WHITE PAPER

Virtual Commissioning with Model-Based Design

How to implement virtual commissioning in your development process

MathWorks®

With virtual commissioning, you use a simulation model of your manufacturing or industrial automation plant to test and validate system changes and upgrades before implementing them on actual equipment.

- The use of virtual commissioning is increasing due to:
- More sophisticated software used in production equipment
- The need to operate equipment continuously and globally
- A lack of testing capabilities to verify the functional behavior of the machine
- The desire to eliminate errors early in the design process

Each of these factors increases the overall cost for commissioning due to project delays and the potential damage to your business reputation.

Instead of testing complex software after parts of the machinery or the complete production line is assembled, engineers use a simulation model of the machine—a virtual machine. With simulation, the interaction between mechanics, machine software, and the product can be tested, optimized, and verified in different scenarios, even if the physical machine is not yet available.

Does Virtual Commissioning Pay Off?

A virtual machine lets you run most of the time-consuming and often expensive tests first in simulation, reducing the time it takes for "real" commissioning of the physical plant. In addition, you can test earlier and in parallel with other activities such as assembly of the mechanical and electrical parts.

The main objection to introducing virtual commissioning is the effort required to develop a model of the machine with sufficient accuracy, as well as the corresponding know-how needed by development teams. So it's important for companies evaluating virtual commissioning to compare the effort with the expected benefits:

- What is the return on investment (ROI)?
  - How much does traditional commissioning (without virtual commissioning) cost?
    Consider staff needed onsite, travel costs, waste of material and energy, and any risks of penalties in case of delay.
  - What does the investment in virtual commissioning look like?
    Consider creating models, building knowledge on the team, and licensing simulation software.
- What are the technical requirements for the model?
  - Level of abstraction: Incorporate simple control signals, dynamics of mechanics, sensor signals, etc.
  - Level of visualization: Consider whether visualization is needed. If so, you may want to create graphs of values over time or even 2D or 3D animations.
  - Real-time capabilities: Does the model need to run in real time?
  - Communication interfaces: Connect the software/PLC to the virtual machine.

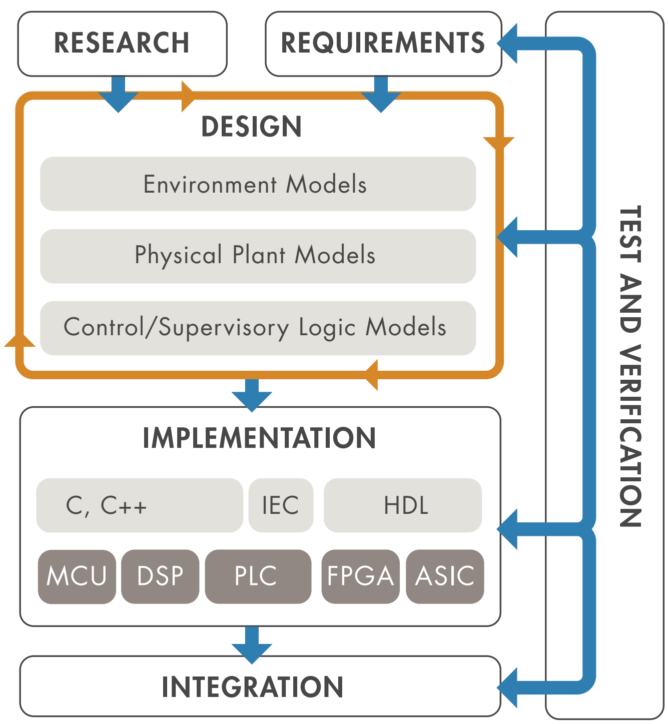
Companies that successfully implement virtual commissioning have two things in common:

- Realistic expectations about the model, the required accuracy, and which requirements it can fulfill.
- A phased introduction of new processes. Steps that provide the best ROI are implemented first and deliver first results. Thus, initial investments pay off early and facilitate later phases.

