SOFTWARE SIMULATION: A TOOL FOR ENHANCING CONTROL SYSTEM DESIGN

B. Sze and G.H. Ridgway

GE Hitachi Nuclear Energy Canada,
Peterborough, Ontario, Canada
{beatrice.sze, guy.ridgway}@ge.com

ABSTRACT

The creation, implementation and management of engineering design tools are important to the quality and efficiency of any large engineering project. Some of the most complicated tools to develop are system simulators.

The development and implementation of system simulators to support replacement fuel handling control systems is of particular interest to the Canadian nuclear industry given the current age of installations and the risk of obsolescence to many utilities. The use of such simulator tools has been known to significantly improve successful deployment of new software packages and maintenance-related software changes while reducing the time required for their overall development. Moreover, these simulation systems can also serve as operator training stations and provide a virtual environment for site engineers to test operational changes before they are uploaded to the actual system.

1. INTRODUCTION

Typical challenges facing today's nuclear power plants in Ontario include: managing obsolescence risk to aging utilities; reducing cost, and schedule overruns during outages; and developing Operator Experience (OPEX) training programs to facilitate the transfer of knowledge from the industry experts to those just entering the workforce.

The development of computer-based simulation tools can aid in all of these major efforts by providing the following: test platforms for designers to use in the rapid development and upkeep of fuel handling control systems, a foundation for computer-based operator training systems, and a virtual environment for both operations and field engineers to test new procedures and troubleshoot control problems.

This paper outlines the best practices and achievements of recent large-scale software-based simulator computer systems created by GE Hitachi. It also discusses their use as engineering test environments, operator training platforms, and process development and troubleshooting tools. The two simulation systems that shall be discussed here are the Universal Delivery Machine (UDM) simulator and the Retube Tooling Equipment (Retube) simulator.

2. Software Simulation – A Justifiable Cost

In today's changing energy market, software solutions developed for CANDU reactors need to be fast, cost effective, of high quality, and adaptable.

Reducing the time it takes to deliver software solutions to customers allows them to maintain outage schedules and ultimately their outage budget for the year. One way of reducing delivery time is to streamline the software design process by reducing the number of test iterations in the design cycle and reducing the time required to perform reliable development and validation testing.

Reducing the cost of control software development can be accomplished by reducing the cost of prototyping and reducing the time required to run development and validation testing. Whereas a complete physical prototype can faithfully mimic the behaviour of a production machine, a software simulation system empowers operators to jump to particular machine-states in the operational procedures without having to go through all previous steps to achieve that state. Furthermore, while a physical prototype may initially match the production machine's performance initially, it will likely age differently from its production counterpart, having never been used in the production environment. This progressive departure in behaviour reduces the prototype's fidelity, and ultimately its value, as a test platform for control software. A software-based simulator, however, will not degrade in behaviour over time. Its parameters can be continually tweaked and changed a to match those of the aging production machine. This allows it to serve as a high fidelity, high value test platform for the production machine's entire service life.

Delivering products of high quality means providing engineering solutions that are correct and sustainable. This means delivering products that work the first time. There should be few reported "snags" and it should continue to work properly throughout the intended machine life with minimal maintenance. This can be accomplished by using simulators in the software development cycle that allow for reliable testing of the software package before it is released.

Developing software products that are adaptable can be helpful to customers looking for tools that can help them tackle a wide variety of challenges, ranging from obsolescence management to the development of OPEX training programs. Thus developing software simulators that can be easily adapted to other plant operations would be highly desirable.

On the other hand, depending on the level of fidelity required, software simulators can be costly to maintain for the life of a mechanical system. This can be minimized if off-the-shelf development packages are used as they often provide extensive libraries to support more complex process and motion control functions. However dependence on third party developer packages can introduce layers of complexity if the package is not designed for the desired application or simulation developers do not correctly implement it.

3. SOFTWARE SIMULATOR SYSTEMS

The simulator tools developed by GE Hitachi are PC-based systems that mimic the electrical feedback of complex field equipment to PLC-based controllers. These tools provide process data to the PLC controllers in response to controller commands to simulated field devices. The intent is that the simulator should act as a platform on which human operators and computer control systems can receive simulated field responses to commands sent to simulated field equipment.

Two GE Hitachi projects that have made use of simulators are the UDM and the Retube system. This section describes the reactor work that the simulations were used to support, the simulation software architecture, and the benefits and limitations of their use.

3.1. Universal Delivery Machine

The UDM was developed by GE Hitachi eight years ago for Ontario Power Generation. It is the primary platform for the delivery of inspection and maintenance tools to reactor fuel channels during outages at Bruce Power, Darlington, and Pickering.

The UDM is used to perform the following functions:

Manipulation of channel end fitting components—This consists of homing and locking onto a channel end fitting, removing and replacing the closure and shield plugs, and coming off the end fitting when the inspection and maintenance operations are complete.

