DYNAMIC POSITIONING CONFERENCE

October 14-15, 2014

RELIABILITY SESSION

System Verification Helps Validate Complex Integrated Systems

By Naveen Selvam

American Bureau of Shipping (ABS)

Return to Session Directory

Abstract

As building and operating offshore assets become more complex, and more integrated, one of the biggest challenges is centered on software. The move toward automation of offshore assets has allowed drilling and production systems to work much more efficiently. However, the introduction of complex, integrated control systems also poses new challenges. These systems can present enormous problems during operations and maintenance without thorough testing of the software. Hardware-In-the-Loop (HIL) testing is one option used for testing software actions and for crew training. This paper provides insights into the usefulness of HIL test and the benefits it had on a recently completed drillship.

Computer-based control of nearly all equipment onboard marine and offshore assets is ubiquitous, offering increased safety and efficiency but at the cost of increased complexity and increased risks of a failure with the associated consequences. Verifying system integrity is critical to identifying and mitigating system failure to safeguard marine and offshore operations. HIL testing is a method that is used in the development and testing of complex real-time integrated systems. Greater understanding resulting from such testing can be used for risk reduction and crew training by evaluating whether the system performs as expected or whether there may be a safer action to mitigate or manage the failure in a better manner.

The marine and offshore industries are realizing the important role of software in controlling the actions of the equipment aboard and especially the actions of the equipment upon a failure. Software verification allows assessment of the equipment during normal, degraded, and failed states. This assessment allows for the identification and mitigation of safety, business and environmental risks with a bonus of providing useful training for the crew. The owners of these assets are requesting some form of guidance in testing these complex control systems. Improved software quality through the use of standardized methodology provides assertion that today's complex systems perform as intended and as expected. The best method is using HIL testing with a detailed verification plan. HIL testing allows for discovering faults before deployment, leading to lower costs and lower risks of any schedule impacts. Additionally, software HIL testing is possible after commissioning for software updates and modifications made during commissioning and beyond. System verification testing can be applied after the system has been installed on existing assets.

Safety and reliability of integrated software-dependent systems is now becoming an industry demand. A control system interacts not only with the operator but also with the numerous connected control systems, input and outputs, as well as the data sources. Cascade failures are possible with integrated systems, resulting in faltering operations, reducing efficiency, and increasing non-productive time (NPT). Integrated software testing using the HIL method may help prevent these failures. The system verification process addresses these challenges and provides guidance for properly testing the control systems. By testing the essential functions within the control systems, it increases owner and operator confidence in the vessel’s performance. This paper provides a testing method for software-dependent control systems that aim to reduce the risk of software failures and improve operational efficiency.

Introduction to HIL

HIL testing is a technique for performing system-level testing of embedded systems in a thorough, cost-effective, and efficient manner. This method wasdeveloped by and is used today in the automotive and the aviation industry. A control system interacts with its environment and other connected systems through a set of Inputs/Outputs (I/O) and networked data channels. All inputs/outputs and network data are held in registers within the control system before being written to the I/O cards or transmitted over the network. To the control systems program, the world is made up of registers containing bits, bytes and other data that the program reads to make decisions upon and writes data to the output registers. Inputs are provided by many different sensors that measure dynamic operating states and parameters of the equipment, as well as inputs from operator stations and other control systems. Based on these inputs and the way the control system is programmed, the control system calculates the necessary signals that are sent to actuators via these I/O channels. By isolating the control system from its environment and having the HIL simulator read and write to the registers, the HIL simulator mimics the environment that the program is trying to manipulate or control. HIL simulation provides an effective verification testing method by including the complexity and the criticality factors of the Equipment Under Control (EUC) as part of the test platform. Hardware-In-the-Loop is a form of real-time simulation to verify that the control system is programmed to behave as expected and to meet the requirements. Hardware-In-the-Loop differs from real-time simulation by the addition of a real component, which controls the equipment, in the loop. HIL test focuses on discovering error sources associated with the software operating within the equipment's control system. This component that will be verified may be a Programmable Logic Controller (PLC), Single Board Computer, or the entire EUC.

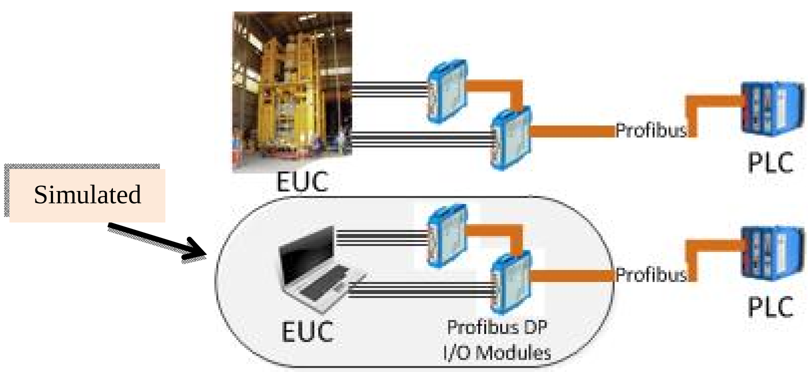

Figure 1: An example of a simulation model of HIL to show what is being simulated.