However, a model of the machine enables more than virtual commissioning. It can also play a central role in the entire development of the machine and the machine software. Furthermore, you can use the model throughout the lifetime of a machine in the form of a digital twin, such as for monitoring machinery or performing predictive maintenance. Taking this into account, the effort and investment in modeling is justified more easily and pays off in multiple ways.

Model-Based Design: Beyond Classic Virtual Commissioning

Even if virtual commissioning is an effective way of addressing the challenges of real commissioning, it is not the most efficient way of using models. In contrast to the machine builder and production equipment industry, companies in other fields like automotive and aerospace are already using models from the first day of development. They create models of the plant (mechanics, electrical drives, hydraulics, etc.) as well as the control software. This approach enables closed-loop system simulation from the very beginning of the product life cycle and allows simultaneous development of mechanics, electronics, and software. Once the software algorithms are completed in design, they are tested against the plant model. The production code for implementation on a PLC or other automation hardware is automatically generated. This workflow—Model-Based Design—shortens development time and increases the quality of the product.



Development workflow with Model-Based Design. The approach spans the entire development process including the commissioning phase.

Modeling and Desktop Simulation

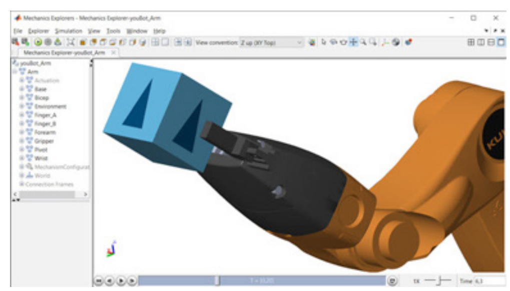
The machine model and controls model are designed based on requirements, measured data, CAD models, and legacy machine software. The models are refined and then combined into a complete system model. Tools such as Simulink® not only allow you to create models for the machine or system with prebuilt elements, but also support the design of control logic and algorithms, as well as real-time code generation for PLCs.

Key Capabilities

- Design of controls model and plant models of the physical system, including CAD import
- Prototyping of new functionality in combination with legacy machine software
- Automated system tests
- Parameter optimization (such as software, mechanics, hydraulics)
- Automatic code generation from models (IEC 61131-3 Structured Text and Ladder Diagram, C/C++, HDL)

Fast Modeling Using CAD Import

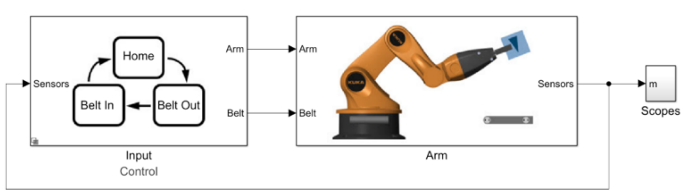
CAD models provide a comprehensive mechanical description of a machine. With *Simscape Multibody™*, CAD assemblies, including masses, inertias, joints, constraints, and 3D geometry, can be imported to obtain the machine model. After the import, the user defines which joints are active (actuator driven), passive, and/or sensed (provide position feedback). Depending on the requirements, electrical drive trains with motors and inverters are added.



System dynamics visualized in 3D.

Benefits

- Simulate mechanical, electrical, and control systems in the same software environment, enabling complete system design optimization
- Use system-level simulation to evaluate and optimize the dynamic interaction of the electrical, mechanical, and control systems and its impact on the embedded software
- Find design errors early in the design process
- Perform costly and safety-critical tests in simulation
- Avoid manual coding errors by automatically generating code
- Reuse models in future projects



Closed-loop system model of a robot arm and its control logic in Simscape™ and Stateflow®.

Hardware-in-the-Loop and Real-Time Tests

A virtual version of the machine is created through automatic code generation from the machine simulation model. This virtual machine runs on a real-time hardware—*a hardware-in-the-loop* simulator—which is connected to the industrial controller over an industrial fieldbus for testing, refining, and optimizing the machine software.

