

Software-in-the-Loop Simulation Speeds up Machine Control Tests

Dr.-Ing. Dirk Wehner, Hydrive Engineering GmbH, Dresden, Germany

Kurzfassung

Die steigende Komplexität und der Umfang von elektronischen Steuerungen macht die Inbetriebnahme und den Test von Maschinen zu einem immer anspruchsvolleren Schritt im Entwicklungsprozess. Dabei spielt die Automatisierung der Softwaretests eine besondere Rolle. Viele Methoden sind dafür bereits entwickelt und im Einsatz. Die ITI GmbH bietet mit ihrem Simulationstool „SimulationX“ ein Software-in-the-Loop Testverfahren an, mit dem es möglich ist, Original-Maschinensoftware in das Simulationsmodell einer Maschine zu implementieren. Der Beitrag beschreibt dieses Verfahren und diskutiert verschiedene Aspekte der Anwendung. Am Beispiel einer Aufgabe aus der Ventilentwicklung werden Vorzüge wie frühere Tests und minimierter Einrichtungsaufwand herausgearbeitet.

Abstract

The validation of modern machinery becomes due to the increasing complexity of machine software a more challenging part of the development process. Different types of automated software testing procedures are established. ITI offers a software-in-the-loop testing method where the original softwarecode can be implemented within a simulation modell. The paper describes that treatment and discusses different aspects of its usage. A valve development task is used as an example to illustrate advantages like early and realistic tests with a minimum of preperation effort.

Introduction

More and more functionalities of todays machinery is determined by it's control software. Customers expect new features, increased comfort or high energy efficiency which is often realized by sophisticated software features. But otherwise they expect the same robustness and long-life cycle they know from former machine generations.

OEM's well know, that the more complex a software is, the more effort has to be spend for testing and validation. That leads to multiple challenges for the development process. Conventional testing methods are allways starting at the end of the development process when all other parts of the machine are started-up. Manual step by step tests are on the one hand time consuming. On the other hand they expose software bugs at a time, were it is often to late for a systematic software review.

For this reason a lot of OEM's are using software testing methods that take effect earlier in the development process. In the past years a lot of different methods, tools and strategies were developed in order to meet the mentioned challenges. The fundamental idea of the most approaches is the virtualisation of the machine or parts of the machine. Simulation tools like SimulationX became more and more part of such testing methods.

Software Testing Methods

The idea of virtualisation can take place in different parts of the machine. Figure 1 illustrates a matrix of software testing methods. On the one hand the machine itself can be integrated in a virtual or in a real manner. But even the software can be treated in a virtual or in a real way. Depending on the actual development tasks one of those four ways can be used.

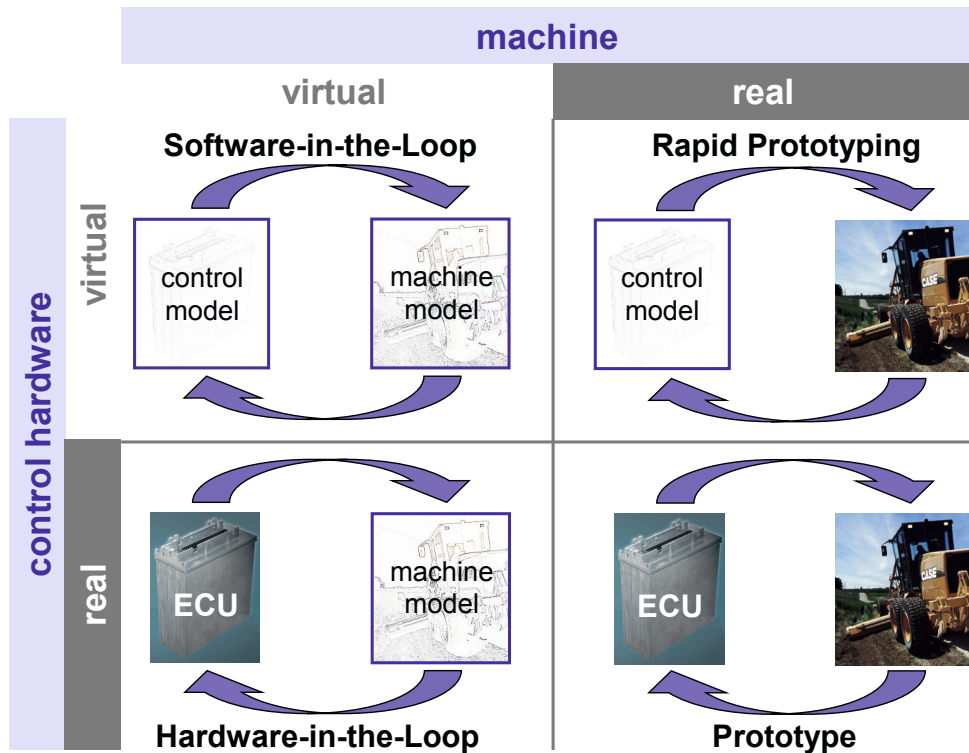


Figure 1: Different Softwaretesting Methods

Prototype

A real prototype includes the machine as well as the control hardware with the original software project. Tests with such a configuration are only possible in the final phase of the development process. It is characterized by a high hardware effort. Testing is often done manually by a test engineer on the proving ground which costs a lot of time. One

fundamental advantage is that all tests can be executed under realistic conditions. The mentioned virtualisation enables testing activities in earlier development phases.

