



AALBORG UNIVERSITY
DENMARK

Aalborg Universitet

A Cost-Effective Approach to Hardware-in-the-Loop Simulation

Pedersen, Mikkel Melters; Hansen, M. R.; Ballebye, M.

Published in:
Mechatronics

DOI (link to publication from Publisher):
[10.1007/978-3-642-23244-2_67](https://doi.org/10.1007/978-3-642-23244-2_67)

Creative Commons License
Unspecified

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Pedersen, M. M., Hansen, M. R., & Ballebye, M. (2012). A Cost-Effective Approach to Hardware-in-the-Loop Simulation. In *Mechatronics: Recent Technological and Scientific Advances, Part V* (pp. 553-560). Springer. https://doi.org/10.1007/978-3-642-23244-2_67

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- ? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- ? You may not further distribute the material or use it for any profit-making activity or commercial gain
- ? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

A Cost-effective Approach to Hardware-in-the-loop Simulation

M. M. Pedersen (a,c) *, M. R. Hansen (b), M. Ballebye (c)

(a) Aalborg University, Dept. of Mechanical and Manufacturing Engineering, DK-9220, Aalborg, Denmark, mmp@me.aau.dk.

(b) University of Agder, Department of Engineering, Grimstad, Norway.

(c) Højbjerg Maskinfabrik A/S, Oddervej 200, DK-8720, Denmark.

Abstract

This paper presents an approach for developing cost effective hardware-in-the-loop (HIL) simulation platforms for the use in controller software test and development. The approach is aimed at the many smaller manufacturers of e.g. mobile hydraulic machinery, which often do not have very advanced testing facilities at their disposal. A case study is presented where a HIL simulation platform is developed for the controller of a truck mounted loader crane. The total expenses in hardware and software is less than 10.000\$.

1. Introduction

The safety demands for machinery are ever increasing, as described in e.g. the new Machinery Directive [1] or EN13849 [2]. More and more electronic safety and control systems are therefore implemented in order to avoid potential danger, both from conventional use, but also from foreseeable misuse.

As an example from the loader crane industry, the new revision of EN12999 [3] requires the stability of the truck to be monitored and secured via the crane safety system, in order to avoid overturning of the truck. In the case of using the loader crane for personal lifts, the safety requirements are further increased due to the obvious risk to the person(s) being lifted. The safety system for an advanced loader crane as shown in Fig. 1 may include in the order of 50 IOs and up to some 20 CAN modules connected

to a main controller. The trend is towards more distributed IO systems communicating via CANbus networks, but also using sensors with integrated CANbus interface.

The complexity of the controller software increases in the same rate as the safety demands, and is further increased by a high level of product configurability. It thus becomes increasingly difficult to perform end-to-end tests for such systems, using a breakout board with switches, diodes and potentiometers.

In other industries with complex safety systems, e.g. the automotive, aerospace or offshore industries, hardware-in-the-loop simulation have long been the standard platform for testing controller software. In the crane industry and other similar industries, controller software testing have typically been performed using the real machine, though this is neither safe nor efficient.



Fig. 1: Truck with loader crane used for personal lift.

Earlier HIL simulation (HILS) platforms typically considered a single component or minor isolated systems for testing controller software with limited visual output. However, advances in computational power have propelled a rapid evolution of HILS platforms which now considers entire systems, such as heavy construction machinery or offshore structures and offer rich 3D animation output.

A HILS system for a hydraulic excavator with both audio and visual simulation output integrated in the operator cabin of a real excavator, was presented by Elton et al. [4]. The system included a remote hydraulic load emulator for a variable displacement pump, connected via Internet. The simulation are thus feed accurate (non-modeled) information from the pump.

Ghabcheloo and Hyvnen [5] discusses development of a motion control system for a hydraulic wheel loader in a HILS platform called GIMsim. It consists of 5 PCs communicating via LAN to accommodate animation, dynamics, tire contact, IO + CAN and simulation control. The simulation is developed in Matlab/Simulink, compiled and transfered to an xPCtarget for hard real-time operational speed.