Key Capabilities

- Emulate the behavior of the physical system (plant model) in real time
- Design and test hardware-independent functionality
- Debug real-time algorithms directly from Simulink (using "External Mode")
- Connect the virtual machine to the PLC over an industrial fieldbus

Benefits

- Test and verify the functionality of machine software before the physical machine is available
- Debug the machine software in operation directly from the model
- Find integration errors earlier in the design process by using real-time simulation for testing and verification
- Simulink Real-Time™ and real-time hardware from Speedgoat provide an optimized solution for hardware-in-the-loop simulation. *Speedgoat* hardware provides interfaces to common industrial fieldbus protocols:

Virtual Commissioning Under Real-Time Conditions

To obtain feedback about the functionality and performance of the machine control software on the final hardware, it is helpful to run the model under real-time conditions to test the algorithms and the fieldbus communication to the PLC.

Simulink Real-Time in conjunction with a real-time hardware system from Speedgoat enables real-time execution of the machine model and supports common industrial fieldbus protocols used in the automation industry.

Protocol	I/O Modules
PROFIBUS®	*IO641* (master), *IO642* (slave)
PROFINET®	*IO751* (controller), *IO752* (device)
EtherCAT®	Via GbE ports on all *real-time target machines*; *IO710*, *IO712*, and *IO713* (master); *IO750* (slave)
Modbus TCP®	*IO753* (client), *IO754* (server)
Modbus RTU®	External gateway
EtherNet/IP™	*IO755* (scanner), *IO756* (adapter)
POWERLINK	*IO758* (controlled node)

Code Generation for PLC Platforms

The machine software can be implemented on several system architectures by automatically generating the code from tested and validated models. Depending on the selected hardware platform, *C/C++, IEC 61131-3 (Structured Text or Ladder Diagram),* or *HDL* code can be generated. Automatic code generation is available for all common PLC and industrial PC platforms. MathWorks cooperates with several hardware partners to ensure compatibility with their integrated development environments (IDEs).

The support of different implementation languages and PLC platforms enables engineers to design and test the machine software independently from the hardware platform. This approach is especially helpful when multiple PLC platforms are used or when the final decision for the hardware has not yet been made.

Vendor	IDE	IEC 61131-3	C/C++	Connections Partner
3S - Smart Software Solutions	CODESYS®	✓		✓
ABB / B&R Industrial Automation	Automation Studio™	✓	✓	✓
Bachmann Electronic	SolutionCenter	✓	✓	✓
Beckhoff Automation	TwinCAT®	✓	✓	✓
Bosch Rexroth	IndraWorks	✓	✓	✓
Mitsubishi Electric	CW Workbench		✓	✓
Ingeteam	Ingesys IC3		✓	✓
Omron	Sysmac® Studio	✓		✓
Phoenix Contact	PC WORX™	✓	✓	✓
Rockwell Automation	RSLogix™/Studio 5000	✓		✓
Schneider Electric	Unity Pro	✓		
Siemens	TIA Portal/STEP® 7	✓	✓	✓

Code generation support by PLC platform. MathWorks supports code generation for PLCs and industrial PCs for all common PLC and industrial PC platforms and works closely with PLC vendors through the Connections Program.

Key Capabilities

- Generation of C/C++, IEC 61131-3, or HDL code
- Integration of automatically generated code into the PLC software through the vendor's IDE
- Online debugging from Simulink and Stateflow using "External Mode"

Benefits

- Perform hardware-independent design and test of PLC software
- Eliminate time and effort spent on manual coding
- Reduce coding errors by automatically generating code

Digital Twin

Although "classic" Model-Based Design and virtual commissioning focus on the design phase of the machine or production plant, simulation models are more and more often used as the basis for "*digital twins*." The digital twin is a virtual representation of the equipment that runs in parallel to the physical asset in operation and is supplied with measured data from the physical system. Digital twins are typically integrated into an edge device or into the IT/OT infrastructure of the production site.