Hardware-in-the-Loop

The hardware-in-the-loop method uses the real control hardware interacting with a simulation model of the machine. The hardware effort is reduced because the machine is virtualized. Tests can be performed without a prototype machine. That enables the software validation to start right after setting up the control hardware in a so called HIL-Teststand. Figure 2 shows an example of such a HIL-Teststand created by Hydrive Engineering. One of the main advantages is that the software code can be tested on the real target system like it will run on the real machine. That enables for example the testing of electrical failiures and the reaction of the software. A special test control software allows to perform automated software tests. That saves testing time an increases the repeatability of the tests.



Figure 2: Hardware-in-the-Loop Teststand for a Mobile Machine

Rapid Prototyping

Rapid Prototyping is often used to validate new control strategies under realistic conditions. A real machine is temporary equipped with a rapid prototyping control hardware. The effort for implementing the control algorithms is much less then programming the software for the final control target. But it is allways necessary to have

a prototype machine available and to install the rapid prototyping hardware on the machine.

Software-in-the-Loop

In early development phases where no hardware is available, validation of machine concepts or software parts can be performed by software-in-the-loop methods. Both the machine as well as the software are only available in a virtual manner. It can help to evaluate machine concept or to optimize control strategies. Like CAD offers a detailed view in several design aspects, simulation tools can describe the expected functionalities and physical relationships. The main advantage is that the engineers get a detailed understanding of their machine design in the early phases of development process. And with the software-in-the-loop they can even study the correlation of their software concepts with the physics of the machines.

Software-in-the-Loop with SimulationX

SimulationX is a multi-domain-simulation software for analyzing or optimization of a great variety of technical systems or components. Specific libraries can be used to develop more or less detailed models of machines or machine parts. These models representing virtual prototypes. Engineers can analyze system behaviour, study parameter influences and thereby optimize the system design. Such machine models typically do not contain software logic. But it is possible and often necessary to extend the model by logical algorithms, which later will be part of the machine software.

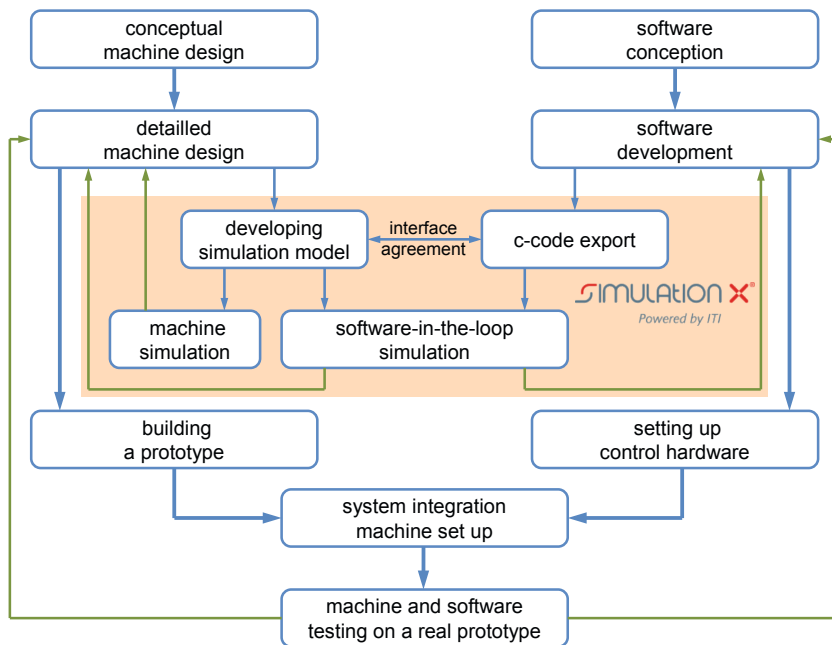


Figure 3: Integration of Software-in-the-Loop Testing in Machine Development

One possible way is to describe it with the internal libraries of SimulationX. That is an easy and useful way. Depending on the required functional range, the code that needs to be modeled, can be much. But that "SimulationX software code" can not be used in final control hardware. So engineers have to develop parts of the software two times, once for the simulation model and once for the real target system.