Tool delivery and manipulation—The tool control system has control of these operations. The UDMs developed for Bruce Power and Pickering have the additional function of defueling and refueling the channel. When required, the UDM pushes fuel out of the channel and into a fueling machine locked onto the opposing end fitting. This empties the channel to allow inspection and maintenance operations to be performed. Once these operations have been completed, the fuel is returned to the channel by the fueling machine and the two machines return the channel end fittings to a safe state.

Figure 1 shows the UDM at a reactor face.

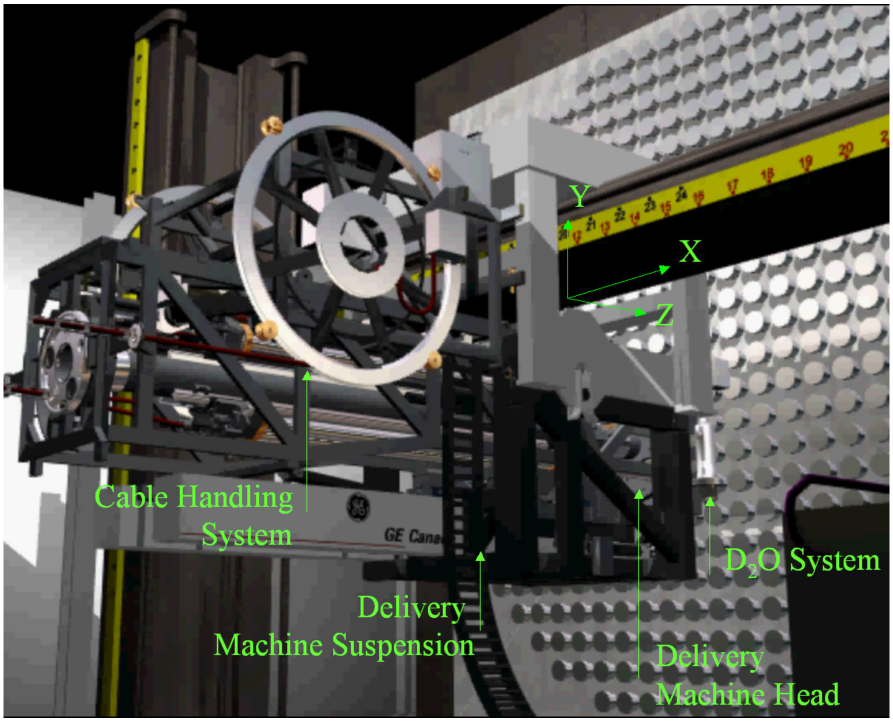

Figure 1 – Universal Delivery Machine at Reactor Face

3.1.1. UDM Mechanical Overview

The delivery machine is moved to a reactor face on a fueling machine transport trolley. It is then picked up by the same carriage and bridge that would normally move a fuelling machine and is moved into coarse X (lateral) and coarse Y (vertical) alignment with a fuel channel end fitting using the existing fuel handling control system.

The head assembly consists of a magazine subassembly that is mounted onto a steel support frame, which is in turn mounted within the UDM suspension. The suspension provides fine X, fine Y, and Z motion. Once aligned with the channel end fitting by the fuel handling system, the machine is advanced under control of the UDM system and the snout assembly, mounted on the front of the magazine, locks the head onto the reactor end fitting. The snout is equipped with control instrumentation used during homing of the head onto the end fitting to sense lateral, vertical, and pitch misalignments.

Mounted on the magazine assembly are the various mechanisms required to remove and install the closure and the shield plug and to deliver the various inspection tools into the channel. A common drive is mounted to the support frame to provide axial and rotary motions required for the shield plug and inspection tool.

3.1.2. UDM Control System Architecture

The UDM control system comprises a central host controller connected to distributed I/O modules located throughout the UDM. The host controller is a programmable logic controller (PLC) that interfaces with other control modules mounted on the PLC backplane. These modules include motion controllers and communication network controllers used for the distributed input/output system and for passing data to and from the operator interface. See Figure 2.