Recently, Trydal [6] presented a very advanced proprietary HILS platform from National Oilwell Varco. It concerns an offshore drill-rig including the entire drilling process equipment and displays a 3D animation in a dome in order to completely immerse the operators. Instead of physical IOs, each controller are connected to a similar controller which simulates the IO functionality.

The general trend is to develop the simulation in Matlab/Simulink, use several PCs connected via LAN and use hardware IO connectivity from dSpace, National Instruments or others. Simulation input are given from real operator interfaces and results are presented using more and more realistic 3D animation. The animation software is generally developed specifically for the considered system using OpenGL.

In this paper, a cost effective approach to HIL simulation is demonstrated for a non time-critical application using only standard high-level engineering software. It should thus be possible for it to be implemented and maintained by engineers, without in-depth software development background.

2. Considered System

We consider a truck with an advanced folding loader crane attached to it, Fig. 1. The crane is equipped with a so-called flyjib, and has 6 degrees of freedom. In the tool point, a variety of accessories can be mounted, such as a rotator and grab or a personal lift basket and a winch.

The crane is mounted with a stabilizer system containing 4 extensible booms each with a stabilizer leg, and an additional fixed front stabilizer leg. In total we have 19 hydraulically actuated DOFs controlled by 12 directional valves and several shift valves.

The safety and control system, Fig. 3, of such a crane primarily covers a number of capacity limitation, stability monitoring and collision avoidance features, but also several secondary safety features, such as smart speed reduction and selective stopping of load-increasing motion. All safety features are supported by visual and audible warning signals.

The electronic hardware consists of a CAN network with a main controller, 3 I/O modules (flyjib, winch and stabilizers), a radio remote control, a slew sensor, 2 dual axis inclinometers, and up to 12 electro-hydraulically actuated directional control valves. Inputs and outputs are thus distributed to the corresponding function via the CAN network. The critical components are redundant, e.g. pressure transducers and inclinometers. Also the I/O modules and controller consist of two separate processors, each of which monitors the operation of the other, in order to achieve a high level of safety.

4. HIL Simulation

The developed HIL simulator consists of three subsections; 1) a simulation PC, 2) a hardware-gateway PC and 3) the real controller hardware, as shown in Fig. 2.

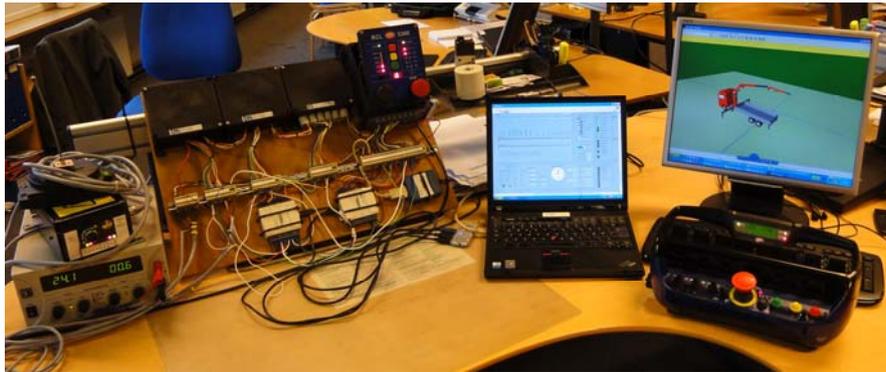


Fig. 2: Physical setup, from left: 1) controller hardware, 2) Gateway PC, 3) Simulation PC and operator interface.

The simulation is developed in Matlab using the Simulink 3D animation toolbox (which doesn't actually require Simulink) for visualization. The Matlab and Simulink programming environments are well known for their simple syntax and high productivity.

