that the Y-axis (child) must move as if it is fastened to the slider of the X-axis (parent).

Likewise, the cylinder (child) moves with the Y-axis (parent) and the gripper tool (child) moves with the cylinder (parent) and so forth (Salamon and Heidari, 2012, Niemann, 2013).

2.3 Control Logic

In order to control device geometry so that it imitates the behaviour of its physical counterpart or cause several devices to cooperate with each other within a digital factory, control logic must be developed. Fig. 4 shows that control elements or logic can be separated into two types namely device logic and control logic, where control logic divides into internal and external control logic.

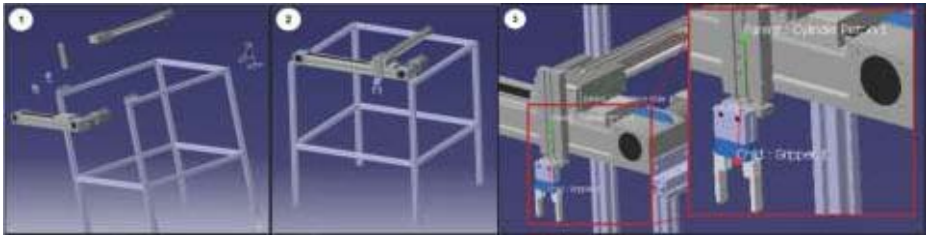


Fig. 3: A completed smart-device

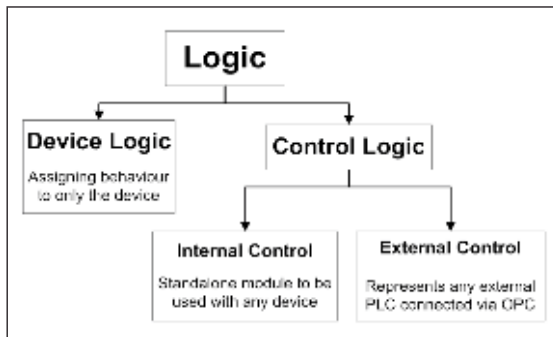


Fig. 4: Device and control logic

Firstly, device logic also known as internal logic, assigns unique behaviour to the devices or assemblies created as in the previous section to make it smart-devices. The IL uses the inputs and outputs of each device to control the actions it must perform, which gives each device a distinct behaviour (shown in Fig. 5).

On the contrary, control logic is a standalone supervisory control instance, which can be used to control multiple smart-devices within a digital factory. To clarify, internal control can be seen as a virtual PLC (running inside the environment) connected to devices to emulate the behaviour of a real PLC.

With external control on the other hand, the devices are connected to a real system PLC via an OLE for Process Control (OPC) server. The latter is used to validate system PLC code.

3. IMPLEMENTATION AND VALIDATION

The steps to obtain virtual commissioning can better be explained by referring to Fig. 6. Firstly all the parts for the system must be obtained by either designing or downloading them. Afterwards, the parts must be assembled into assemblies