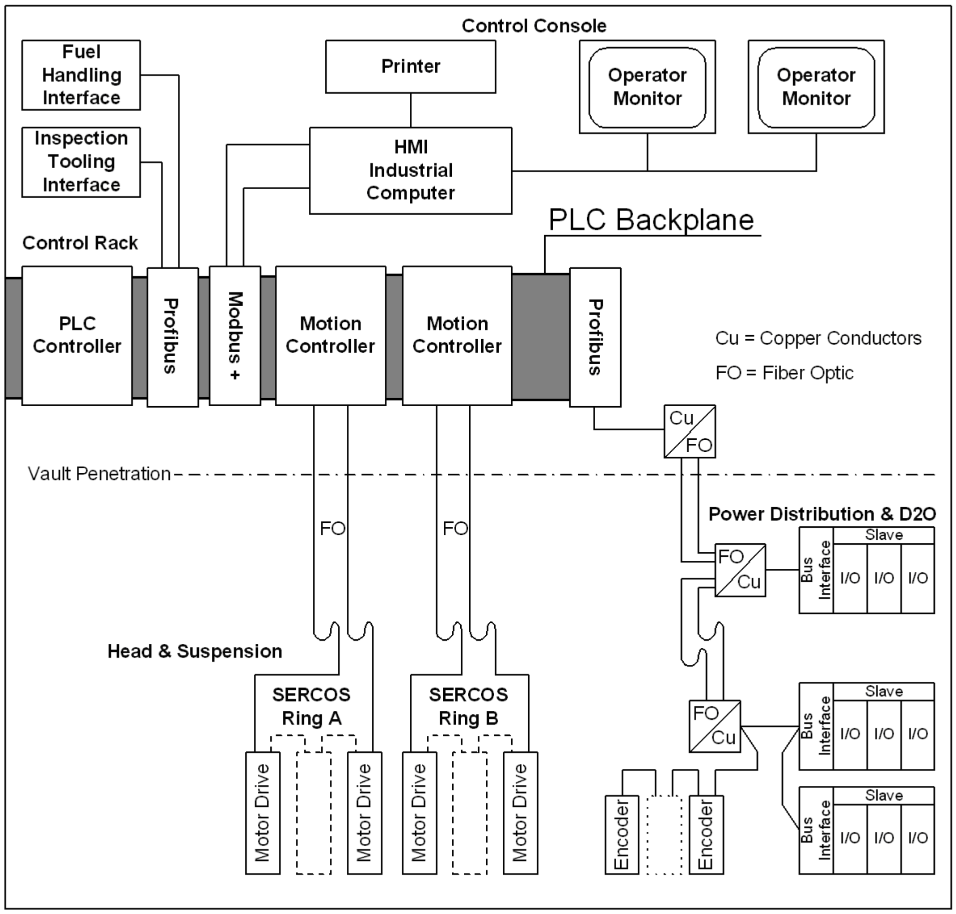

Figure 2 – UDM Control System

Programmable Logic Controller – The PLC executes an application program that controls the operation of the machine. The PLC controller used in this project was the Schneider Automation Quantum 586 running an application programmed with Concept, a PC-based utility compliant with the IEC 1131-3 PLC programming standard, running on Microsoft Windows NT.

Motion Control System –The PLC issues commands to the 486-based motion controllers over the backplane. These instructions are translated into commands passed to the drive amplifiers, which provide the modulated power signals that control speed and torque within the servomotors.

The motion control system is based on DC brushless servomotors controlled through a multi-axis fiber optic control bus called SErial Real-time COmmunications System (SERCOS). SERCOS is a fiber token ring bus system that operates at a speed of up to 4 MBit/s. The motion controllers control each of the drive amplifiers through the fiber optic ring.

Human Machine Interface—The HMI system consists of an industrial PC running Schneider Automation's Monitor Pro software package on a Windows NT operating system.

Fieldbus—Each UDM control system has two PROFIBUS Class 1 masters. The PROFIBUS electrical cable feeds from the field I/O master into a converter that changes the physical medium from copper wire to optical fiber.

3.1.3. UDM Simulation Software Architecture

The UDM simulator provides process simulation data to the PLC in response to operator inputs from simulation displays or by animated response to the PLC commands. The intent is that the PLC inputs should behave as if there is an actual UDM being controlled. Simulated feedback from drive motors indicate movement in response to speed and position requests from the PLC and proximity switches change state based on simulated motor movement. Also level indicators rise and fall in response to commands sent from the PLC to turn pumps on and off. Figure 3 shows which components of the control system are replaced when the PLC is receiving inputs from the simulator computer system. When in simulator mode, the PLC receives I/O from a remote I/O module on the backplane rather than from field devices (the Profibus modules are physically by the remote I/O module). When simulation mode is activated, the simulator simulates the signals of the shaded components of Figure 3.

One of the most powerful features of the UDM simulator is its ability to simulate all or a selected portion of the field inputs. This allows the control system to be tested using a combination of signals from existing field devices and simulated signals to cover any non-existent devices. All or a portion of the crosshatched components shown in Figure 3 may be simulated.

