Using Simulations for Virtual Commissioning during Control Software Development

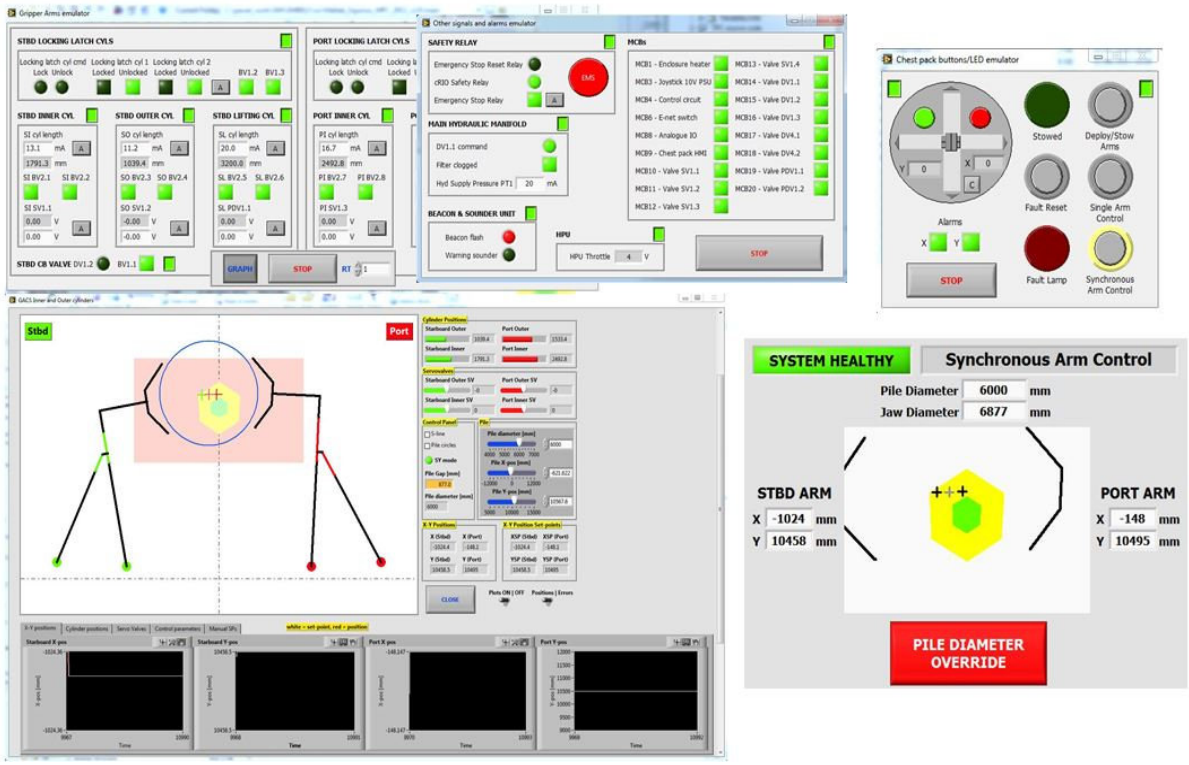
Andy Clegg, Managing Director, ISC Ltd
andy@isc-ltd.com, www.isc-ltd.com

- **Nonlinear Dynamic Simulation** - early in the design process simulations can be used to evaluate different design options (e.g. which sensor to use), assess likely performance (e.g. accuracy of control) and evaluate different features of the system (e.g. friction, aging effects). Typically this is confined to dynamic simulations where there may be fundamental uncertainties around performance and prove feasibility of designs. However, logic based functions can also be simulated at this stage should there be a need to de-risk some aspects of a design. ISC use both Matlab/Simulink and LabVIEW CD&SIM to build such simulations. Monte Carlo simulations can be useful to explore the performance over a random, wide range of operating scenarios.

    - An example of ISC using dynamic simulations early in the design process was on the *Turbine Access System* (see Case Study http://sine.ni.com/cs/app/doc/p/id/cs-14813). These simulations were used to assess the feasibility of achieving the required accuracy, by exploring different sensor types and combinations, and the most effective feedback control scheme. If the assessment had indicated that it needed something too big, too expensive or insufficient performance would result, then early decisions could have been made. These simulations whilst kept simple as possible to evaluated dynamic performance, were well justified since the predicted accuracies matched what was observed on the final system. ISC also used nonlinear dynamic simulations on the *Gripper Arm* (see http://sine.ni.com/cs/app/doc/p/id/cs-15650 for Case Study) hydraulic design, to assess the operation of counterbalance valves under different load scenarios and review their potential interaction with the position control loops.