In order to avoid two times programming, it might be helpful to integrate original software code within the simulation model. SimulationX provides with the external object interface an opportunity for testing C-code within its simulation environment. Figure 3 illustrates the basic idea of that principle. On the one hand a machine model is derived from the actual machine design. This virtual prototype does not necessarily need to describe the entire machine behaviour. The software control part is still missing within the model. Software engineers can start in parallel with the software conception and setting up a first release of the control software. The software developer and the simulation engineer agree on the interfaces between the machine software and the machine model. The software developer exports the current version of his machine software as an encoded DLL-file. The simulation engineer incorporates this file into the machine's simulation model, creating a connection between software and machine. Now it is possible to run software-in-the-loop simulation tests for analyzing the machine behaviour while interacting with parts of the future software.

The following advantages can be utilized by using this technology:

- the methodology can be used in early development phases
- it helps to ensure design- as well as software-concepts
- engineers of the different departments are forced to work together as early as possible
- the simulation model of the machine can be used for multiple purposes
- first software tests can be performed with no hardware effort
- once established, this technique will allow software revisions to be tested quickly and reliably

Development of Valve Control Strategies using Software-in-the-Loop

Hydrive Engineering used this Software-in-the-Loop method in a valve development project. Two development tasks were closely attended by means of simulation. On the one hand a valve needed to be developed for a special purpose. A simulation model of the valve were used to evaluate the design draft as well as to optimize critical design elements. On the other hand a new embedded control hardware were used. It shall be equipped with a special software that contains new functionalities. The main challenge here were the strong interactions between the valve and the software. Practical tests were only possible in limited time spans.

A detailed model of the valve were available from the origin development process. The control hardware requires software code in C/C++. While setting up the software project, first relevant control algorithms were developed and implemented. Figure 4 shows the simulation model of the valve with the integrated software part. The part of the valve physics can be found in section A. Section B contains the external type connected by the

external object interface to the dll-file of the exported software part. The external type is set up configured with the type designer. Input and output variables can be mapped on several connetions. The configuration dialog of the type contains parameters of the software part. All necessary software inputs are placed in section C.

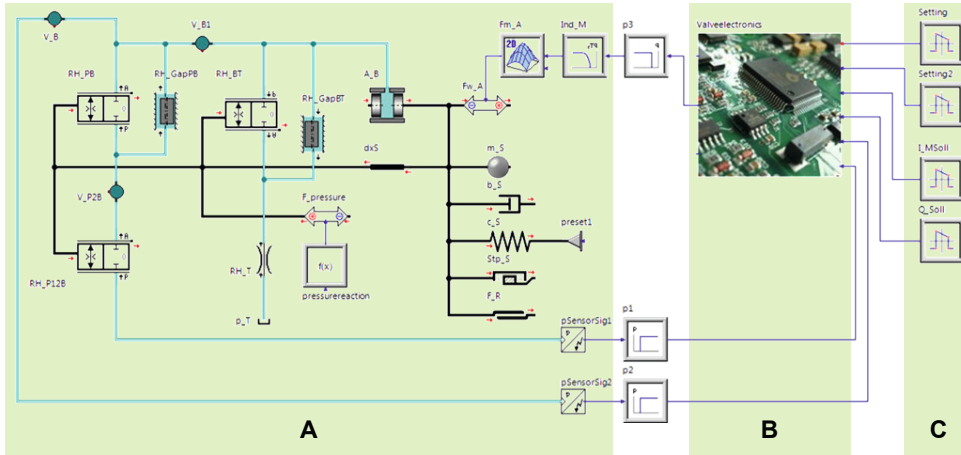


Figure 4: Simulation Model of the Valve with Integrated Valve Control Software

The effort for setting up the external object interface is small. The definition of the external type is well explained in ITI documentation. Once the interface established, SimulationX can be used like engineers know it. A software release can be integrated only by replacing the dll-file in the specified folder. Tests can be repeated and automated by definition of a parameter analysis.

Conclusion

Software-in-the-loop testing can be a helpful measure in early design phases. ITI offers with the external object interface an easy and beneficial way for integration of c-code. An existing simulation model of a machine can simply be extended by original software parts. Users benefit from a minimum of installation effort. The do not need to code the software twice. Software engineers are have the chance to get a very early feedback about their software.

References

- [1] Neubert,Th., Weber, J., Lautner, E. Den gestiegenen Anforderungen begegnen – Softwareentwicklung für elektrohydraulische Antriebssysteme in mobilen Arbeitsmaschinen, In: O+P Journal, Vol. 6/2009, pp. 244-249, 2009.
- [2] ITI GmbH, Einbindung von externem C-Code in SimulationX Modellen, Dokumentation, 2014