



Verifying Control Logic on Soft PLCs Using Co-Simulation

Feng He

Application Engineer

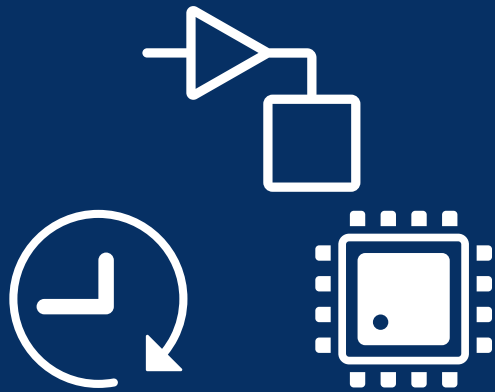
Martin Förster

Application Engineer

Rareş Curatu

Industry Manager, EMEA

Agenda

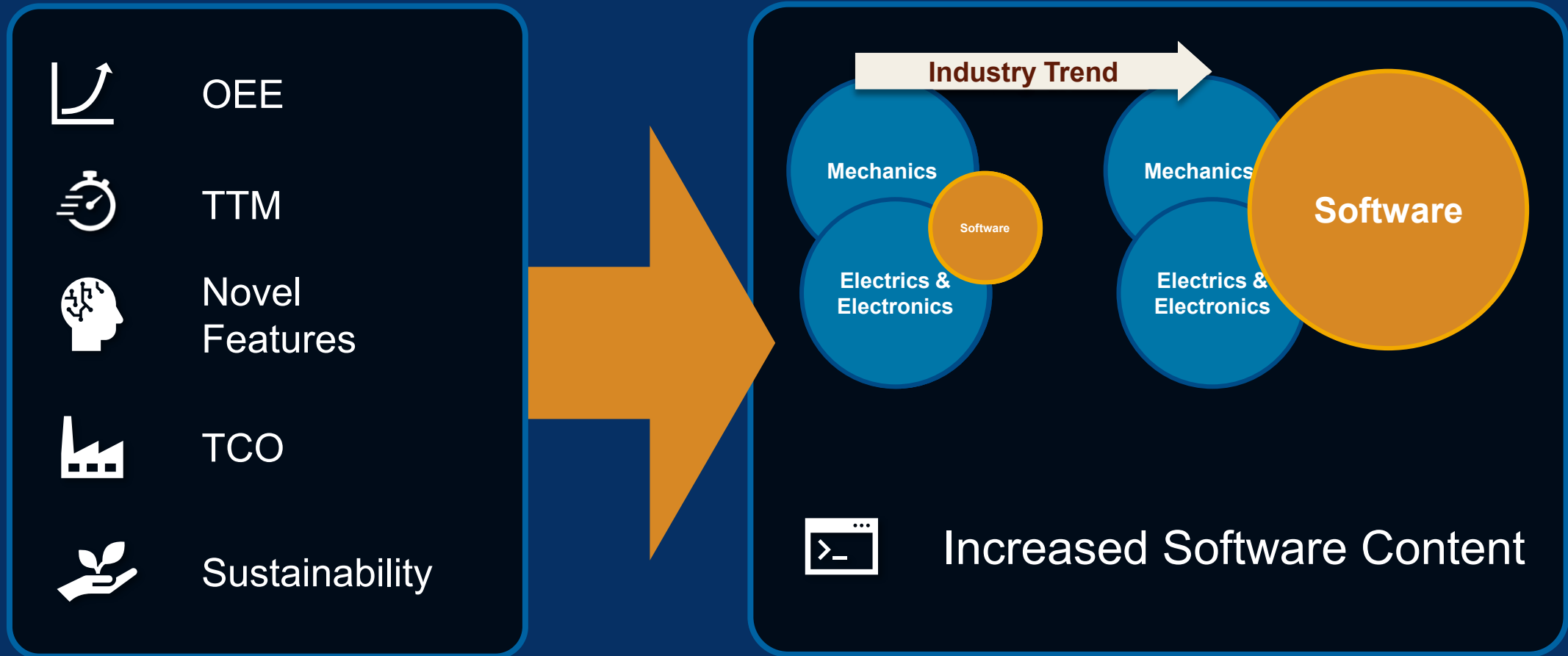


3 Engineering Imperatives and the Role of Co-Simulation



Verifying Control Logic on Soft PLCs Using Co-Simulation

Increasing Software Content



OEE



TTM



Novel Features



TCO



Sustainability

Industry Trend

Mechanics

Electrics & Electronics

Software

Mechanics

Electrics & Electronics

Software



Increased Software Content

Increased Complexity & Opportunities



Increased Software Content



Challenges

- Increased Complexity
- Increased Engineering & Development effort



Opportunities

- Design improvement
- New services and products



Software

Software Development with Model-Based Design: **3 Engineering Imperatives**

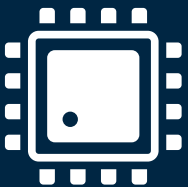
3 Engineering Imperatives



Mastering Mechatronic Complexity and Variant Management



Continuous Simulation & AI Validation



Architectural Flexibility & Algorithm Portability

3 Engineering Imperatives

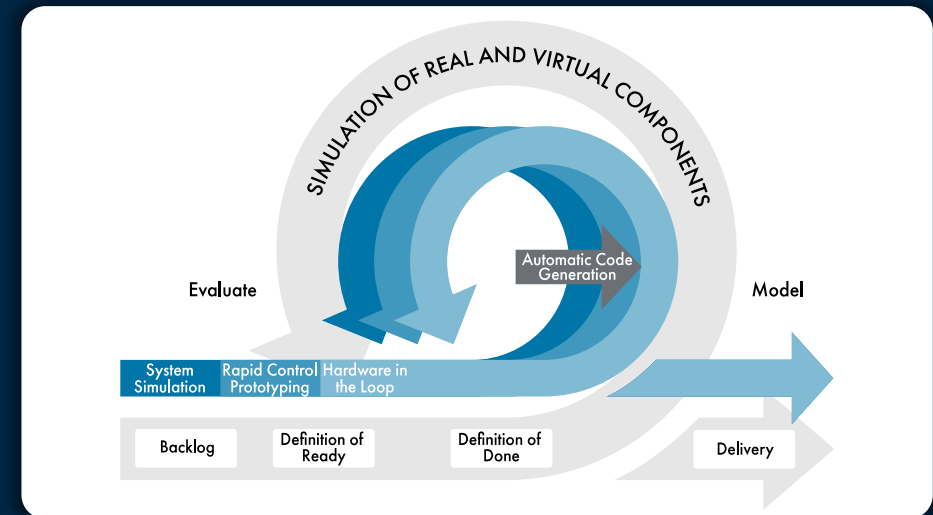
Mastering Mechatronic Complexity and Variant Management

Continuous Simulation & AI Validation

The Bottleneck: Testing complex logic and non-deterministic AI on physical iron causes project delays.