Typical Applications

- Model-based health monitoring and predictive maintenance
- Reproduction of errors from field data
- Operator training on new systems

Benefits

- Reuse simulation models over the entire lifespan of the equipment
- Integrate into the IT/OT infrastructure at the production site and *IoT platforms*
- Continuously update the model based on measured in production data

Conclusion

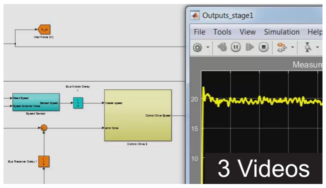
While virtual commissioning provides machine builders and industrial equipment builders with a valuable means for testing their embedded software early in the design process—before the physical machine or prototype is available—it covers only a limited segment of the entire design process or even lifespan of the equipment.

Model-Based Design enables users to benefit from their simulation models over the entire design process, including automatic testing and code generation of software that later runs on a PLC, industrial PC, or embedded controller.

Digital twins based on the simulation models developed for the design phase go even one step further and serve as the basis for health monitoring, predictive maintenance, and many other valuable applications for in-production use.

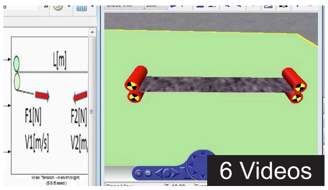
Online Resources

Webinars and Videos



Virtual Commissioning with Simulink

See how you can test your software in early project stages by using virtual commissioning with Model-Based Design.



Virtual Commissioning of Production Machines

Modern production machines contain increasingly complex real-time software, typically running on a PLC, PAC, or industrial PC. To test and verify the functionality before the physical machine is available, virtual commissioning based on modeling, simulation, and code generation has become a well-proven design workflow.

Model-Based Design in Industrial Automation

At Tetra Pak, using Model-Based Design has enabled the new flagship packaging equipment lines to run at double the capacity of the previously fastest lines.

Verification and Virtual Commissioning of Configurable Handling Systems

Manufacturing companies that aim to increase production rates and to reduce costs need a method to virtually test system behavior. Reishauer AG, a producer of high-precision gear grinding machines, addresses this challenge by using Simscape to unite CAD designs with mechatronic simulations.

Customer Statements

Metso Develops Controller for Energy-Saving Digital Hydraulic System for Papermaking Equipment Using Model-Based Design

> "Using Model-Based Design with MATLAB and Simulink, we achieved multiple goals simultaneously. We developed a sophisticated controller for digital hydraulics that is more reliable, accurate, and efficient than previous systems, and we accelerated development, which gives us a competitive advantage."
>
> — *Kari Leminen, Metso*

Papermaking equipment. The machine's calender is controlled by a digital hydraulic system.

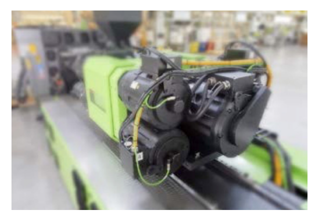
FMTC Designs and Optimizes a Hybrid Hydrostatic Drivetrain with Model-Based Design

> "Model-Based Design supports a systematic approach to the design of drivetrains and other complex mechatronics systems. Detailed analysis of design alternatives based on the simulation of dynamic physical models and optimal controllers enabled us to make informed decisions early in the design phase."
>
> — *Kristof Berx, FMTC*

The hybrid hydrostatic drivetrain setup.

ENGEL Speeds Development of Injection Molding Machine Controllers

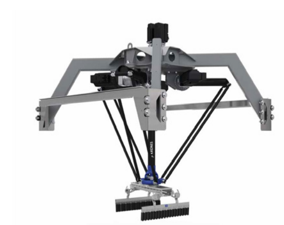
> "Model-Based Design reduces the time needed to produce quality control algorithms. Simulations help us understand the system, and code generation enables us to maintain a single source for the design. The results are faster development and higher-quality systems."
>
> — *Hannes Bernhard, ENGEL*



The injection unit, which is driven by four synchronized drives.

Krones Develops Package-Handling Robot Digital Twin

> "Simulations of the digital twin in Simulink enabled us to obtain data and insights that would be either impossible to get via hardware tests or simply too costly and time-consuming. Visualizing forces and moments helped us to understand the effects of individual components on a highly dynamic robot."
>
> — *Benedikt Böttcher, Krones*



The Krones Robobox T-GM package-handling robot.