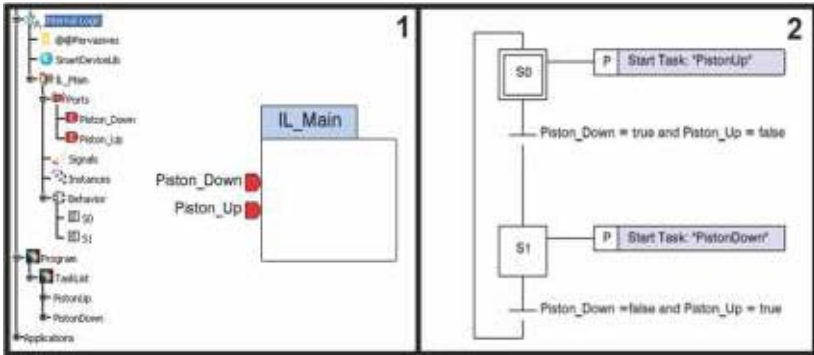


Fig. 5: Control block with internal logic

to build up the devices of the system. The devices can then be given behaviour by defining constraints and allowable movements. At this stage the devices must be verified for accurate mechanical operation by performing joint simulations. Next, the internal logic for each device to be validated are developed and then applied to it. This represents the intelligence of each device (internal code of devices). After this step is complete, the geometry can be referred to as smart-devices. Now all the smart-devices can be imported into the environment to complete the entire system to be validated. At this stage the environment is set up for virtual commissioning (Niemann, 2013, Cachapa et al., 2007).

To demonstrate how virtual commissioning can aid the validation of a system and its PLC code, the validation procedure is divided into several tests. An optional simulation can be done to verify the operation of the virtual system in the form of a process plan simulation. Operational tasks are assigned to each smart-device to emulate the overall process flow of the system (Bley and Franke, 2004). This method does not validate any control or devices logic, only the functional operation of the virtual system and is not handled in this paper. For test 1 a simulation is set up to validate the virtual version of the system along with device logic. This is done to predict how the virtual system will react to control code. This test is only done to validate the behaviour of the virtual system (device operations and internal logic).

In test 2 the virtual system is connected to the physical system PLC via an OPC server. Here simulations are repeatedly done until the operation of the virtual system is satisfactory and thus the physical PLC code

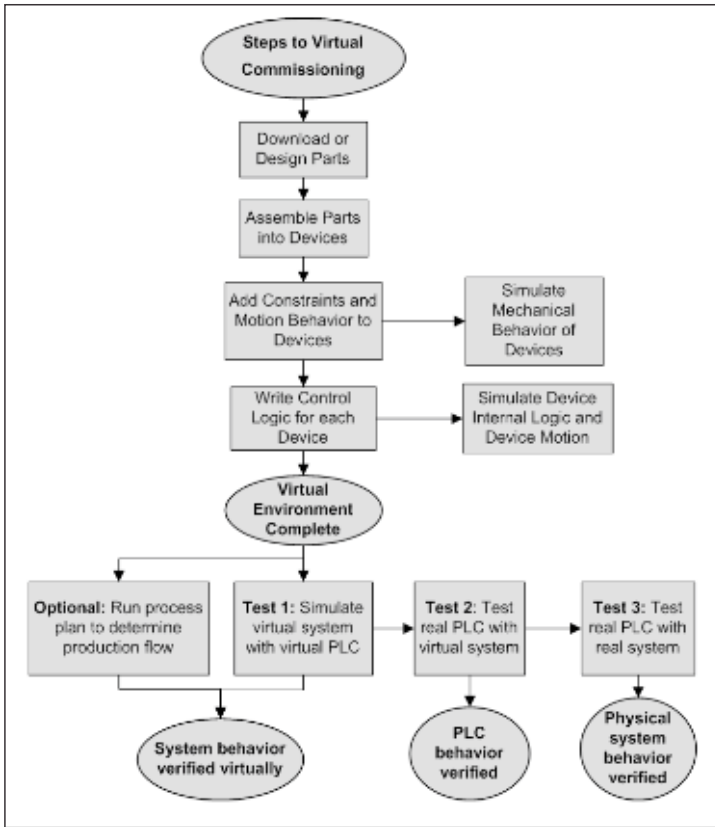


Fig. 6: Steps to virtual commissioning

is verified successfully. This is known as virtual commissioning. For the last test, the physical system is connected to the verified PLC and operated. Now the operation of the system can be compared to the predictions made through virtual commissioning. This will evaluate if virtual commissioning successfully predicted and validated the operation of the physical system (Kong et al., 2012, Lee and Park, 2014, Hoffmann et al., 2010, Liu et al., 2012).

3.1 Setup Overview

To achieve this, the setup is as shown in Fig. 7. The virtual version of a device is built and programmed with its behaviour, on a computer with the simulation software environment installed on it.

