

Virtual Reality Machines to Improve Training in Automation

Manuel Eduardo Macías García

Tecnológico de Monterrey, Campus Monterrey, Monterrey, N.L., México, mmacias@itesm.mx

Alfredo Rafael Izaguirre Alegría

Tecnológico de Monterrey, Campus Monterrey, Monterrey, N.L., México, a00804821@itesm.mx

ABSTRACT

Current market requirements in industrial sector have motivated the development and adoption of digital manufacturing software tools for control systems design, training, and process optimization. This practice, known as Virtual Commissioning, emulates the real process behavior in a computer software environment. This technology represents an opportunity for education where the virtual emulation of real processes can be used to equip Control and Automation laboratories where students can test, validate, and debug their control and automation strategies, contributing to student formation and solving the need of having costly, real industrial machinery to reinforce the understanding of classroom theory with practice. This is an excellent option for universities without enough resources, mainly in developing countries, where laboratories are commonly equipped with improvised homemade systems that don't represent what students will face in an industrial environment. The development of a general procedure of 6 stages for the creation of an automation simulation virtual final application called Virtual Reality Machine (VRM) is presented. VRM capabilities, scope, usage, and contribution to student's formation are presented. Finally an equipping solution for automation and control laboratories is proposed.

Keywords: Virtual Machines, Training, Automation, Virtual Commissioning, Virtual Labs

1. INTRODUCTION

The consumption world market constantly is increasing and changing its requirements. Day by day more and better quality products are asked for customers what shrink the market product life cycle, encourage a more extensive product variety and reduce product launch times. Moreover this is happening while market price erode, global sourcing increases and quality product must to keep high (ARC Advisory Group, 2006). New requirements represent challenges that manufacturers have to face in a high competition environment what forces companies to implement technologies, processes, and practices that enhance their competitiveness and differentiate from the others competitors. Product improvement, constant product offer changing, process standardization and optimization, price and cost reduction, quality, etc. are some of the practice that companies implement daily oriented to accomplish new market requirements. However some of these practices appear to be opposed, by one hand having a fresh product offering requires constantly production lines changing what opposes to cost reduction and process standardization. Then to remain successfully in market and in some cases even to survive, companies must be able to keep a constant innovation. This targeted to look for necessary practices and tools that support them to face actual challenges and assurance them that changes in any production aspect, will impact in the planned way all related company sectors.

Market globalization has forced that companies increase production quotes motivating that these open plants in other countries, increase production lines, change or totally replace their process and the way of fabricate goods, etc. This growing has brought a bigger organization structure inside companies, more process and equipment to control and manage. In addition at the same time, practices like lean manufacturing, six sigma, QFD, ISO quality

certifications and other production quality activities are being implemented. This is translated in more product's specifications, processes and procedures information to handle. For solving this, many software tools focused to different sectors and with different scope have arisen as support for manufacturing companies; CAD/CAM, CAE, CAPP and PPC are some of these tools. These tools main objective is to enable companies to achieve the business imperative of timely and cost effective product launches. These software tools oriented to improve and enhance manufacturing's resources and production are known as Digital Manufacturing tools (DM) as merge the virtual and physical manufacturing world.

1.1 DIGITAL MANUFACTURING (DM)

DM is an integrated suite of software solutions that supports manufacturing process, tool design, and visualization through 3D virtual simulation tools. The factory environment and the production process can be modeled including buildings, production lines, transportation, workflow, and other facilities that represent the complete physical production environment¹. On these environments, manufacturing engineers can validate and optimize the processes, designing, synchronizing, and validating production lines, robotic work cells, machine centers, production equipment, control systems functionality and requirements completely prior to the purchase, installation or commissioning of physical equipment. In essence, DM facilitates the complete view of product and process design as integral components of the overall product life cycle. This virtual validation and commissioning have turned more important lately as the looking for major production and trusty processes have caused that totally automated complex manufacturing systems be used more frequently in industry. Because of this there is a section of DM that allows manufacturing engineers to merge virtual models of production equipment with automation and controls what enables the complete validation of controls logic, automation strategies and HMI functionality in a process called Automation Simulation (Caie, 2008). This extended level of manufacturing process design let executing "perfect" launches and production changes by validating totally all aspect related with the process from the tool and machine design to the final automation strategy.