The Shift: Using high-fidelity plant simulation as the automated testing environment for CI/CD pipelines.

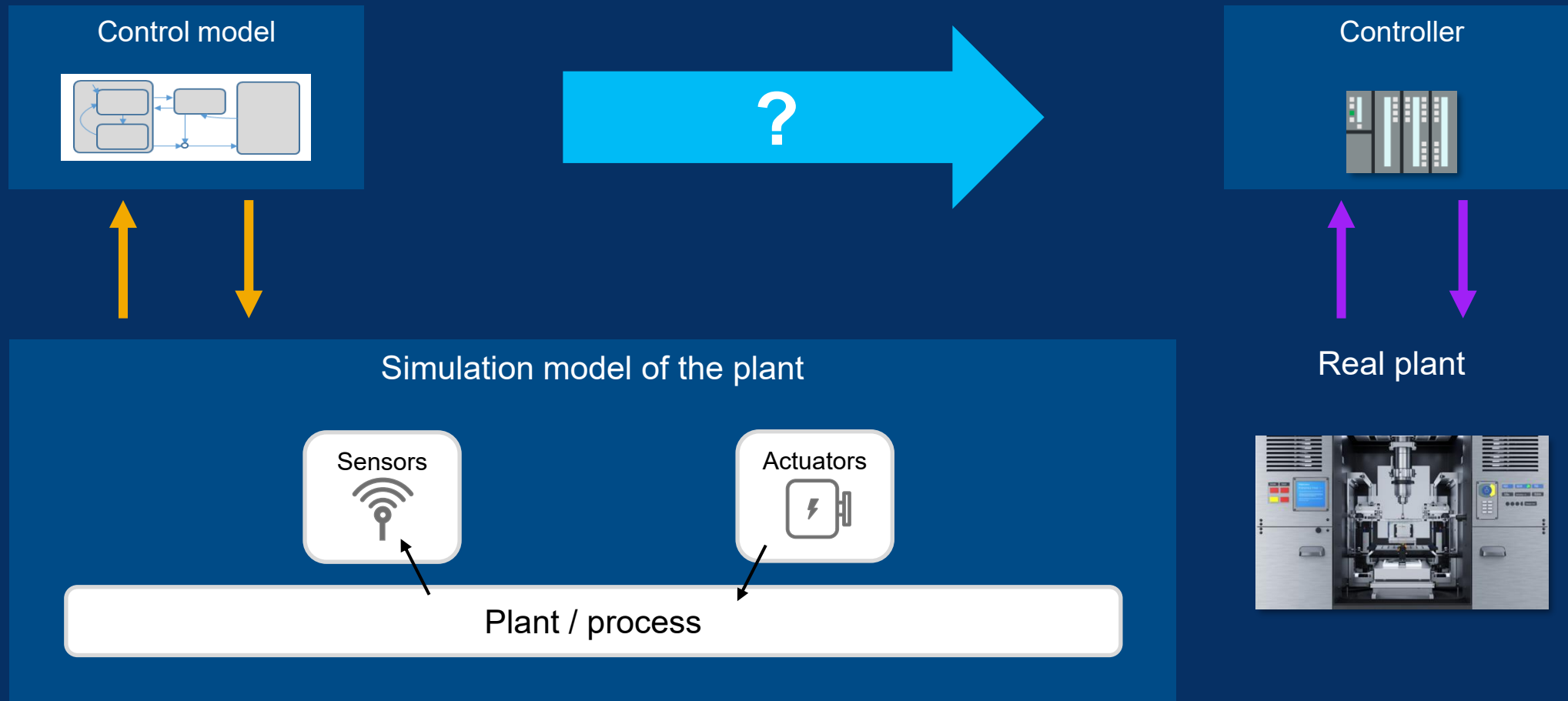
The Value: Safely run automated regression tests (MiL/SiL/HiL), test and train AI, and execute Virtual Commissioning continuously.