The hardware-gateway is developed in LabView using low-cost USB I/O and CAN devices from National Instruments for low level communication with the controller hardware. The link between the simulation running in Matlab and the hardware controlled by LabView is achieved over the local area network (LAN), using the IP/UDP protocol.

4.1 Co-simulation via LAN/UDP

LAN communication using the UDP protocol is an effective approach to achieve co-simulation with two different programs, e.g. Matlab and LabView or others. The programs can run on either two different PCs connected by LAN, or on a single PC sending data to itself - between programs. Using several PCs gives a performance advantage though and enables additional monitoring possibilities and distributed location of equipment.

The UDP protocol is typically used for streaming media over the Internet and other applications where it is preferable to drop packets rather than wait for them. In many applications, this is not a problem, if the same data are transmitted repeatedly. Sending data with UDP requires specification of the receiver IP address and port number, receiving requires only specification of the local port number and the size of the packet in bytes.

In this case study, the UDP protocol is used to communicate between the simulation running in Matlab on the Simulation PC and the Hardware Gateway PC running LabView, see Fig. 3. An UDP packet are sent both ways at every time-step in the simulation, all necessary data are collected and transferred in these packets.

The simulation model is thereby updated with the valve flow command sent from the controller to the virtual valves, and the state of all outputs from the controller and modules, e.g. to activate shift valves or perform engine control. In the other direction, the controller hardware is updated with the state of the virtual sensors in the simulation, e.g. hydraulic pressure, truck inclination, slew angle and estimated flow from the simulated valves, etc.

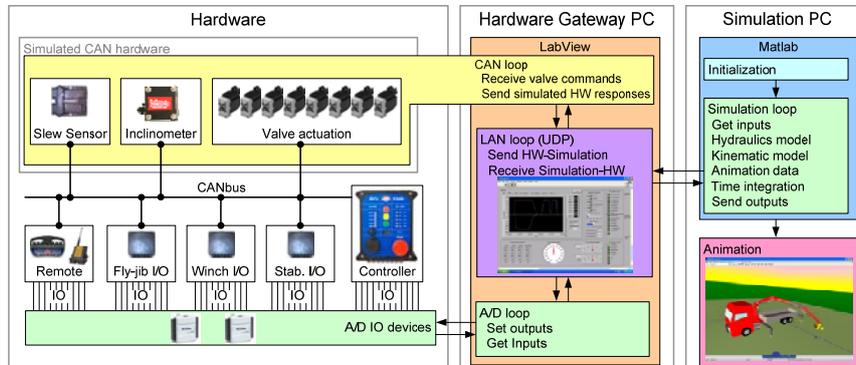


Fig. 3: Overview of the HIL simulation.

4.2 Hardware Gateway

Using a separate PC with LabView to facilitate communication between the controller hardware and the simulation has several advantages. Firstly, the price is low compared to dedicated systems. Secondly, the system is very flexible and scalable. The gateway PC can furthermore be used to control and monitor values in the HIL simulation while it runs. Additional filtering or noise can also be added to the signals here.

As shown in Fig. 3 the LabView program is divided in three separate loops; a LAN loop, CAN loop and A/D loop. Each loop covers communication between the gateway and LAN network, CAN network or analog/digital hardware. Separate loops in LabView runs in parallel for efficient execution and at different rates of 200Hz, 50Hz and 20Hz respectively. The hardware gateway additionally simulates a number of CAN devices, i.e. sensors and valve actuation.

4.3 Simulating CANopen Devices

The communication profile of most CAN devices such as sensors and valves is relatively simple and these can therefore be simulated quite easily. For a thorough overview of CAN and CANopen, see Pfeiffer et al. [7]. A first approach is to let LabView send CAN frames corresponding to the necessary transmit PDO(s) of the devices to the controller.

If needed, it is also possible to implement a limited object dictionary and SDO server in the LabView program containing the entries the controller (or other devices, e.g. a service terminal) will inquire on. Typically, to simulate e.g. a generic CAN I/O module, it will only be necessary to include in the order of 10 entries in the object dictionary.