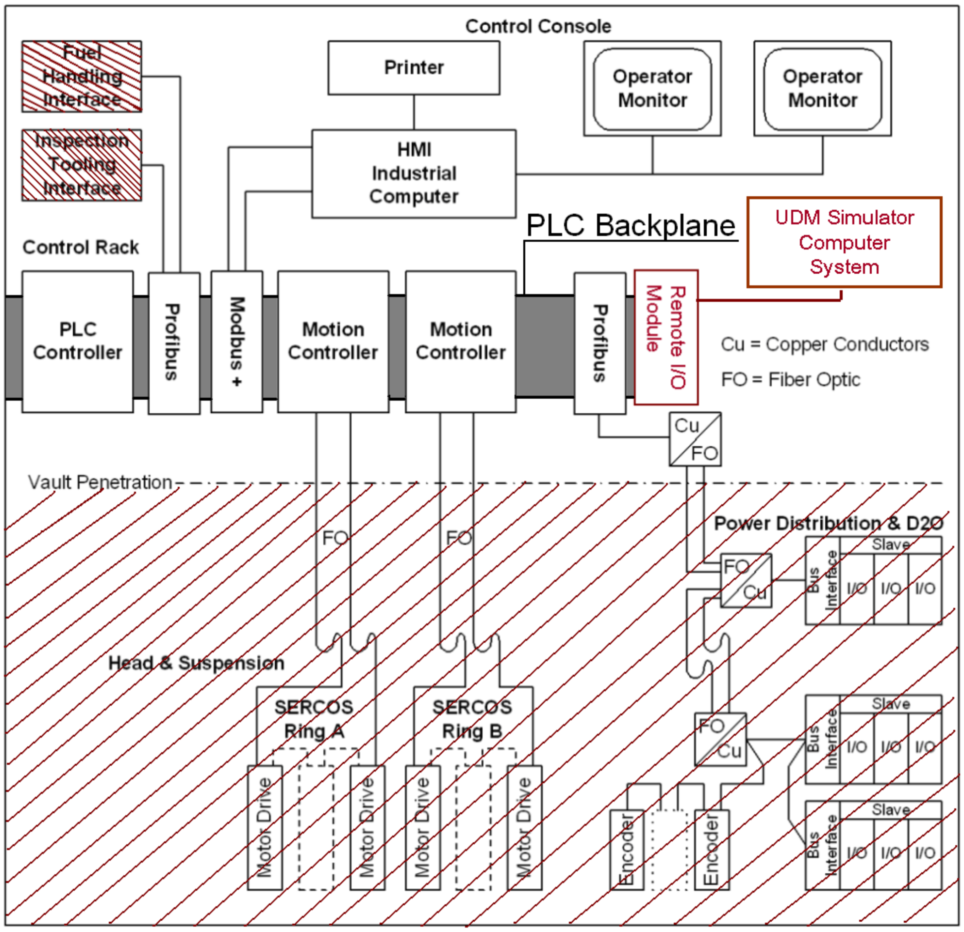

Figure 3 – UDM Control System In Simulation Mode

The simulation application was developed using SST's[1] "PICS® Simulation For Windows" software.

The translation from simulator and real field inputs to data used by the PLC logic in execution of control functions is expressed in the following formula,

Result = (Actual Input AND NOT (Sim Mask)) OR (Sim Data AND Sim Mask)

Where,
"**Actual Input**" refers to input from existing field devices, read from the Field Inputs.
"**Sim Mask**" is a register specifying which inputs are to be simulated

[1] SST is now a trademark of Molex Inc.

"**Sim Data**" is a register specifying the desired values for the simulated inputs
"**Result**" ("**Sim States**" in Figure 4) refers to the final values that the PLC uses for execution of the control logic

Through the simulator's graphical interface, the user sets the bit values for the Sim Mask to specify which bits are to be simulated, and the Sim Data to specify the states for the simulated bits, thereby overriding the field states for those bits.