Architectural Flexibility & Algorithm Portability

Model-Based Design with Industrial Controllers & PLCs

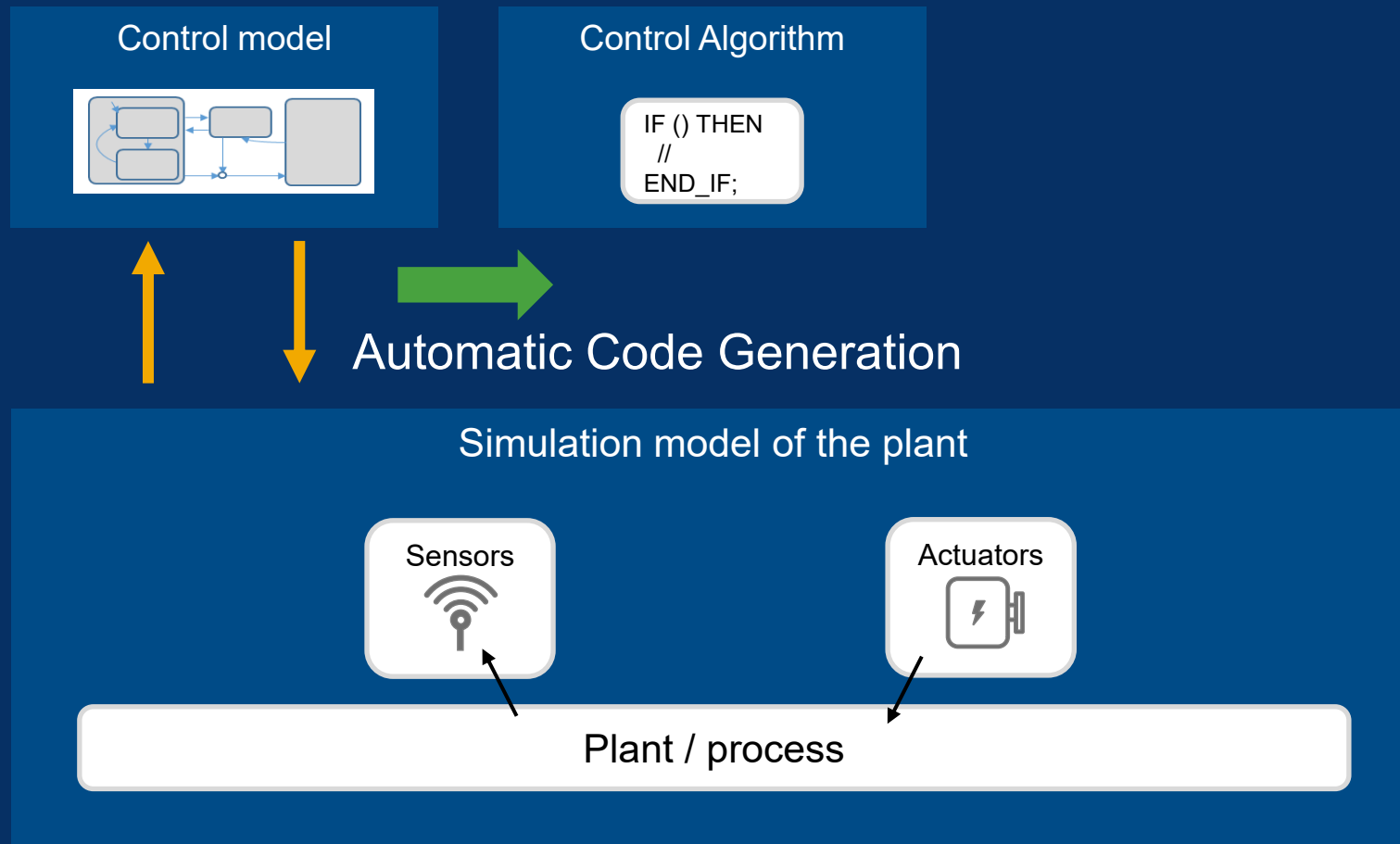
- Non-RT communication
- RT communication