The SDO protocol is straightforward for the expedited transfers (up to 4bytes), however if segmented or block transfers are required, this will take some more work, see CiA [8].

For the application at hand, strict real-time performance is not critical. Most of the CAN devices sample at 10-20Hz only and a small delay in the response of the simulation is acceptable.

4.4 Simulation Model

Since the simulation model must run in real time (or close to) in order to communicate with the controller, it is essential to limit the computational burden to what can be achieved with the available hardware.

Most dynamics of the system is omitted, which is acceptable since the controller mainly considers steady state values of the input. This is necessary in order to achieve near real-time performance in Matlab.

Referring to Fig. 3, the simulation is initiated by loading the crane parameters, such geometric and inertial properties, joint states and limits, hydraulic system properties, etc.. The main simulation loop is then started which is executed as fast as possible (around 200Hz) and updates all simulation parameters and controls the animation.

All required data is retrieved from the UDP packet; flow command to the valves, state of the outputs on the controller and modules, and necessary simulation control parameters which are set in the Hardware Gateway PC, i.e. mass of payload, mass of truck, crane setup, accessories mounted, etc.

Based on the flow commands and output states, the reference velocity of all DOFs are then calculated. Since several outputs controls shift valves, in order to use the directional control valves for multiple functions, the state of these leads to different modes of operation; i.e. crane mode, stabilizer mode, winch mode, etc.

Then a complete kinematic analysis is performed in order to determine position, orientation and velocity of all parts and the resulting tool point motion. The kinematic parameters are needed in order to drive the animation.

A simple kinetic analysis is also conducted, which determines the needed parameters; crane lifting moment, truck inclination due to crane load, hydraulic pressures in the main cylinder, flyjib cylinder and winch. Additionally, the ground projection of the total Center of Mass (COM), are calculated and displayed in the animation, in order to visualize the stability margin.

The only dynamics included is for a suspended payload, which is modeled as a 2 DOF pendulum. This gives the user the impression of operating a dynamic system and thus experience the consequences of abrupt motion.

The state of all virtual sensors are then determined. These include digital sensors, monitoring whether e.g. stabilizers are fully deployed, the jib is above horizontal, the extension system is fully retracted, etc. Several analog sensors are also simulated, e.g. pressure transducers for measuring pressure in selected cylinders, crane slew angle, truck inclination and directional control estimated valve flow, etc.

Time integration of the calculated velocities for all DOF is then performed using the trapezoidal rule and the animation is updated. All the simulated devices are modeled as ideal, i.e. they respond instantaneously and with perfect precision. In the Hardware Gateway, however, noise is added to selected signals, in order to test the controller filtering capabilities.

The necessary values are then assembled in a UDP packet and feed back to the controller via the Hardware gateway, and the simulation loop starts over.

4.5 3D Animation

Using the Simulink 3D Animation Toolbox, a Virtual Reality (VR) world can be manipulated from Matlab by translating, rotating or scaling the 3D parts at each time-step, to make it appear that the crane is moving. The VR world is defined using the VRML markup language, i.e. the relation between the parts and the surroundings.