In figure 1, the controlling PLC is connected to a laptop which mimics the profibus I/O modules, the ECU, and the signals that are used by the EUC in real-time. The PLC reads and writes to the registers simulating the "real world". The purpose of a Hardware-In-the-Loop testing is to provide all of I/Os needed to fully test the control system's functionality. This also allows for the PLC to be thoroughly tested for its vulnerabilities, such as coding errors, logic errors, output to wrong registry, during a failure without damaging the actual equipment. The PLC may use other communication protocols such as LAN, Modbus, PROFINET, DirectNet, ControlNet etc.

HIL testing facilitates the quantity and quality of the testing by expanding the test plan and number of the tests for the EUC. Ideally, an embedded system would be tested against the real equipment, but most of the time the real equipment itself imposes limitations in terms of scope of the testing. For example, testing an Engine Control Unit (ECU) as real equipment can create the following limitations and potentially dangerous conditions for the ECU and test engineer:

- Testing at or beyond limitation of the engine's parameters (high RPM, high oil pressure, etc.)
- Testing the software program action at various failure conditions
- Damaging the equipment under control due to testing its functional limits
- Time delays due to delivery of the equipment that is to be tested
- Availability of personnel who are competent in testing the said equipment
- It is not cost effective to test all the connected equipment to test the PLC

Due to these reasons, software testing on actual equipment is not the most efficient way for verifying the control system. However, HIL testing can provide a solution to these issues during a project's lifecycle.

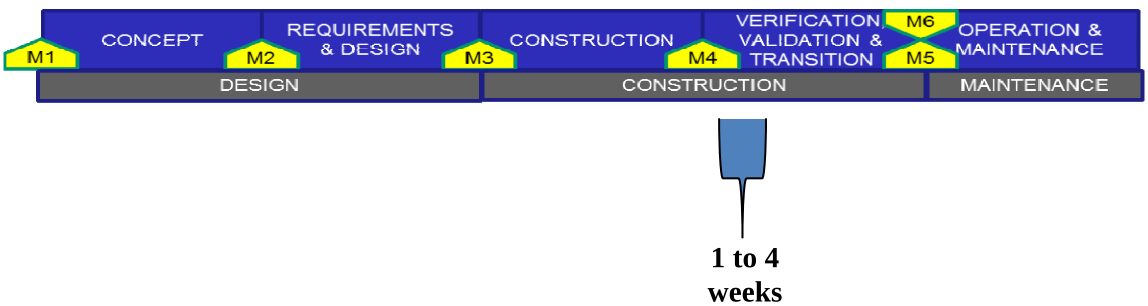

Figure 2: HIL testing occurs during the verification phase of the project

Since it is done before the commissioning, HIL testing minimizes schedule delays due to software changes during commissioning where sometimes missing functionality is discovered. Earlier and more thorough testing is beneficial for the Owner and the Shipyard to achieve the promised delivery date. The Owner of the asset has experience and knows what systems are most critical and also the highest risk. These systems are selected for HIL testing. The functions that fail to provide the expected results are typically corrected during the testing or sometime before commissioning and then re-tested.

The test requirement for the HIL testing is different for each control system. The test plan includes testing during normal state where the hardware and network equipment is working normally. The plan also includes degraded (partial failure, Network storm, EUC's failure) and failed states (where the software stops execution such as "Microsoft's blue screen of death"). The plan should also allow the Owner to add tests where they may have unique insight due to their experience. This allows the Owners to have inputs on any risk concerning the EUC and the software controlling it. This helps bring in unique aspects to each test plan, including the crucial risk factor. Testing the normal functionality proves that the function performs as expected where there are no additional inputs or any other unexpected failures. The programmers focus on the normal functionality. In addition to the normal functionality, the degraded and failed states are also tested. This allows the verification organization to identify as many defects as possible to minimize the probability where the equipment fails or behaves unexpectedly. The functions that fail to perform as expected, are fixed and re-tested later in the HIL test routine or possibly later during commissioning.

How does HIL help find defects in the given test environment?