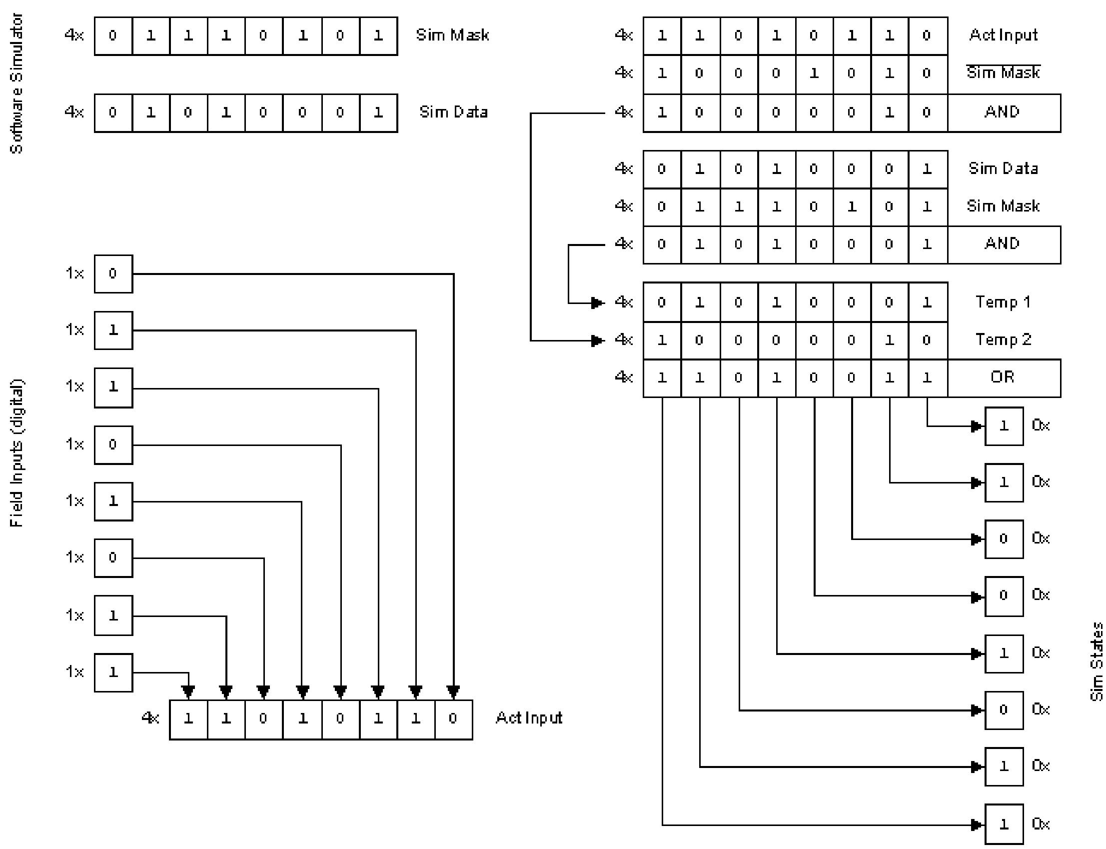

Figure 4 – UDM Simulation Logic

Consider the example presented by Figure 4. Assume that each bit in the Field Inputs word represent the states of eight valves, where 1 indicates valve closed and 0 indicates open. Counting from right to left in register "Act Input" we see that field valves 2, 3, 5, 7, and 8 are closed and 1, 4, and 6 are open. The simulator user then specifies which valve states are to be simulated by setting states on the simulator's graphical interface. In the example, the user has decided to simulate the states of valves 1, 3, 5, 6, and 7, as indicated in the value for "Sim Mask". "Sim Data" then indicates that the user has decided to set valves 1, 5, and 7 closed and valves 3 and 6 open, regardless of the true states of those valves in the field. The states of the remaining valves (2, 4, and 8) are not simulated and the PLC uses their actual field values. "Sim States" values are determined by the translation formula and properly indicate that valves 1, 2, 5, 7, and 8 are closed.

3.1.4. **Benefits of the UDM Simulator**

The key benefits of the UDM simulator are described below

Capable of running with physical components online – Because input signals to the PLC are determined from the simulation mask and data, the UDM simulator is able to work in tandem with various configurations of actual field components of the UDM when it comes time to validate the controller.

Complete I/O access –The simulator displays the status of all I/O points associated with the PLC. This allows engineers developing the UDM controller application to use it as an effective platform for development and validation testing.

Simulating abnormal conditions – The simulator user is able to force the states of field inputs to simulate a system fault in order to monitor the PLC's response to abnormal conditions. The trainee then becomes aware of the system's behaviour through information presented on the HMI, such as alarm and log messages.

Development and training modes – The simulator has two modes of operation: development and training. Development mode, accessible by password, allows users to make changes to the simulation in addition to downloading and running it whereas training mode only allows users to download and run simulations. This allows for the safe and reliable use of the simulator as a training station by operators at Bruce Power.

Training using simulation scenarios – With the use of simulation "scenarios" the trainee is able to jump to a particular point in a complex sequence of steps, representing a significant saving in the time it would otherwise take if the entire sequence had to be run in real time. Training efficiency can be further improved by running a simulation of a sequence of process steps in an accelerated, rather than real-time mode, a feature of more advanced simulation packages.

3.1.5. **Limitations of the UDM Simulator**

One of the major limitations of the UDM simulator is that it does not accurately simulate the following aspects of the SERCOS Motion Controller.

Drive acceleration/deceleration—All speed changes are instantaneous. Motor acceleration/deceleration characteristics are not modeled in the UDM Simulator largely because this functionality was not required for the development and validation testing of the controller at the time. For future simulator projects, it is worth considering whether acceleration/deceleration profiles are pertinent to the process control problem and implementing them if necessary. This would allow for a wider range of "what-if" scenarios to be tested. However, higher fidelity models are usually associated with higher cost and longer development test time so this decision should take into account the project budget, timeline and required level of detail from the simulator.

Motor current—Similarly, the UDM simulator does not model effects of drive speed, acceleration, or process loading. This is also largely because this functionality was not required for the development and validation testing of the UDM controller at the time of application development. As with the acceleration/deceleration characteristics, this might be a desirable feature to add to future simulator projects.