Model-Based Design with Industrial Controllers & PLCs

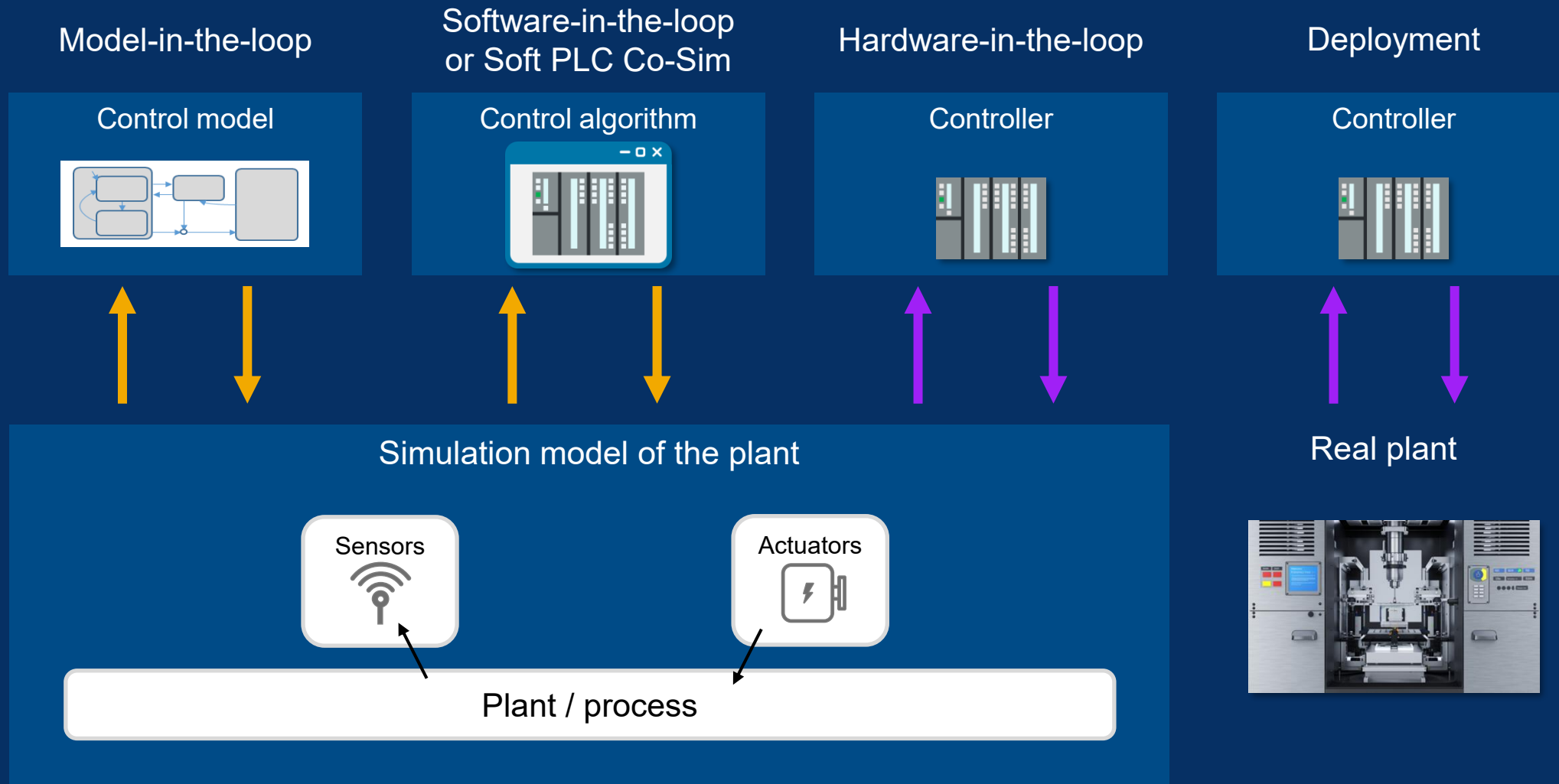
→ Non-RT communication

→ RT communication



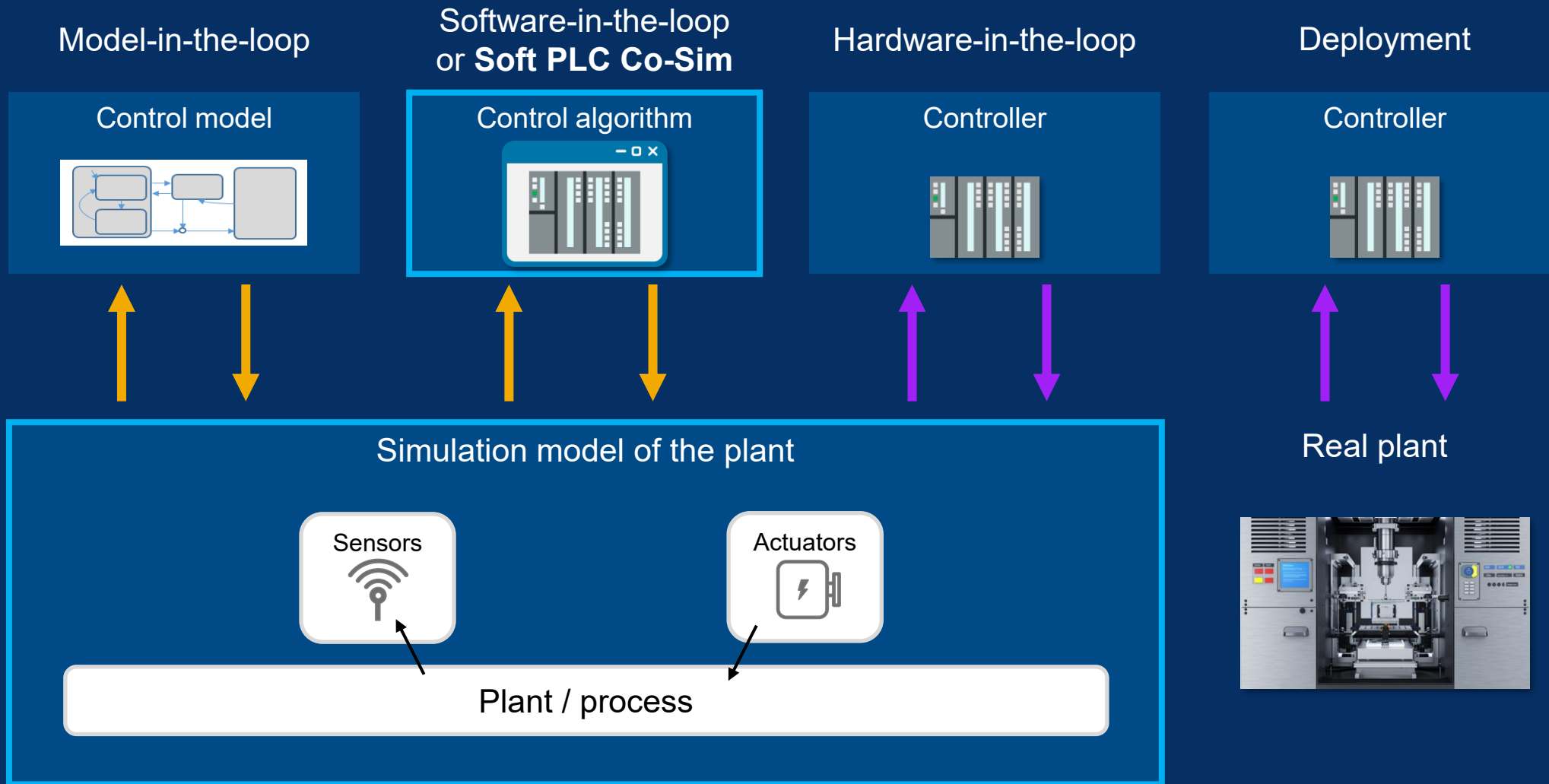
Model-Based Design with Industrial Controllers & PLCs

Non-RT communication
 RT communication



Model-Based Design with Industrial Controllers & PLCs

Non-RT communication
 RT communication



Co-Simulation with Virtual PLCs

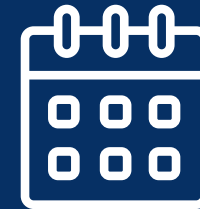
Numerous Use Cases



Early behavior validation
(before hardware is available)