A thorough test planning, analysis, and design process based on the functional description and safety analysis of the equipment is crucial for a productive and efficient testing. This analysis can also contain the expected actions of the equipment when the equipment is in normal, degraded and failed states. The main goal of the testing is to capture software defects and missing functionality during all the different states a function can have. Due to schedule demands, minimal testing is possible during commissioning and the number of personnel aboard to complete the construction and commissioning. This may increase the risk factor of software behaving unpredictably due to a nominal testing under the above constraints. HIL testing is aimed to reveal specific defects such as incorrect interrupt handling, I/O handling errors, real-time requirements, stack overflow, and memory allocation errors. The test plan may contain numerous test cases and these cases are considered as "traps" to capture the failures in a given test environment.

The number of test cases will depend on the possible consequence of a software failure and the nature of the target system, limited by factors such as project constraints, size of the program, number of test case etc. Organizing these test cases within the test plan and the test environment creates a more effective trap to capture the defects. By increasing the test cases for each function, it becomes easier to trap these defects with HIL testing. A typical/common HIL test program therefore consists of several types of tests:

- *Functional testing:* Verification of control system functions and modes during normal operation.
- *Failure mode testing:* Testing of control system detection and handling of failures and errors in signals, sensors, actuators and equipment.
- *Performance testing:* Testing of control system performance under different operational and environmental conditions. Performance testing requires high fidelity models and should be subject to careful analysis of model accuracy and sensitivity.
- *Integration testing:* Testing of integration between at least two control systems.

Different verification organizations may have different types of testing as part of their HIL test plan. Maximizing the test coverage and the frequency of discovering these faults, while adhering to project constraints, are two important goals to secure the value of performing HIL testing. As an example, we can explore the requirements for Dynamic Positioning (DP) control system as shown below.

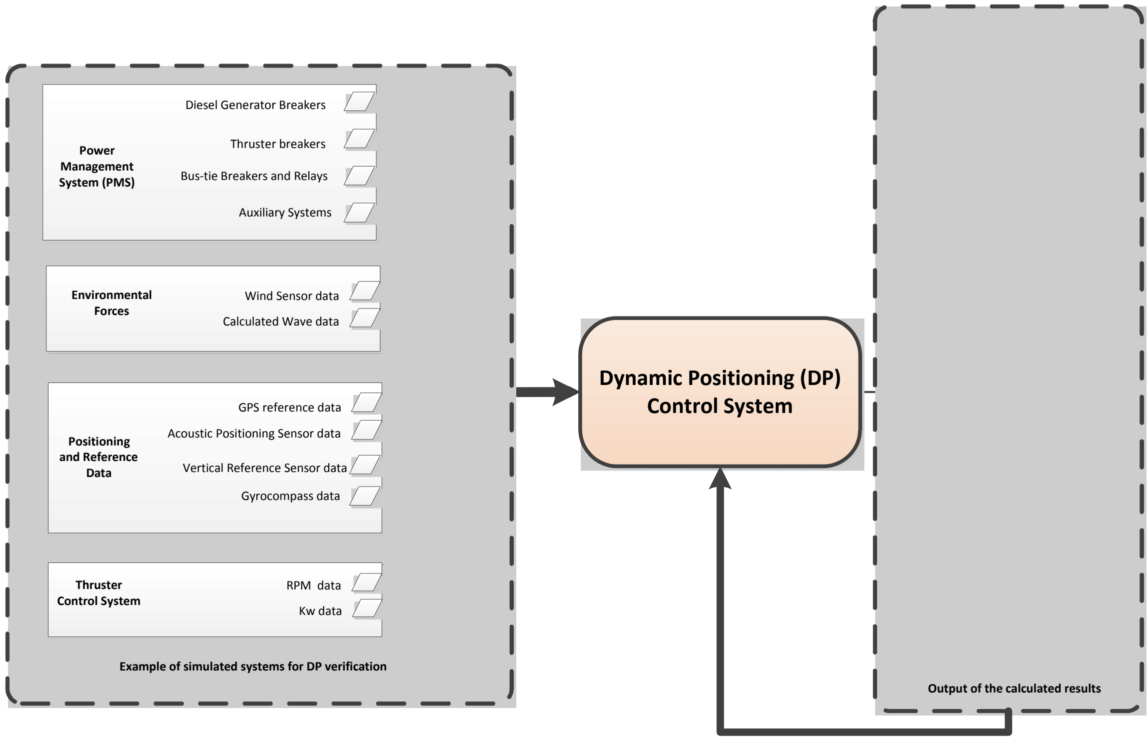

Figure 3: Simulation model for DP control system

The HIL simulation model for DP control system receives the I/O signals from the necessary input components on the left and the expected output signals on the right with the feedback to the DP control system. The initial conditions and the expected results are provided in the test plan by the verification organization. This is where the control system can be tested with upper and lower limits of the input data to predict how the control system might behave under different conditions.