In addition, the simulation environment is connected to an OPC server, which is installed and also runs on the same computer. Furthermore, the server connects to the system PLC via an Ethernet connection, which enables the server to access and communicate with the PLC. To complete the setup, the physical version of the device is built and also connected directly to the main PLC, which enables it to control it.

3.2 Execution Environment and Simulation

In order to set up the simulation, an execution environment must be created. Firstly, the virtual device must be designed, assembled and imported into the environment. Next, the virtual equipment is programmed with the behaviour of the physical equipment. This will transform the virtual device into a smart-device. In addition, joint simulations must repeatedly be done until the operations and movements of the virtual device can imitate its real counterpart based on input/output (IO) signals from a PLC. After the virtual device is perfected, the PLC code to control the system is developed on the actual system PLC. When the PLC code is completed, it is necessary to establish a connection with the OPC server. To achieve this, the IOs of the PLC must be mapped to the virtual device. This can be seen in Fig. 8. Here the IO signals of the PLC are mapped to the virtual equipment exactly the way it should be physically wired to the real equipment.

Fig. 7: System validation overview

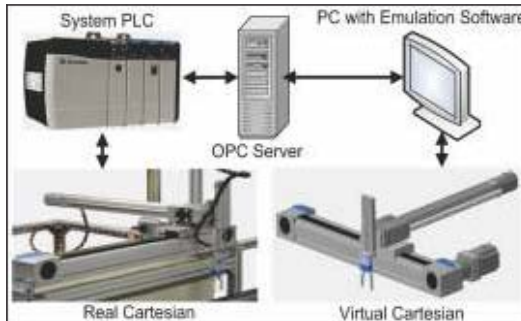
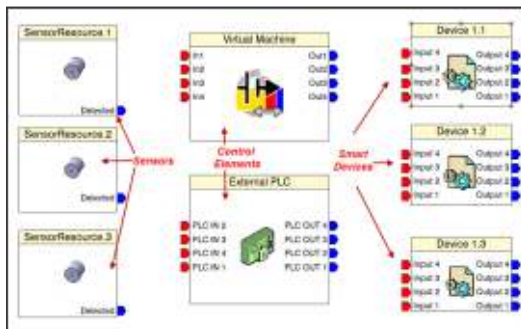


Fig. 8: Mapping devices to PLC



When all the connections are made, the execution environment can be compiled and checked for errors and signal strength. If no errors are present and the signal quality of the OPC server is good, the environment can be simulated to validate the PLC code and predict the movements of the system based on the instructions from the PLC. At this stage the PLC code can be modified until the virtual system delivers the desired intended operation of the system. If the virtual environment functions flawlessly with the PLC code, the PLC can be connected directly to the physical equipment. At this stage the validated PLC code runs and controls the physical system in real-time. Now a comparison can be made between the prediction made through virtual commissioning and the actual running system (Niemann, 2013).

4. DISCUSSION

The simulations which were performed showed the desired outcome. The simulation showed that the virtual device executed the operations and device movements in accordance with the PLC program. By obtaining this result, the PLC code is successfully validated. In addition, the simulations also provided a method to easily develop, improve, and troubleshoot the system PLC code.

With the physical commissioning, it was found that the predictions made through virtual commissioning, were successfully obtained. The only difference between the virtual simulation and physical commissioning was the speed at which it executed. This however can be rectified by changing the execution speed of the simulation and make the simulation more realistic; this will result in more accurate predictions and validation of the system.

5. CONCLUSION

The results obtained proved that virtual commissioning can be utilized to expedite the planning, verifying and optimizing of a system without the risk of damaging the real system equipment. It was possible to confirm the movements and operations of a device as well as validating the system PLC code through virtual commissioning in real-time. This shows that by using DELMIA to establish virtual commissioning, proper initial planning can be performed, apparent design faults can be resolved early in design stages, analysis can be done to validate changes to a system, and determine if it is actually profitable to build a system—thus build it right the first time or not at all.