1.2 AUTOMATION SIMULATION

Automation simulation involves the entire process of modeling, animation, evaluation, optimization, and validation of controls systems for automation equipment and systems in a virtual environment. Which represent manufacturing work cell with 2D image or 3D solids elaborated in CAD that may represent from single process with primitive graphics to elaborated production systems, including multiple robots, complex tooling and fixtures, clamp automation and PLCs. Automation Simulation intends to mimic and simulate the real process to be automated by animation sequences, with the objective of commissioning and debugging total or partial control logic changes of the manufacturing work cell (Caie, 2008) . This can be done even weeks or months before the real machinery be present in shop floor, as control logic strategy is simulated in virtual cell where interaction and control sequences of tooling, robots, clamps, safety devices, electrical, hydraulics and pneumatics, etc. can be tested. Inside automation simulation this practice is known as virtual commissioning and helps to assurance that any change in automation system will cause the desired impact in production, providing to manufacturing and controls engineers an opportunity to ensure controls design before production starts.

1.3 VIRTUAL COMMISSIONING

Virtual commissioning principal objective is assurance optimizes and validates control and automation implementation and changes in virtual automated production systems prior to real commissioning. This is on state-of-the-art digital manufacturing simulation technology; such depends on advanced simulation methods that trusty represent the merging of 3D virtual simulation environment, to accomplish the level of automation and synchronization required, with the physical automation world of control logic and control platforms that perform the production processes (Dassault Systems, 2007). Virtual commissioning is carried out on virtual prototypes of production systems and equipment which are based on direct real model capabilities and appearance. Originally was intended for allowing the debugging of the control code on an actual Programmable Logic Controller (PLC) (Dassault Systems, 2007). However its scope goes further as allows the user efficiently and cost effectively optimizes and validates any implementation or change in manufacturing processes control strategies, testing different control scenarios, accelerate learning curve and enables control engineers with the ability to reduce

costly errors occurrences, and mitigate risks in a virtual environment well before using real equipment to accomplish commissioning. This Virtual Commissioning features have caused that also is used in industry with training purposes. This way, automation simulation supported by virtual commissioning is used for industry and for education. In the first one it has two orientations; production systems optimization and validation, and control engineers and laborer training. In the second one it is used in control and automation courses where final applications are used for supporting engineering student's formation with practice.

1.6 AUTOMATION SIMULATION IN EDUCATION

Automation simulation software tools are used with two orientations in education. The first one is their usage principally in universities focusing on control, automation, mechanical, manufacturing and electrical programs. In universities, automation simulation tools are taught principally in advanced engineering courses where students learn about the automation tool itself, its features, limitations, scope, etc. Students work with the development environment and not with final emulations. This is because the teaching is oriented to the automation simulation tool as it is used in the industry and it also requires engineers working with these tools. The second automation simulation orientation consists in using a final emulation application to teach engineering students and to support the automation and control theory taught in classrooms and is later practiced in laboratories. On these virtual environments, a student's control and automation strategies can be tested through the emulation of real production systems, without the risk of damaging costly equipment. This is done taking advantage of the automation simulation objective of validating, optimizing and testing control and automation strategies in virtual environments. Then, these are deployed in the real production systems equipment. It is also possible to carry out all the automation commissioning process, and never deploy the automation control strategies in a real model, just as the training is done with engineers in the industry. This way of using automation simulation tools offers the capability that with the knowledge of the electrical, mechanical, and dynamic processes practically any production system, machine, work cell, production line, etc. can be emulated by virtual commissioning.

Having this variety of emulations enables students to deploy different control strategies for each one, relating them with different sensors, actuators and machinery functionalities. From this point of view, if automation simulation is well used in education, it opens the opportunity to solve the problems that many universities face in regards to equipping control and automation laboratories. The reason to have a process simulation for the students to practice the knowledge acquired in the classroom is because a real work cell used in the industry is very expensive. Universities can seldom afford to get one. Therefore, the possibility of having a variety of virtual emulations processes with which students increase, reaffirm and practice automation and control skills is an opportunity for higher education institutions to complement and support their automation and control courses, and to increase their students' knowledge and abilities. However in market there isn't a big offer of final virtual application. There are only a reduce number of companies like Festo by Ciro's Mechatronics, SIEMENS by SIMIT SCE and EasyPLC that offer 3D final application for carry out virtual commissioning and learning automation. Prologix an independent tool offers 2D final virtual applications, others like Delmia Automation by Dassault Systemes, Tecnomatix by SIEMENS, RsTestStand by Rockweel, Unity Pro by Schneider are oriented to give the software tool for building virtual applications and not offer final virtual environments where students can testing and validate their programs.