Subroutine call timing—For both the Retube and UDM simulators accurate timing of signals was not required for the development and validation testing of the controller at the time. The UDM simulator happens to respond faster than the field devices it's mimicking. Like the drive acceleration/deceleration feature however, for future simulator projects, it is worth considering whether timed-communication profiles are pertinent to the process control problem and implementing them if necessary. Because the PICS® software executes subroutines based on computer clock, often the subroutine responses are faster than what would normally occur in the field. Timers could have been used to more accurately simulate field responses, but there was no data to characterize system time delays. This would allow for a wider range of "what if" scenarios to be tested. However, like the drive acceleration/deceleration profiles higher fidelity models are usually associated with higher cost and longer development test time so this decision should take into account the project budget, timeline and required level of detail from the simulator.

3.2. Retube Tooling

A number of Retube tools were developed by GE Hitachi under contract to AECL as part of a primary tool set used for the dismantling of the Bruce A's unit 2 reactor.

These tools were used to perform the following functions:

- *Stage 1:* Cutting the pressure tube (PT) at each end of the channel
- *Stage 2:* Cutting the bellows and stop collar on each end of the channel
- *Stage 3:* Removing the end fitting (EF) and replacing it with a lattice tube guide sleeve (LTGS) and lattice tube shield plug (LTSP)
- *Stage 4:* Removing the Flow Restricting Outlet Bundles (FROBs)
- *Stage 5:* Removing the PTs
- *Stage 6:* Releasing the calandria tube insert (CTI)
- *Stage 7:* Removing the CTI
- *Stage 8:* Removing the calandria tube (CT)

GE Hitachi was responsible for the development of the pressure tube severing tool (PTST) used in stage 1, the end fitting shield plug insert and removal tool (EF SPIRT) used in stage 1, the end fitting removal tool (EFRT) used in stage 3, and the lattice tube shield plug insert and removal tool (LT SPIRT) used in stages 3 to 8.

3.2.1. Retube Tooling Control System Architecture

The Retube control system is more extensive than the UDM's. While the UDM is a single tooling machine controlled by one centralized PLC controller, the Retube system consists of multiple tools, each with its own PLC controller, all of which are coordinated by a master sequencer PLC. Figure 6 shows the communication between the tool PLCs, the sequencer PLCs, the HMIs, and the Retube simulators for the east and west face systems.

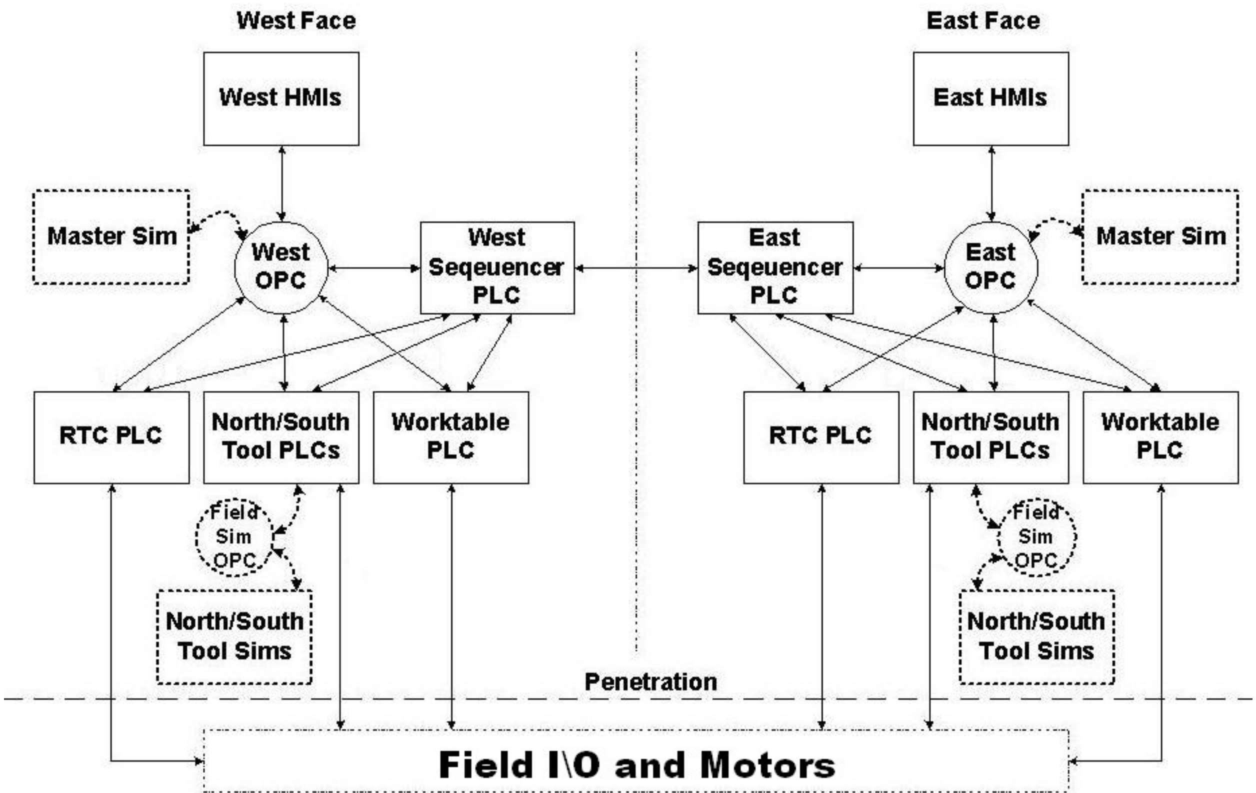

Figure 5 – Retube Control System

3.2.2. Retube Simulator Software Architecture

As with UDM, the Retube simulator, also developed using the PICS® software, provides process and simulated field data to the tool PLCs.