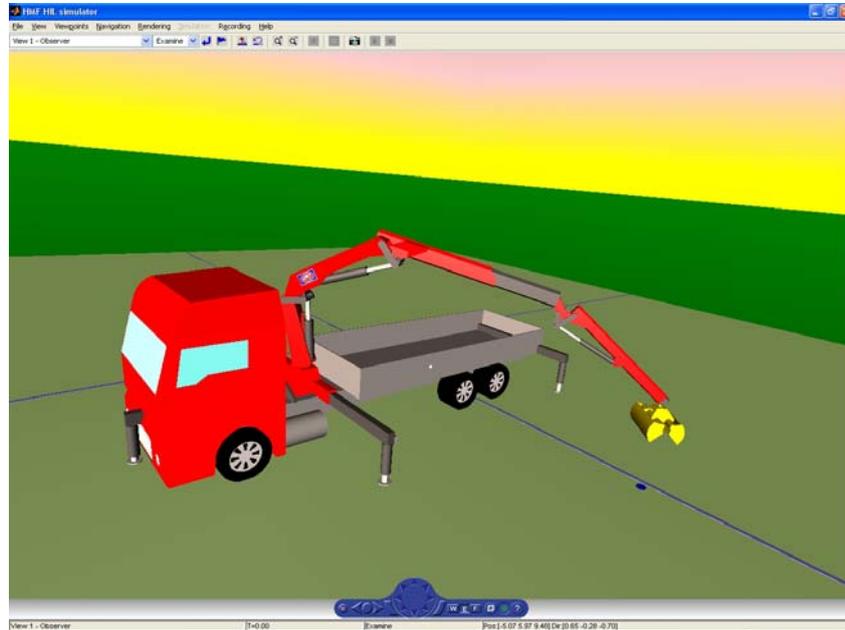


Fig. 4: Screenshot from the 3D animation.

Fig. 4 shows a screenshot from the implemented 3D animation. Some CAD programs can export parts to VRML format, otherwise, simple geometries, e.g. boxes and cylinders, can be directly defined in the VRML language.

Using the Simulink 3D Animation Toolbox makes it simple to create an advanced 3D environment for simulation output. Its flexible and scalable and based on a simple markup language, unlike the typical custom-made OpenGL applications used for HILS animation. Output from the animation can either be displayed directly or recorded for later use.

5. Conclusion

The HIL simulator described in this paper have proven to possess the necessary level of detail to test all the features of the controller concerned. Indeed, it's possible to create much more precise testing conditions, compared to using the real machine or a board with switches and potentiometers.

Especially when testing dangerous conditions, such as truck overturning or faults in the control system, simulation is preferred, since it is both expensive and time consuming to perform such test using the real machine.

The HIL simulator currently only comprise ad-hoc testing capability, however, it could be extended to include fully automated end-to-end testing of the controller, which will lead to additional savings on the testing expenses.

As a replacement for ad-hoc testing of controller software using the real machine or a breakout board, HIL simulation is a very effective tool. With the additional control over the testing conditions and increased monitoring possibilities, it is possible to create more repeatable and precise testing conditions. The effort required and risk of testing controller software can thus be significantly reduced.

It's difficult to determine the payback period on the investment of such HIL simulator, but a qualified estimate is 1-2 years. This is due to saved engineer-hours for testing, when comparing to testing using the real machine.

References

- [1] EU Machinery Directive, 2006/42/EG *European Union*, 2006.
- [2] EN ISO 13849-1:2006 Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design. *CEN*, 2006.
- [3] FprEN12999:2010 (Final Draft) Cranes - Loader cranes. *CEN*, 2010.
- [4] M.D. Elton, A.R. Enes and W.J. Book A Virtual Reality Operator Interface Station with Hydraulic Hardware-in-the-Loop Simulation for Prototyping Excavator Control Systems. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore, July 14-17, 2009.
- [5] R. Ghabcheloo and M. Hyvnen Modeling and motion control of an articulated-frame-steering hydraulic mobile machine. *17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, 2009.
- [6] S. Trydal HIL Simulator. Hardware In Loop Simulator med fokus på integrerte prosesser. (Norwegian) *NODE seminar*, Grimstad, Norway, 29 September, 2010. <http://www.nodeproject.no/index.php?articleid=393>
- [7] O. Pfeiffer, A. Ayre and C. Keydel *Embedded Networking with CAN and CANopen* Copperhill Media, MA, USA, 2003.
- [8] CAN in Automation (CiA), e.V. CANopen, Application Layer and Communication Profile. *CiA 301*, 2002.