2. VIRTUAL REALITY MACHINE (VRM)

There isn't a single term for calling final automation simulation. In market every vendor gives a different name, trying to cover the particularities of everyone. The features of the virtual final application that is going to be presented, explained and developed turn it different from those mentioned in table 1. For this reason the concept of Virtual Reality Machine (VRM) is introduced and defined next.

A VRM is a computational application that models and emulates appearance and behavior of a real machine, process or production system. It is based on a set of 3D solids created in CAD software. It represents real machine elements drawn from reality and then converted. Considered the dynamics of the process, the LabVIEW Virtual Instruments (VI) called Virtual Machine Elements (VME) are placed spatially in the correct (X, Y, Z) positions

through an assembled process for mimicking visually and representing virtually real machines in a production system. These are animated to create the Virtual Machine Mechanism (VMM) emulating the real machine dynamics and functionality, as close as possible to the real one. The VMEs also represent virtual sensors and actuators as in real machines. Their physical addresses are defined and established in a pin out table by a connecting process enabling them to interact with external control devices or applications by sending and receiving control signals through communication libraries, from/to automation platforms either real PLC or SoftPLC. What makes that dynamic and functionality of the VMM depends only on the control logic sequences programmed by the user in these automation platforms.

2.1 GENERAL PROCEDURE FOR VRM DEVELOPMENT

A VRM is a virtual final application. It is intended to emulate a real process; thus whenever a process is emulated, a VRM has to be created. Despite the fact that every VRM has a specific final functionality, when these are created, they share certain general developing aspects that repeat every time a new VRM is developed. These aspects can be summarized, organized and separated in a set of steps according to their developing objective. For each one of these steps, a developing method is followed. The conjunction of all steps integrate a general VRM developing procedure that can be clearly described in 6 steps, as follows: 3D CAD creation, Conversion process, Assembly process, Animation process and VRM Validation process. Each one of the VRM developing stages and its procedures are described next. Figure 1.1 shows this stages and their order within the general procedure.

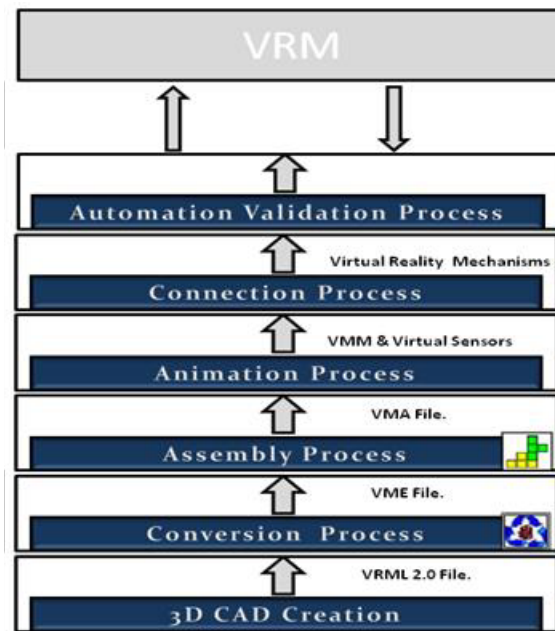


Figure 1: General Procedure for VRM Creation

3. 3D CAD CREATION

The first step for creating the VRM is the Virtual Machine Elements (VME) modeling in CAD software, the only requirement from the CAD software to use is that it has the capability for saving the solid created in VRML 2.0 (97) (Carey et al., 1997) format, requirement that must be accomplished for future conversion in LabVIEW. Two consideration must to be taking in care when modeling in CAD software that later are reflected in the VME due to VRML structure features. The first one is “solid number”. When the VRML file is converted in a VME this turns in a LabVIEW Virtual Instrument (VI) turning the solid information in LabVIEW code part. The conversion makes more efficient the final application, extends manipulation capabilities, and enhances the features that can be represented, accessed or gotten from it before or during simulation running. This is deeper detailed in Solid

Conversion Section. The second consideration is “solid functionality” aspect that is important for defining if a part or an assembly is going to be created. As is necessary consider the dynamic that will have the 3D CAD solid in the final VRM. Similar to real machine in a VRM there are two classes of VMEs, dynamic VMEs, that move as result of a PLC input signal and static VMEs that are part of the VRM structure, this difference in VMEs has to be considered when the decision of what solid include in an specific VRML that is going to be converted is taken.