Unlike UDM, the Retube simulation system consists of two types of simulators: tool simulators and master simulators.

As indicated by the horizontal crosshatching in Figure 6, the tool simulators are responsible for mimicking the field I/O for individual tools, enabling software developers to test those tools' control programs. Diagonal crosshatching indicates that the master simulators are responsible for providing process simulation data and handshaking signals to the sequencer PLCs and tool PLCs in the absence of one or more of the tool, worktable, or RTC (Retube Tooling Carrier)

PLCs. It also provides simulated signals to represent any local I/O devices belonging to the sequencer that are not present during development testing, and provides some limited support for the I/O signals between the tool PLCs and the HMIs. The intent is to provide the HMIs with adequate feedback from tool systems to prevent nuisance alarms during test.

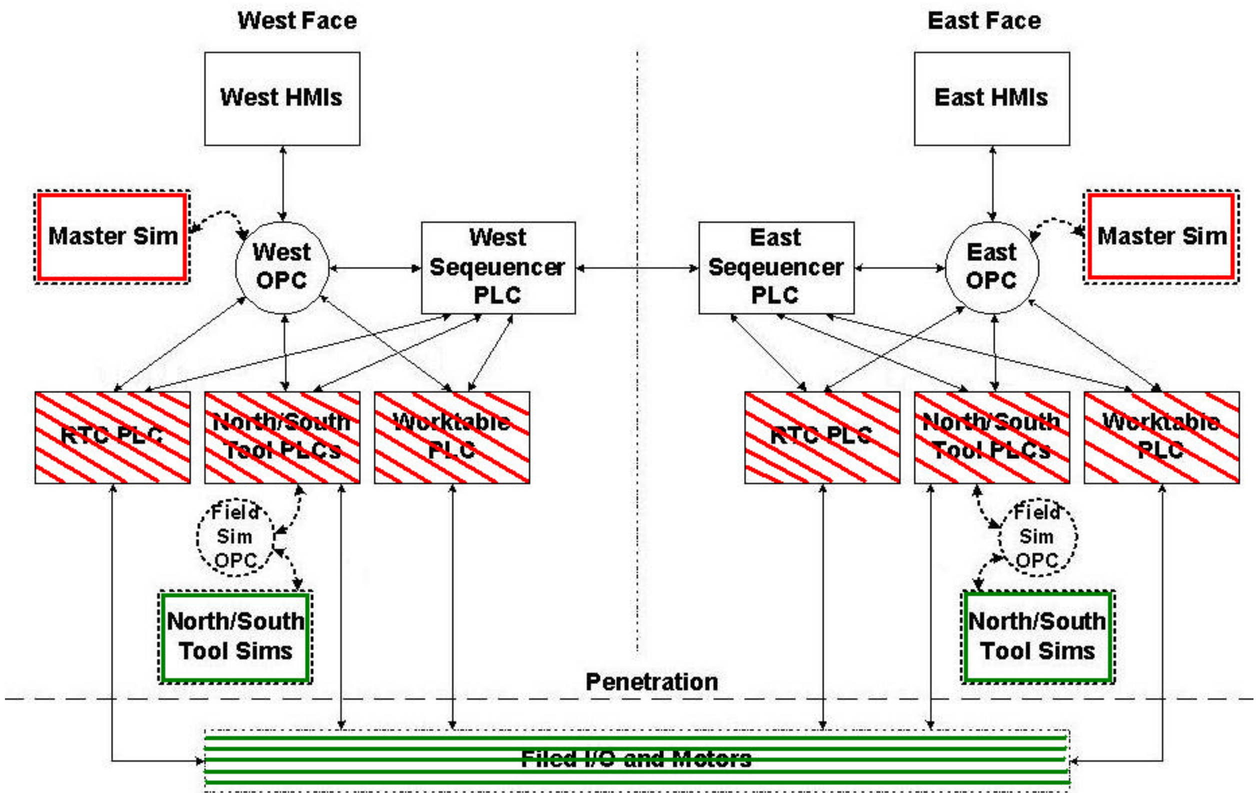

Figure 6 – Retube Control System In Simulation Mode

Figure 7 shows how the master simulator can be used to test the sequencer PLC.