Catching integration issues
earlier



Reducing commissioning risk
and time



Supporting virtual
commissioning and reuse

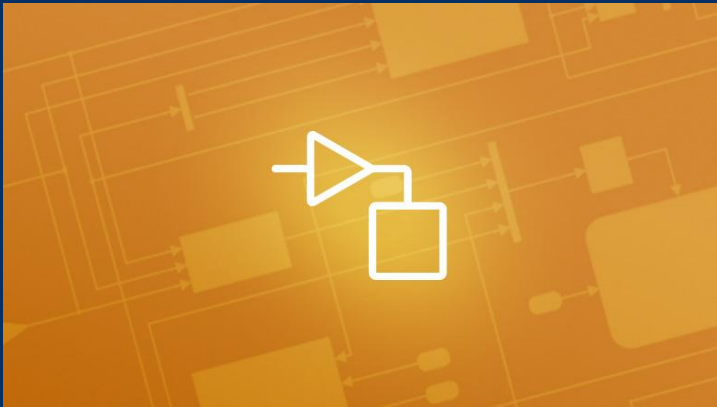


Scenario simulation and
operator training

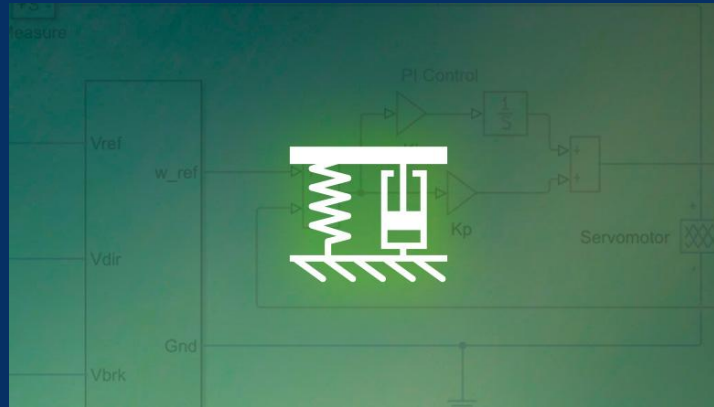


Testing across PLC platforms
and variants

Simulink Getting Started *Onramps*



Simulink Onramp



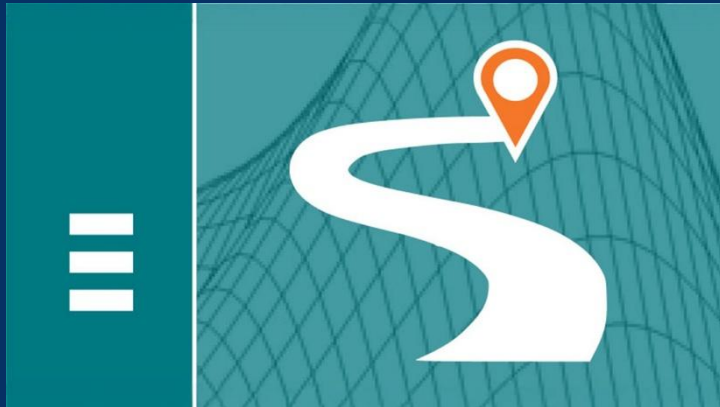
Simscape Onramp



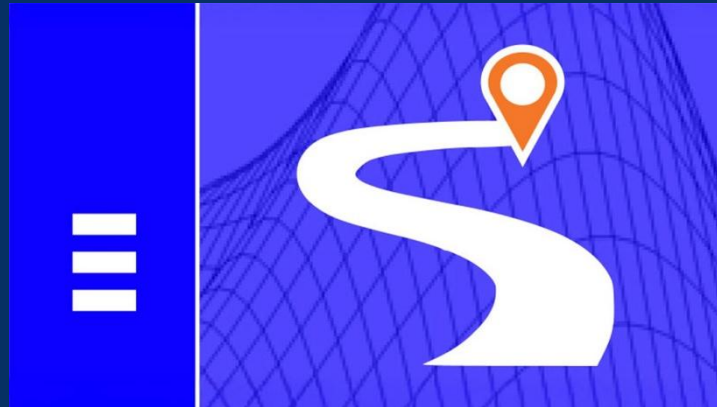
Simscape Multibody Onramp

Simulink Getting Started

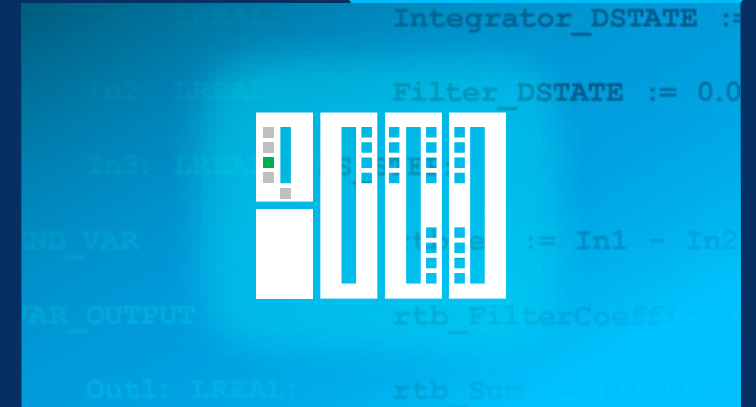
Learning Paths and Training



Build Simulink Proficiency



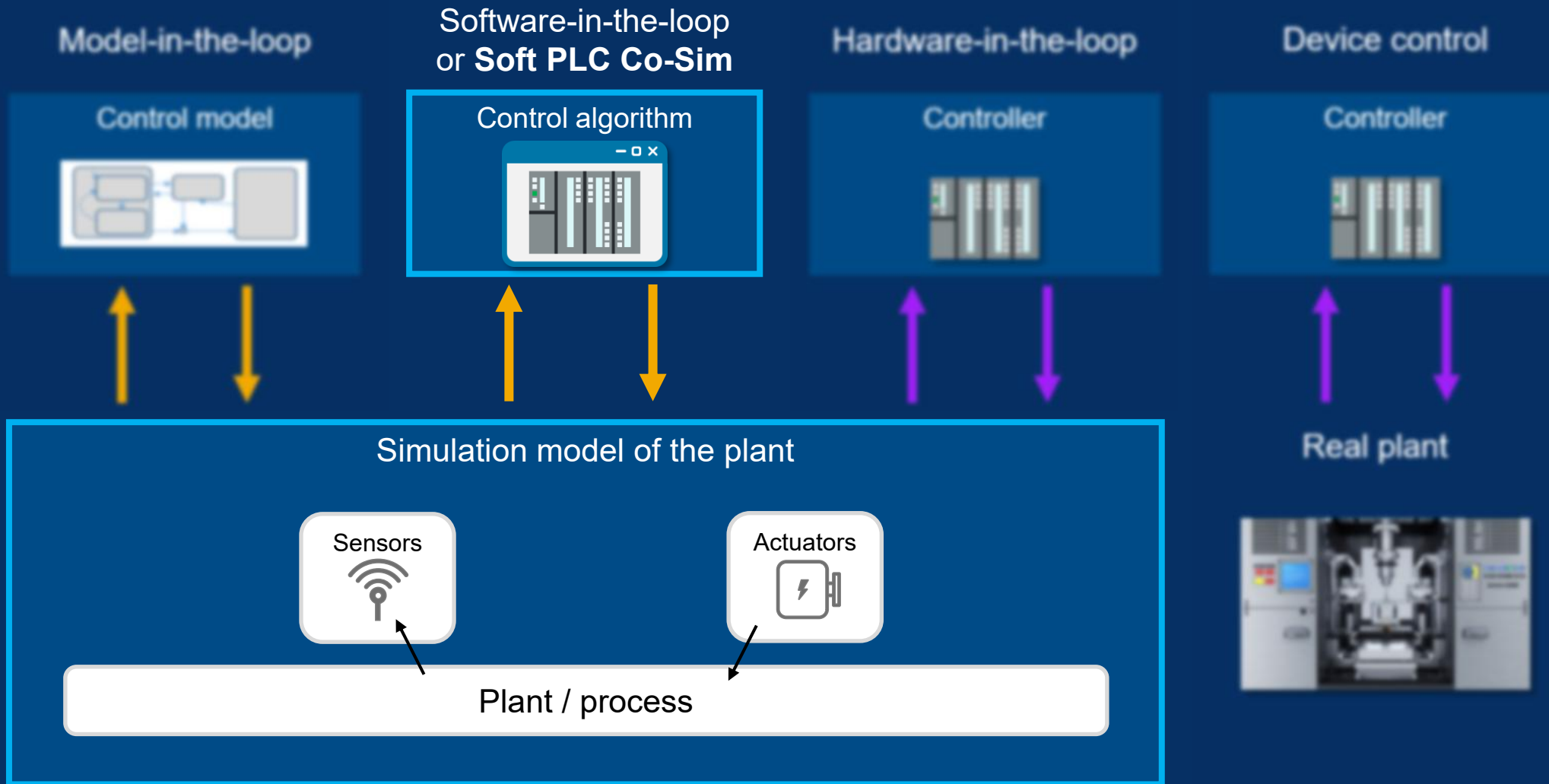
Control System Design with
MATLAB and Simulink