- **Virtual Commissioning using Software-based Emulators ("Software in the Loop")** – during the development of the main control software application, a software emulation of the physical system can be built that replicates the operating modes and anything that requires testing (e.g. ability to inject faults as defined in the System Test Specification). Typically this may be mainly logic based to sufficiently replicate the full operation from start-up, operating modes and shut down. Only simple dynamics may be needed here, unless the performance remains critical to assess at this stage. In addition to testing the implemented software, such a software emulator can be used to review operational aspects with the end-users / customers, when linked to the user HMI screens and emulated button and lamps. This not only provides a valuable early review, but also helps build confidence with the final end-users. With LabVIEW's ability to run the same code on a PC or a real-time target, there is minimal overhead when moving between an entirely PC-based emulator and the real-time target for HIL testing, though the early definition of the software architecture to facilitate switching between either emulated IO or the real IO helps. The ability to extensively test software (either during development or for later revisions or bug fixes) without the need to have access to the physical system or the real-time target can be very helpful, as testing on the real system is often limited. This is more suited to logic-based systems and slower sample rate control systems as it is not fully deterministic.

- The *Gripper Arm Control System* developed by ISC made extensive use of a software-based emulator to demonstrate operating sequences to customers, thoroughly test the software prior to actual commissioning on the boat and investigate and test revisions once the system was in-service. The emulator provided the ability to inject most faults, and included simple dynamics for the hydraulic motions, graphical representation of the gripper arm positions and additional graphing for the motions.



Emulator developed for the Gripper Arm Control Software

- **Hardware in the Loop Testing** – with LabVIEW it can be relatively straightforward to move from an entirely PC-based emulator to Hardware in the Loop (HIL) testing where the application software is running on the real-time target (e.g. CompactRIO). There are a few options for this. Firstly, the system emulator can remain on the PC which interfaces to the real-time target using emulated IO (typically shared variables). This "**Processor in the Loop"** testing proves the real-time realisation, for example assessing CPU loading, specific FPGA features. Full HIL testing is where the sensors and actuators are emulated electrically, which could simply use the software emulator running on a second real-time processor with interfacing using real IO modules. Clearly this is a more rigorous test of the application software, but the value of which depends on the criticality of determinism and dynamic performance in the final system. As well as functional testing, HIL testing can be used to obtain initial tuning parameters for feedback controllers and test the controller tuning procedures to be used on the real system.

- For the *Gripper Arm Control System*, ISC developed the system emulator so that it could run on either a PC or the real-time target for HIL testing. A feature that proved useful was that the emulator could be engaged in parts, in that each of the six large hydraulic cylinders could be either simulated or connected to a real cylinder. Similarly most of the digital IO (circuit breakers, sounders) could also be either real or emulated depending what was connected. This allowed some early testing of the full system software at the factory when just a two of the cylinders were available, with the emulator showing how the gripper arms would be as if installed on the boat. This would not have been possible without this feature, and provided valuable reassurance ahead of the final, short commissioning window on the boat.

- **Operator Training Simulators** – training on a simulator allows operators to learn the normal operations and how fault conditions are indicated and handled within a safe simulation environment. Developing such a training simulator may require nothing more than re-using the software-based emulator with some well defined instruction.

Main benefits of using Simulations during Control Software Development:

- To de-risk uncertainties in the design by addressing design issues as early and as effectively as possible – to get the design right first time.
- Reduce commissioning time – this can be extremely important for large and/or high value systems where commissioning time is even more at a premium, particularly since software is always the last thing installed.
- Bug fixing and software revisions are easier to prove and test once a system is in-service.
- All of which give higher engineering quality, minimising expensive late design changes and faster development process.

The specific benefits do depend on the size, complexity, accessibility and performance of the system being developed. For small, simple, low performance systems where testing isn't onerous, the use of a simulation may be less important. For larger, more complex systems, simulations can be extremely helpful, with the additional cost of doing it is offset against the prospect of having to resolve costly design issues late in the development.

Finally, one vital aspect of using models and simulations is that their usefulness depends greatly on the quality of the models built. They should be neither too simple nor too complex. This comes from the experience of the people building the models. ISC has a great track record in building well validated, nonlinear dynamic models of mechanical, hydraulic, electrical, combustion, process and chemical systems for many industrial applications.