To draw a conclusion, DELMIA provides a great tool for overall system verification and will ensure that manufacturing companies can validate their systems earlier, ramp-up production quicker and compete for global markets.

However, future considerations regarding virtual commissioning may include suggestive or predictive action from the virtual environment where the environment can propose solutions overlooked by a system analyst.

Furthermore, features like automatically generating system code based on virtually validated systems will also be a welcome addition to virtual commissioning.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Central University of Technology Free State and AMTS under the project 07-11-P funding agreement; for the opportunity to conduct research on this topic.

7. REFERENCES

3DS - Dassault Systemes, Available Online: <http://www.3ds.com/products/delmia>, last accessed in August 2013. [Online].

CATIA Online Documentation, Available Online: http://catiadoc.free.fr/online/CATIAfr_C2/prtugCATIAfrs.htm, last accessed in August 2013. [Online].

Kepware Technologies - OPC Servers / Communications for Automation, Available Online: <http://www.kepware.com>, last accessed in August 2013. [Online].

BLEY, H. & FRANKE, C. 2004. Integration of product design and assembly planning in the digital factory. CIRP Annals-Manufacturing Technology, 53, 25-30.

CACHAPA, D., COLOMBO, A., FEIKE, M. & BEPPERLING, A. An approach for integrating real and virtual production automation devices applying the service-oriented architecture paradigm. Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on, 2007. IEEE, 309-314.

CARLSSON, H., SVENSSON, B., DANIELSSON, F. & LENNARTSON, B. 2012. Methods for reliable simulation-based PLC code verification. Industrial Informatics, IEEE Transactions on, 8, 267-278.

HOFFMANN, P., SCHUMANN, R., MAKSOUD, T. M. & PREMIER, G. C. Virtual Commissioning Of Manufacturing Systems A Review And New Approaches For Simplification. ECMS, 2010. 175-181.

HOLLANDER, A. & SAPPEI, S. 2011. Virtual preparation of Tetra Pak filling machine. Master Thesis, Department of Signals and Systems, Chalmers University of Technology, Sweden.

KONG, X., AHMAD, B., HARRISON, R., PARK, Y. & LEE, L. J. Direct deployment of component-based automation systems. Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on, 2012. IEEE, 1-4.

KRAUSE, H. Virtual commissioning of a large LNG plant with the DCS 800XA by ABB. 6th EUROSIM Congress on Modelling and Simulation, Ljubljana, Slovénie, 2007.

LEE, C. G. & PARK, S. C. 2014. Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, 1, 213-222.

LIU, Z., DIEDRICH, C. & SUCHOLD, N. 2012. *Virtual Commissioning of Automated Systems*, INTECH Open Access Publisher.

NIEMANN, J. 2013. *Development of a Reconfigurable Assembly System with Enhanced Control Capabilities and Virtual Commissioning*. Master Dissertation, Faculty of Engineering and Information Technology, Central University of Technology, Free State.

PAPAKOSTAS, N., MICHALOS, G., MAKRIS, S., ZOUZIAS, D. & CHRYSOLOURIS, G. 2011. Industrial applications with cooperating robots for the flexible assembly. *International Journal of Computer Integrated Manufacturing*, 24, 650-660.

PARK, C. M., BAJIMAYA, S. M., PARK, S. C., WANG, G. N., KWAK, J. G., HAN, K. H. & CHANG, M. Development of virtual simulator for visual validation of PLC program. *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, 2006*. IEEE, 32-32.

PARK, S. C. 2005. A methodology for creating a virtual model for a flexible manufacturing system. *Computers in Industry*, 56, 734-746.

SALAMON, O. & HEIDARI, A. 2012. *Virtual commissioning of an existing manufacturing cell at Volvo Car Corporation using DELMIA V6*. Master Thesis, Department of Signals and Systems, Chalmers University of Technology, Sweden.