4. CONVERSION PROCESS

Once that the VRML file is created it is processed in an application that has been developed called VRML-VME Converter. The VRML file structure contains in ASCII code the complete information that defines the solid, this is separated in object nodes, where each one of this contains information about the triangle’s coordinates vertex, normal, color values and index of each solid part that forms the solid (Carey et al., 1997). The application recognizes this structure and processes the VRML files, gets and organizes the important data required for render the new file in LabVIEW. Then the VRML file is processed only once by LabVIEW. As result a VME (Virtual Machine Element) is created. This is a LabVIEW Virtual Instrument (VI), which represents exactly the solid models as the VRML file, since nothing of information is lost during the conversion. Once that the process finishes the new VME in saved in a library. Then the VME now is a LabVIEW native file and has all the information necessary for representing and rendered the 3D solid without need of VRML or any CAD software or file. Having the 3D solid as VME represents some advantages as: faster execution time, deeper control of the 3D solid, integration of VME code to other LabVIEW applications, makes available the usage of some special properties because all solid data are available during execution. There are two classes of VME for Virtual Mechanisms and VME for Environment.

VME for Virtual Mechanism: A Virtual Machine Mechanism (VMM) is integrated by two or more VMEs which are related by at least one dynamic action one over another and at least one of them is affected by a change of signal received from the external controller. It can be affected with translational and rotational movement or color and shape change animation.

VME for Environment: The objective of this VME type is to serve as VRM decoration exclusively this represents for example floor, walls or any other solid in the scene that has not connection with the real controller. This can have movement or be static during VRM functionality.

5. ASSEMBLY PROCESS

A set of virtual machine mechanisms (VMM) is used for creating a VRM. These VMMs are integrated by a set of animated VMEs which first have to be placed spatially in their corresponding position inside the VRM in a previous VRM assembly process, for this, a process similar to the assembly that is carried out in CAD software, where assemblies or parts have to be placed in certain (X, Y, Z) coordinates, has to be done. However as LabVIEW is not intended for being a CAD Software this process is not easy and friendly to realize. This assembled has to be done because when a set of VMEs are called and rendered in the same LabVIEW scene by default are placed in the position where they were saved in CAD software during the Solid Creation stage, This position is usually the origin (0, 0, 0); however, the position can be any point in space, and not precisely the correct one, inside the VRM final assembly. For this reason, a set of VMEs are collocated normally overlapped in origin when rendered and it is necessary to place them in their correct spatial position through a LabVIEW reassembly process.

6. ANIMATION PROCESS

The internal programming of the VRM is defined in this process. Animation includes all aspects related to the VRM functionality, as VME movements inside the virtual machine mechanism (VMM), sensors programming, condition validations, signal generations, etcetera. It is necessary to specify that this procedure does not refer to VRM automation control, but animation sequences required for the VMR act dynamically according to control sequences received from the controller. This process is executed in LabVIEW using the necessary operations,

conditions, structures, validation and instructions that the VRM needs for having a behavior the closest to the real process as possible. Animation depends on every VRM functionality and the way that programming is done varies according to VRM's logic programmer. Animation programming is always running inside a while cycle that executes until the user closes the VRM. These instructions are the most demanding for the PC where VRM is running. Animation programming has to be the most efficient possible. As animation affects directly the VME's data, which for VMEs features of high definition and high quality representation of 3D solids are large sets of data storage in clusters. When data are handled or modified, it demands resources from the PC. Despite every VRM having a particular internal animation programming according to its functionality, some important and general aspects independent of the VRM can be underlined. These are VMM programming, and virtual sensors programming.