Simulink PLC Coder
Training

Model-Based Design with Industrial Controllers & PLCs

Non-RT communication
 RT communication



Key Takeaways

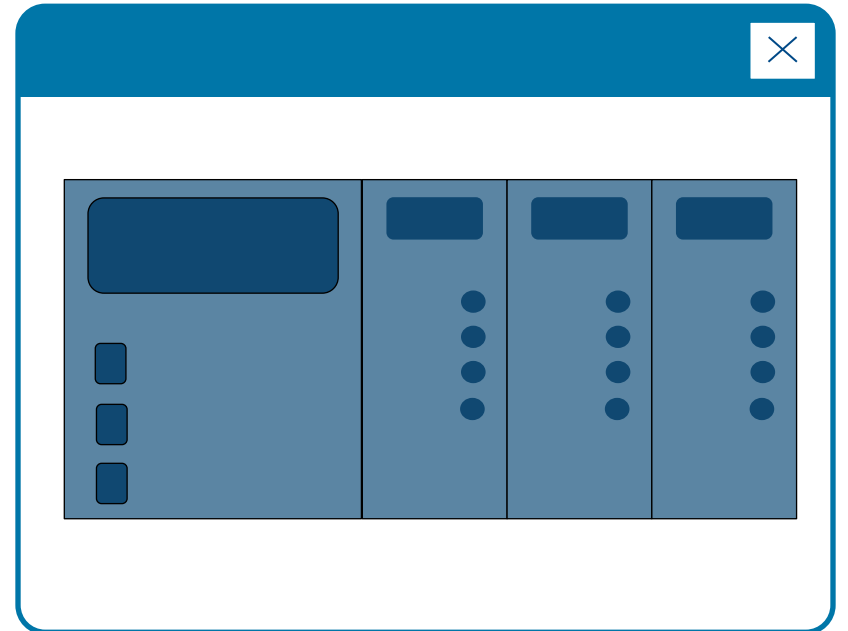
Verifying control logic on Soft PLC using co-simulation let you

- Gain early confidence instead of late validation
- Lower integration risk and faster commissioning
- Create consistency across design, verification, and deployment

Soft PLC – Control Logic Without Hardware Constraints

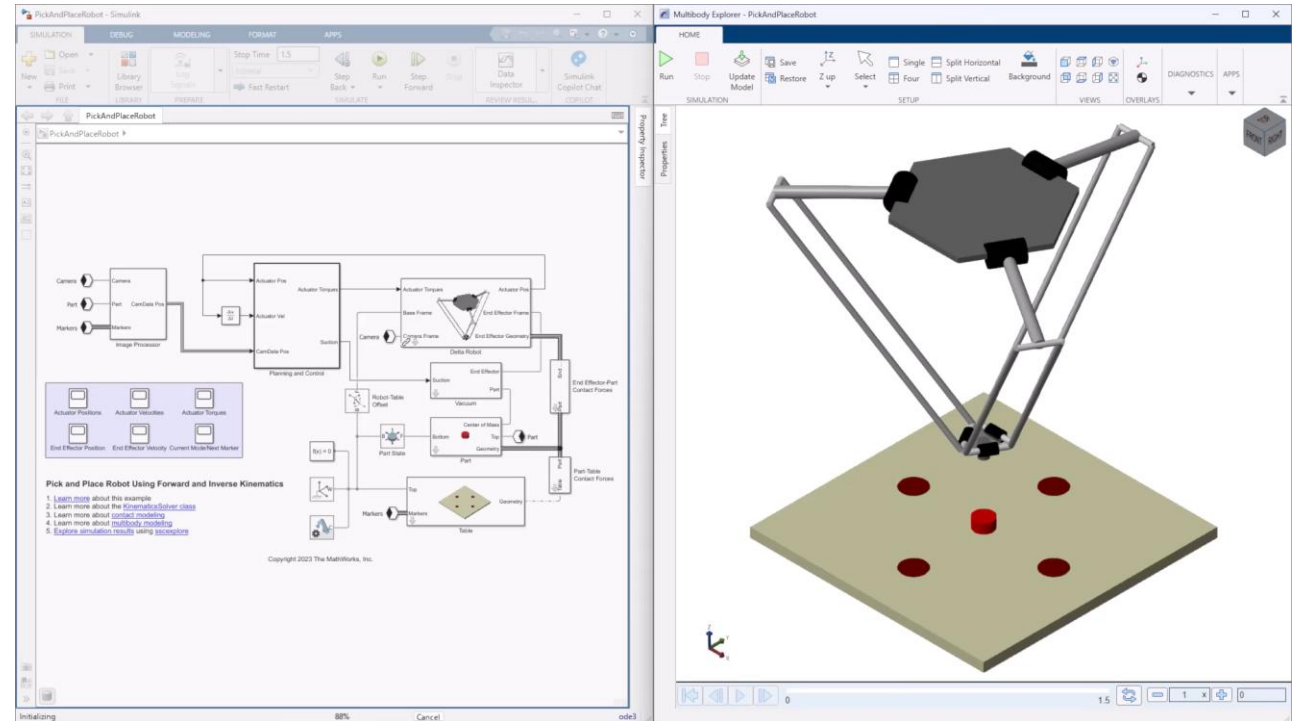
- Software-based PLC runtime
- Executes logic written in IEC 61131-3 languages or C/C++
- Runs on IPC, virtual machines, or servers
- Same control semantics as classical PLCs
- Easier access to interfaces and data