Safety factor and development duration are typically equated to a cost measure. Specific conditions that warrant the use of HIL simulation may include:

- Safety, environmental and operation risk identification and reduction
- Tight development schedules
- High-burden-rate project
- Early process human factor development

In the above mentioned simulation model, HIL testing provides an efficient control and safe environment where test or application engineer can focus on the functionality of the DP control system. The discovered defects are noted, corrected, and re-tested. Besides the complex DP control system, in a recently delivered drillship, HIL testing was utilized on the entire drill floor equipment. There were approximately 32 different control systems involved as part of this HIL test plan. Seamless integration of different drill floor equipment is crucial during drilling operations. Performing HIL test on these systems has helped reveal defects and shortfalls of the control system where the defects were corrected before delivery of the software.

Why use HIL testing versus other testing methods?

The demand for robust, cost-efficient system level testing is increasing every day. There are several other methods that can be utilized for testing embedded systems. They may include Software-in-The-Loop, Closed Loop testing, Regression testing, Unit testing, and White Box testing. Although these methods are widely used, they all have their challenges and shortcomings when it comes to testing embedded systems. HIL testing requires the development of a real-time simulation that models parts of the embedded system and the interactions with other connected systems. As part of the scope, HIL testing is inclusive of other testing types such as Load testing, Sanity testing, Stress testing, Performance tests and Usability testing. HIL testing can be applied to a wide variety of systems, from relatively simple systems such as Gantry cranes, to more complex systems such as DP control system. Unlike some of the other testing methods, HIL is a non-intrusive testing method which does not require any proprietary information for the testing of the EUC. Testing using the HIL method following a thorough test plan lowers the risk of residual defects remaining in the equipment's software which should provide a more productive asset. The test itself is well documented so that it can be utilized for different projects and for training.

What are the benefits of using HIL testing method for verification?

In offshore and marine engineering, control systems and mechanical structures are generally designed in parallel. Traditionally, to thoroughly test the control systems, it was only possible after integration. As a result, errors are found that have to be solved during the commissioning, with the risks of personnel injuries, damaging equipment and delays in schedule. HIL simulation is gaining widespread attention among the Owners, Shipyards, and vendors who provide the control systems due to its benefits. Some benefits of using HIL testing include:

- Frontloading: Time and cost savings by transferring integration, optimization and verification tasks to earlier in the verification phases of the project.
- Consistent methodology and tool chain: Enables the re-usability of simulation models, parameters, measuring data and calculation results over the entire development process
- Open and scalable simulation solutions: To allow the flexible integration of customized simulation models and tools
- HIL testing facilitates the seamless integration of hundreds of software dependent control systems for any offshore asset
- HIL testing can help hardware and control system manufacturers improve their product designs.
- HIL testing is non-intrusive and does not rely upon accessing the computer source code or examining the equipment manufacturer's proprietary software
- Improving safety and the robustness of the functions that are being tested
- Helping the control system provide the expected results every time.
- Can be used for training purposes.
- Can be used for troubleshooting issues that may arise.

Conclusion

HIL simulation is an efficient, cost-saving, valuable technique that has been used for decades in the development and testing of complex embedded systems in the automotive and aerospace industries. HIL testing can be applied to a wide range of control systems by taking advantage of low-cost, high-powered computers and I/O devices for simulation of real time systems. Accurately designed and implemented HIL simulation can help develop systems faster and test them more thoroughly at a cost that may be significantly less than the cost of using traditional hard-wired testing methods. With compliance to Class requirements regarding safety and environmental factors, concerns like operational availability and performance are equally important to the vessel owner. By increasing the robustness of the tested systems, HIL testing has the additional benefit of potentially increasing the asset availability. In addition, experience has shown that unexpected multiple failures, often combined with some level of human error, may have undesirable consequences. HIL testing provides a solution to address these issues and to meet these requirements and demands of the Owners.

Acknowledgements

- Global Maritime
- Marine Cybernetics
- Samsung Heavy Industries (SHI) – Shipyard
- National Oilwell Varco (NOV)
- Kongsberg Maritime
- IEEE 610, IEEE 1012,

References

1. ABS Integrated Software Quality Management (ISQM) – 2012
2. System Verification (SV) - 2014
3. http://www.idsc.ethz.ch/Courses/embedded_control_systems/Exercises/Hardware-in-the-Loop.pdf
4. http://www.embedded.com/design/prototyping-and-development/4024865/Hardware-in-the-Loop-Simulation
5. http://www.ni.com/hil/ko/
6. http://www.marinecybernetics.com/
7. http://www.altera.com/literature/wp/wp-01208-hardware-in-the-loop.pdf
8. http://www.softwaretestinghelp.com/types-of-software-testing/