6.1 VMM PROGRAMMING

During animation programming, VMEs are grouped and turned into Virtual Machine Mechanisms (VMM) using inheritance defined in Assembly process. A Virtual Machine Mechanism (VMM) is integrated by two or more VMEs which are related by at least one dynamic action, one over another and at least one of them affected by a change of signal received from the controller. In VMM some VMEs behave as actuators. This means that, when a change in an external signal is received from the controller, this signal triggers actions inside VMMs as specific VMEs of the VMM are affected usually with transformation operations like rotation, escalation, translation, color change, etc. until another change in the signal is received. To achieve this behavior in VMM, four tasks are carried out: validating the reception of an external signal, recognizing the signal, deciding what action to take and executing the action

6.2 VIRTUAL SENSORS PROGRAMMING

The reception and processing of signals from the controller is closely related to VMM functionality as these are validated and processed inside the VRM. However, the response of VMMs according to these signals has to be fed back to the controller somehow. As in the control programming running in it, the effects that these cause have to be validated to achieve a correct control of the process. In a real process, signals are sent from the process to the controller through many types of sensors that are placed in the correct position inside the process. In a real process, signals are feedback from the process to the external controller through many types of sensors that are placed in a specific position. As VRM emulates totally a real process, the feedback of signals must also be done; this is done in VRM with virtual sensors. Virtual sensors are individual VMEs or VMEs part of VMMs; that in addition include programming oriented to emulate the behavior of the real sensor logic and generates the signals that a real one would generate. When certain conditions inside sensor programming accomplished a change in virtual sensor's state occurs; this is reflected in the same indicator that communicates with the external controller which is established in connection process, this means that the output of a virtual sensor gotten after validations must affect the same variable in the programming that serves as input to the external controller similar to real sensor.

7. CONNECTION PROCESS

After animation process, the VRM could only be considered as a VM (Virtual Machine) for, after animation, this still has no connection to external control devices and is activated only through LabVIEW front panel controls. In VM, signals from virtual sensors are only indicators in LabVIEW front panel. Without accomplishing VRM connection process, an automation strategy implemented in a controller cannot be carried out on the VM. To accomplish VRM functionality, it is necessary to turn the VM into VRM with a connection process that gives VM connection capability. This is achieved through a procedure whose objective is the same that is carried out in a real automated system. That is, to connect the controller outputs to the VM's virtual actuators and the controller inputs to the VM's virtual sensors. However, in contrast to real automated systems, the VM is a software application that emulates mechanisms and sensors; therefore, physical devices do not exist and wires for carrying signals cannot be placed. Then connection between VM and external controller also has to be emulated by

software, what makes that VRM's connection process be particular. This is integrated by 3 steps: Identification, addressing and programming.

8. AUTOMATION VALIDATIONS PROCESS

Once the VRM is created, a general validation of its functionality has to be performed before creating the executable file and ending the VRM creation process. In this process, the complete functionality of the VRM is validated. With this validation, there is assurance the VRM's correct functioning depends totally on the correct control sequences programmed in the external controller by the PLC programmer; this means that if a mistake occurs or there is an incorrect visualization, this depends totally on the control program, not on the VRM functioning. This process includes: Mechanical validation of VMM functionality, virtual sensor's signal validation, animation validation, automation validation.

9. VRM APPLICATION

After the validation process, the VRM's creation process ends. Now, an executable file of the final VRM is created. This is an application file and is created after the program compilation process that carries out LabVIEW. This final application is intended to be used by the external programmable controller. Once the VRM executable file is compiled, an installer is created. The installer makes possible that VRMs behave as any commercial PC program that has to be installed in a PC. The capability of turning the VRM into an executable file offers advantages, both for the VRM usage as for the VRM's user. The most important is that, licensing is not needed for running the VRM executable file. This way, the VRM can be used in PCs by only installing a LabVIEW runtime which is obtained free from the National Instruments website.