→ Real control logic - detached from hardware



Digital Twin – An Executable Model of the Plant

- Dynamic, time-dependent model
- Responds to control inputs
- Physics-based behavior
- **It is not a static diagram**
- **It is a running system model**
- Typical content of the twin includes
 - Mechanics, hydraulics, electronics
 - Constraints, nonlinearities, fault injection

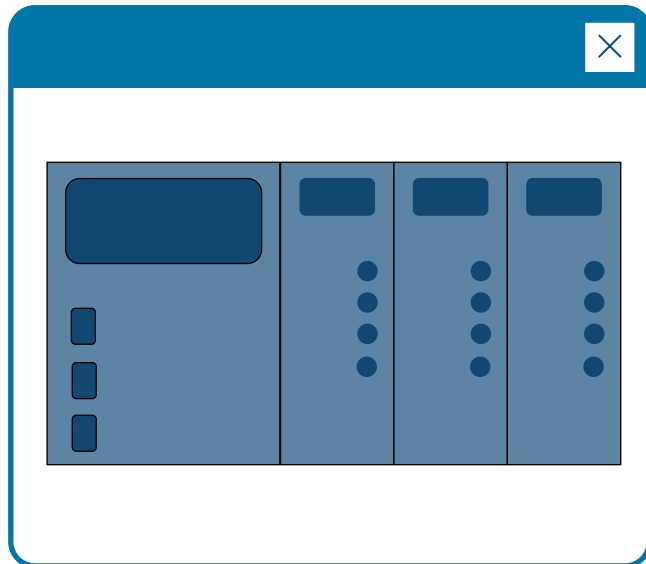


→ The digital twin represents expected plant behavior before hardware exists

Co-Simulation – Bridging Control and System Behavior

- Automation Project executes in PLC IDE
- PLC does not know about simulator

Soft PLC



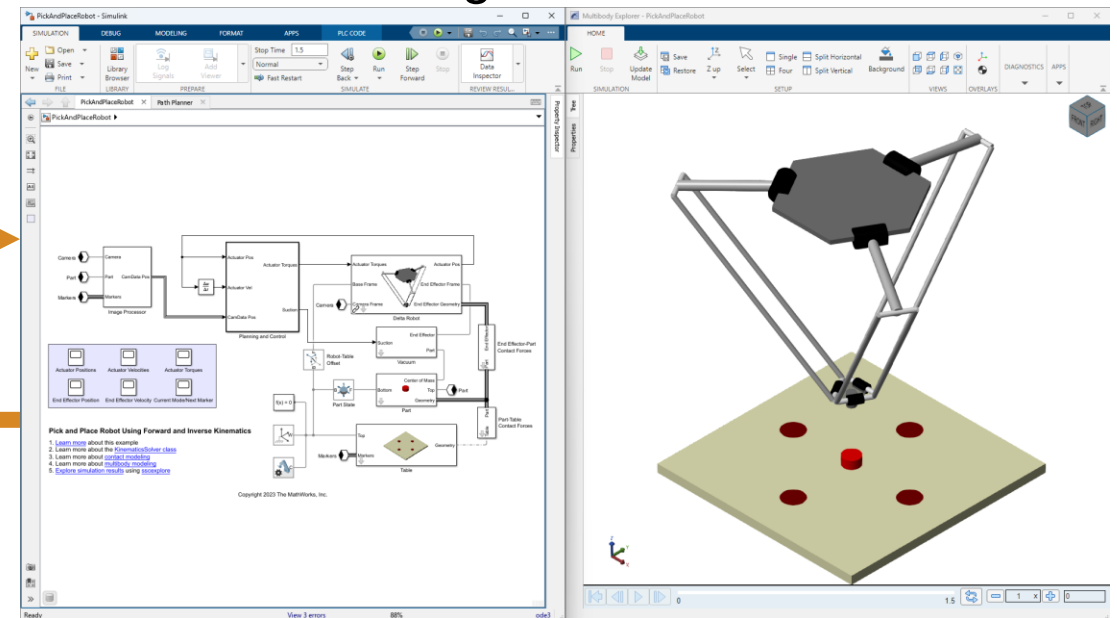
Setpoints



Process Values



Digital Twin

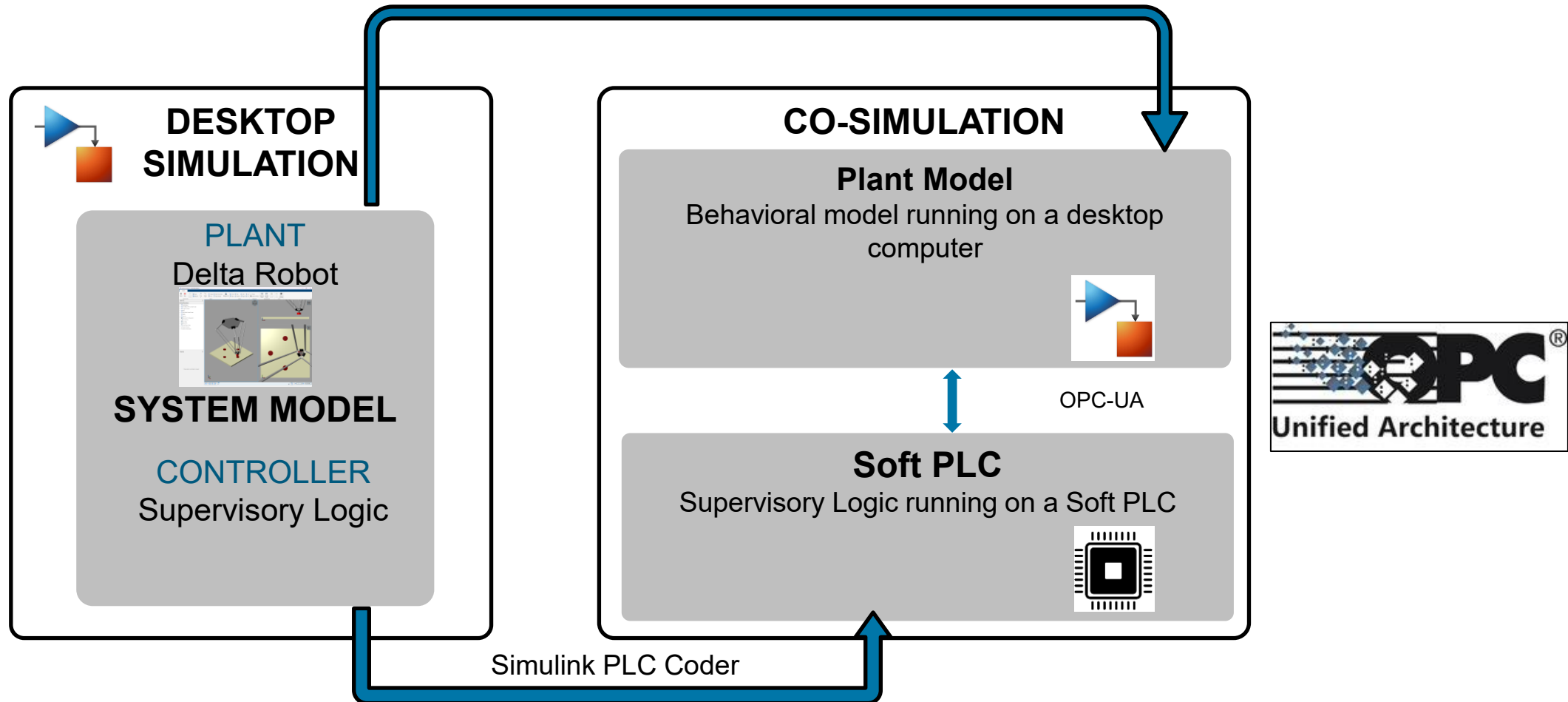


→ Connects real PLC logic with realistic plant behavior

Verifying Control Logic on Soft PLCs Using Co-Simulation

- Generate **Structured Text (ST)** automatically from a Simulink model
- Import the generated code into **Soft PLC**
- Run the control logic on a **Soft PLC**
- Establish an **OPC UA communication** between the Soft PLC and Simulink
- Run the **co-simulation**

Summary



Simulink PLC Coder

- Generates IEC 61131-3
- Supports various IDEs
- Provides verification support including test benches

MathWorks® Products Solutions Academia Support Community Events

Get MATLAB

Simulink PLC Coder

Search MathWorks.com

Simulink PLC Coder

Generate IEC 61131-3 Structured Text and Ladder Diagrams for PLCs and PACs

[Request a free trial](#) [Request a quote](#)

Simulink PLC Coder™ generates hardware-independent IEC 61131-3 Structured Text and Ladder Diagrams from Simulink® models, Stateflow® charts, and MATLAB® functions. Structured Text is generated in PLCopen XML and other file formats supported by widely used integrated development environments (IDEs), including 3S-Smart Software Solutions CODESYS®, Rockwell Automation