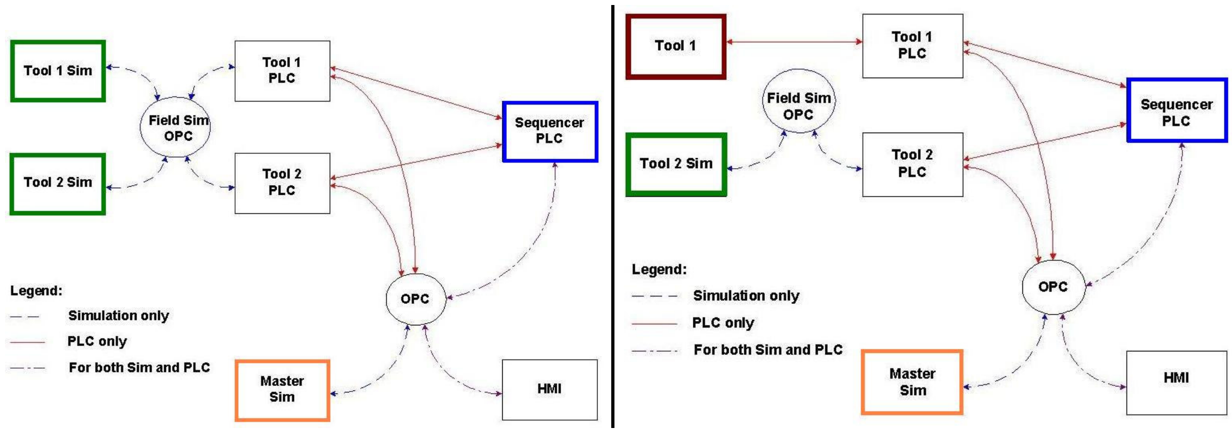

Figure 7 – Tool and Master Simulators

The figure on the left shows the test setup for the sequencer PLC when the tools are not physically present. The tool 1 and tool 2 simulators provide I/O signals to the tool 1 and tool 2 PLCs and the master simulator provides the handshaking signals for all other Retube system components (tools, worktable, RTC, etc.) to the sequencer PLC.

The figure on the right shows the test setup for the sequencer PLC when one of the tools is physically present. The tool 2 simulator provides I/O signals to the tool 2 PLC, while the tool 1 PLC receives its I/O directly from tool 1. Again, the master simulator provides the handshaking signals of all other Retube system components (tools, worktable, RTC, etc.) to the sequencer.

3.2.3. Benefits of the Retube Simulator

Effectively, all the benefits outlined for the UDM simulator also apply to the Retube system.

One difference between the two systems is their respective abilities to run with physical components online. While the UDM simulator can be configured to simulate all or a portion of the field I/O, down to the level of discrete devices, the Retube system only allows for an "all or nothing" simulation for a given tool's complement of I/O. The more sophisticated capability of the UDM system was not required for the Retube design. Omitting this feature meant faster deployment of the Retube tool simulators, but resulted in a less flexible test platform.

For future projects, the question of flexibility versus development cost, specifically pertaining to the system's ability to run with field devices present, should be considered early on in the development life cycle

3.2.4. Limitations of the Retube Simulator

One of the major limitations of the Retube simulator was that it did not accurately simulate the timing of signals to the various PLC controllers.

Limitation of Time-Critical Communication — It can be difficult to design a simulator based on third-party software to achieve high fidelity with respect to time-critical performance. This could be a limitation of the software development utility, problems with the implementation, or poor definition of performance requirements for the simulator.

For future projects, it is worth considering early on whether high fidelity time-critical communications are pertinent to the process control problem. Prototyping the simulation system with target communication platforms can resolve questions about performance if high fidelity is a requirement.

4. CONCLUSION

Software simulators can be cost-effective tools for use in the development, testing, and maintenance of control system designs for nuclear plant applications. Alternatives such as production machine prototypes are limited in their flexibility, making testing and maintenance difficult. Some means for comprehensive testing of changes to a custom-developed control application are usually required once the production system has been in operation within the plant for some time. Once production components are installed within the plant's containment boundary, direct access to those components becomes impossible for testing purposes. A flexible simulator design, with an appropriate level of fidelity, provides greater confidence in the application change before it is installed in the production system. Furthermore, a simulator enables testing of abnormal operating conditions, or "what-if" scenarios, and thereby offers a powerful training tool for operators and maintenance staff.

GE Hitachi has successfully designed effective simulation solutions for complex control projects using off-the-shelf simulation software. Lessons learned from our experience have shown that the option for software simulation should be one of the first considerations in the requirements phase for any large-scale control system design.

At GE Hitachi, our commitment to provide innovative and reliable solutions remains a cornerstone in our engineering development philosophy.

5. ACKNOWLEDGEMENTS

The authors of this paper would like to thank Jean Wang[1] and Fang Tong[1] for their contributions to this paper.

[1] GE Hitachi Nuclear Energy Canada, Peterborough, Ontario, Canada