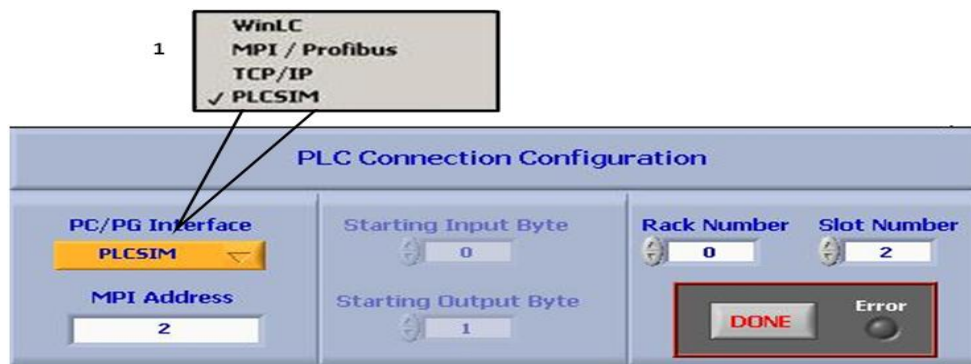


Figure 3: VRM Connection Configuration

VRM feature makes possible that VRMs be practically free and a viable learning option for control engineers, automation students, university laboratories, training centers, etcetera, When the VRM is executed, a configuration box is initially displayed, as shown in Figure 3. Here, the user chooses the type of connection to be established between the external controller and the VRM. The chosen connection goes according to the controller that the user has available and the communication protocols it handles

9.1 COMMUNICATIONS PROTOCOLS HANDLED FOR THE VRM

According to the chosen communication protocol, it is necessary to write the address of the external device used to control the VRM. This is the number that identifies the programmable controller where data are going to be sent and received. This address is necessary for communication libraries and varies depending on the addressing structure handled by the protocol. In figure 4 it is shown the VRM communication diagram when SIEMENS hardware and software are used to automate the VRM. This exemplifies the VRM's communication with the external control devices (Salazar and Macías, 2009).

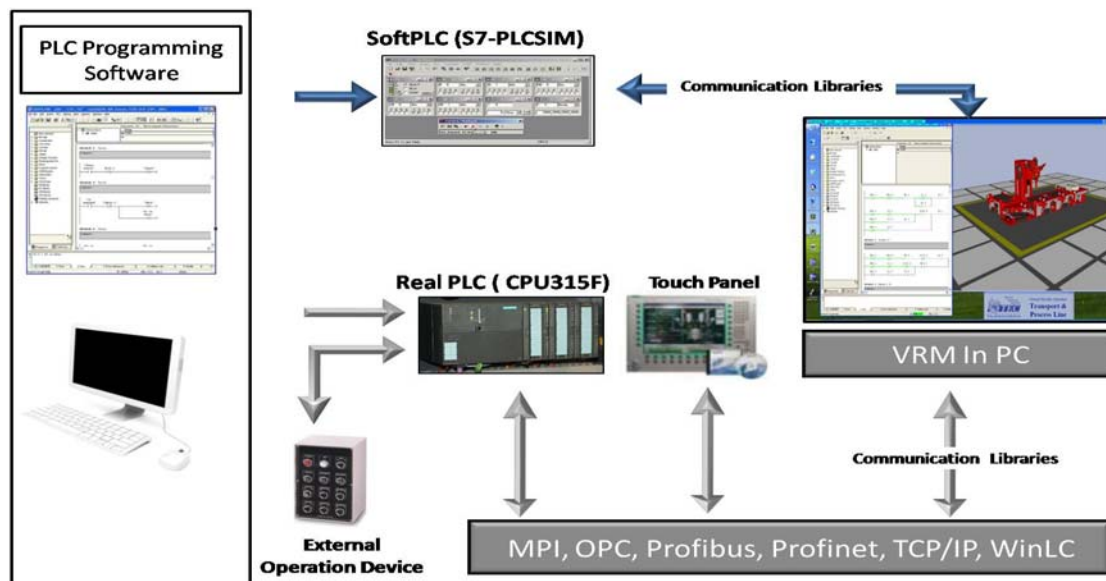


Figure 4: VRM Communication Diagram

Figure 4 shows that there are two ways of connecting the VRM to the controller: by softPLC (in this case S7PLCSIM is used) or by a real PLC (in this case the CPU315F is used). If PLCSIM connection is chosen VRM and PLCSIM will communicate only through software, this means there will be no connection to a real hardware and the VRM's user will work with a PLC emulation and with a process emulation without interconnection between real and virtual world. If PLCSIM is used, it must be installed and running in the PC that will communicate with the VRM. Monitoring of the I/O signal and manual activation of actuators can be done in the PLCSIM software. When TCP/IP, Industrial Ethernet, MPI or PROFIBUS protocols are chosen, it is necessary to have an external PLC with the communication interface that supports these protocols.

9.2 SOLUTION FOR AUTOMATION EQUIPMENT

For plenty accomplishing its objective, automation and control laboratories need an infrastructure that in addition of industrial automation hardware and software also includes a process or plant where students implement, test, debug and validate their control and automation strategies. However this represents the highest costs for laboratory equipment and in the best scenario laboratories are equipped with few real automated devices used in industry which because of their high cost are commonly used only by professors for demonstrative purpose. Another option is the usage of training workstations of vendors such as Festo, FISHER Technics, etc. that as a consequence of their price, are used only for a reduced number of universities. Other practice is testing students control programs in a set of individual industrial sensors and actuators that activate according to the control programmed, however these don't represent a real industry manufacturing process. A common practice is the construction of home-made systems that are a set of buttons, actuators and led indicators that with imagination try to represent a process. The construction of primitive structures as elevator, mixer, etc. is also a common practice. In addition that any one of these options has a cost, some of these don't achieve to impact totally in the students understanding, as some students don't visualize the essence and sense of automation and control strategy carried out on these basic laboratories' resources. For this reason the automation and control laboratory station shown in figure 5 is proposed. This represents a viable solution to the lack of equipment that face many universities, as the highest cost of laboratories' equipment is represented for the process to control, what is solved with the VRM as this emulates a real process that students can automate. In addition VRM makes possible that laboratories have a set of VRMs in every station without additional cost what means that every student can has his/her own process to automate. The Automation Lab Workstation proposed is integrated by a Real PLC, a touch panel and a PC where the VRM executes, and the PLC and Touch panel software are also installed.

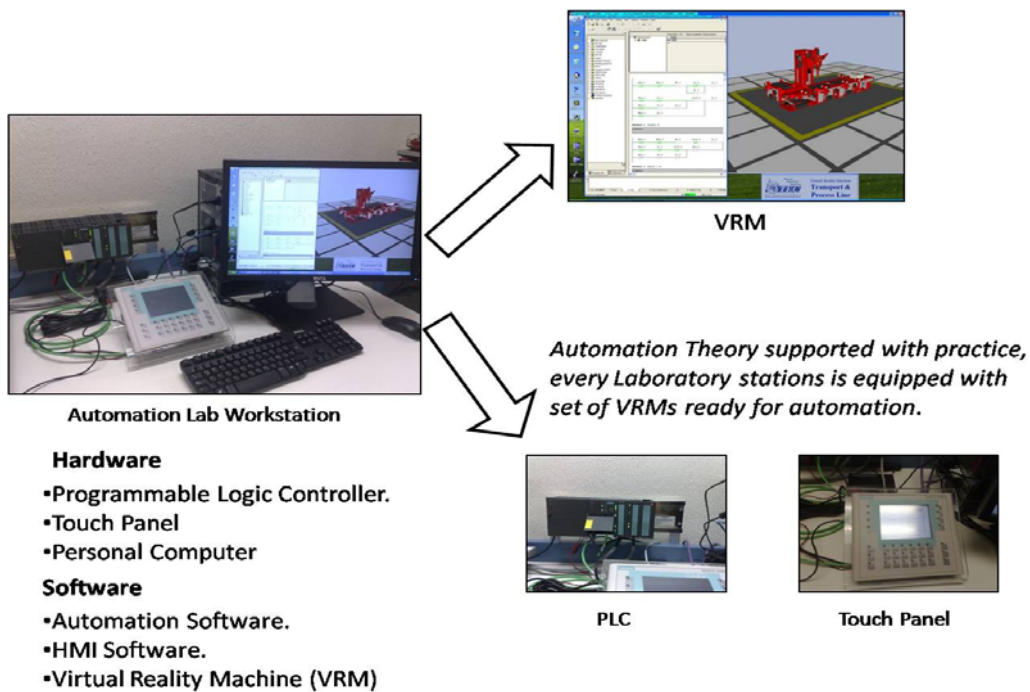


Figure 5: Automation & Control Laboratory Equipping Proposed

10. CONCLUSIONS

In engineering theory is closely related with practice. The level of integration between theory and practice and how they are complemented in universities is reflected directly in students' formation as the lack of practice and the deficient training received for students in universities impact directly in their competitive advantage, what is reflected in the job opportunities for students. In many countries where manufacturing is one of the principal activities, automation and control careers are a key aspect for industry, and engineers with this orientation are constantly wanted. However because of lack of practice many of these engineers haven't the skills needed for some manufacturing companies, what is reflected in some aspects as of their personal life as in the country growing.

According to the VRM's features presented, the usage and implementation of these as process emulation where students test, debug and validate their automation and control strategies, is an option for educational institutions with a not enough laboratory infrastructure. These are a high performance and low cost solution that resolves the problem of lack of laboratory equipping that face many low resources universities, principally in developing countries. Even help to any university to complement and supporting the automation and control course offered. Despite that with VRM investment in automation hardware and control must to be done this cost represents only an small portion of the total investment required for equipping a laboratory with traditional methods, as the final process where to implement the control is the most expensive part of the equipment necessary in laboratory. With VRM this cost hasn't to be paid as there isn't necessity of licensing and the emulations are so close to reality what for automation and control practice objective can be considered just like a real process, as the knowledge transmitted to students when the control strategy is programmed on these is the same that if this would has been carry out in a real process. VRM offers to student the possibility of complement his/her educational training, supporting the knowledge acquire in automation and control courses with process emulations that impact directly in the grade and quality of training received in laboratory, skills acquired and knowledge won in classroom and laboratory. This practicing impacts directly in the student formation and training what reflects later in their professional life as gives add value to student's resume and prepare them for some of the problems that will face in industry.

11. ACKNOWLEDGMENT

The authors wish to acknowledge the support of the Tecnológico de Monterrey, Campus Monterrey to develop this work which is a contribution of the Computerized Design of Manufacturing Cells Research Project, Registration No. PRY059.

REFERENCES

- ARC Advisory Group. (2006). “DELMIA V5 Automation Platform: Merging Digital Manufacturing with Automation”.
- Berger, H. (2008). “Automating with Step 7 in LAD and FBD”. Erlangen: Publicis Corporate Publishing.
- Bitter, R., Mohiuddin, T. and Nawrocki, M. (2007). LabVIEW Advanced Programming Techniques Boca Raton CRC PRESS.
- Caie, J. (2008), “ARC White Paper, Discrete Manufacturers Driving Results with DELMIA V5 Automation Platform”.
- Carey, R., Addison-Wesley and Bell, G. (1997). Annotated VRML 2.0 Reference Manual, Reading Massachusetts, 1997.
- Dassault Systems. (2007). Virtual Commissioning Solution: A realistic and intelligent virtual environment for Control Engineers to debug and validate PLC Programs, 2007.
- Davidson, J., and Sennö, T. (2005). Interactive Control of a Virtual Machine. Department of Industrial Electrical Engineering and Automation, Lund University, Lund, SWEDEN.
- Izquierdo, R. (2008). EasyPLC Machines Simulator Demo Instruction Guide, <http://www.nirtec.com>, 12/10/2012.
- Ing. Buro Weiss. (2005). 2005 SPS-VISU mHj software, <http://www.mhj-online.de/de/> 12/10/2012.
- Macías, M.E., Guridi, E.D. and Ortiz, A. (2007). “Extending the Laboratory Concept with Computer Emulations in Automation”. Milwaukee : Electrical and Computing Engineering, ITESM Campus Monterrey, 2007. IEEE Frontiers in Education Conference. pp. S3G-18 - S3G18-22.
- Macías, M.E. and Guridi, E.D. (2010). “Computer Emulations t Support Training in Automation”. [book auth.] Guedes, L.A. (2010) Programmable Logic Controller. Croatia : s.n., pp. 151-160.
- Mueller, J. (2005). “Controlling with SIMATIC”. Nuremberg: Publicis Corporate Publishing.
- Salazar, E.A. and Macias, M.E. (2009). Virtual 3D Controllable Machine Models for implementation of Automations Laboratories. San Antonio : Electrical and Computing Engineering, ITESM Campus Monterrey, 2009. IEEE Frontiers in Education Conferennce. pp. T2C1 - T2C5.
- Tarakos GmbH. (2003). Tarakos 3D visualizer manual and control tutorial, <http://www.tarakos.com/eng/Produkte/control.html>, 12/10/2012.
- Travis, J., and Kring, J. (2008). LabVIEW for Everyone. Upper Saddle River: Prentice Hall.
- UGS Corp. (2005). eM-PLC and STEP 7 Professional – eMPower for manufacturing process management Model-based, automatic PLC program generation, simulation and verification.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.