Studio 5000, Siemens TIA Portal, and Omron® Sysmac® Studio. Ladder diagrams are generated in file formats supported by Rockwell Automation Studio 5000. As a result, you can compile and deploy your application to numerous programmable logic controller (PLC) and programmable automation controller (PAC) devices.

Anti-Windup PID Control Demonstration with Feedforward Control

1:56

[Visit the product website for more information](#)

Industrial Communication Toolbox

- [Industrial Communication Toolbox Product Page](#)
- [OPC UA](#)

Industrial Communication Toolbox

Industrial Communication Toolbox

Exchange data over OPC UA, Modbus, MQTT, and other industrial protocols

[Get a free trial](#) [View pricing](#)

Have questions? [Contact Sales](#).

```
'localhost'  
{'Matrikon.OPC.Simulation.1'}  
: {'Matrikon OPC Server for Simulation and  
ion: {'DA2'}  
ctor: {'opcda('localhost', 'Matrikon.OPC.Simula
```

Industrial Communication Toolbox provides access to live and historical industrial plant data directly from MATLAB and Simulink. You can read, write, and log OPC Unified Architecture (UA) data from devices such as distributed control systems, supervisory control and data acquisition systems, and programmable logic controllers. You can also access plant and manufacturing data directly from AVEVA™ PI System™, and use this data for process monitoring, process improvement, and predictive maintenance applications.

You can work with data from live servers and data historians that conform to the OPC UA, OPC Data Access (DA), and OPC Classic Historical Data Access (HDA) standards. When communicating over

[Show more](#)

[Process Monitoring](#) [Predictive Maintenance](#) [Industrial IoT](#) 2:28

What Is Industrial Communication Toolbox?

Digital Transformation in Automation: Real-Time MATLAB–PLC Integration for Industry 4.0

Shubham Kapur



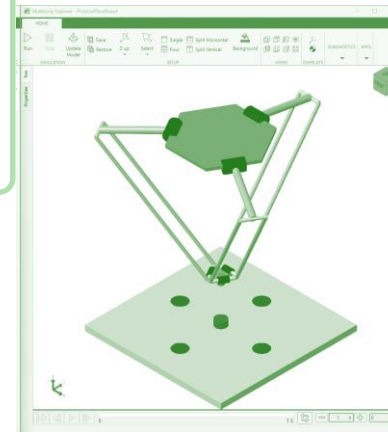
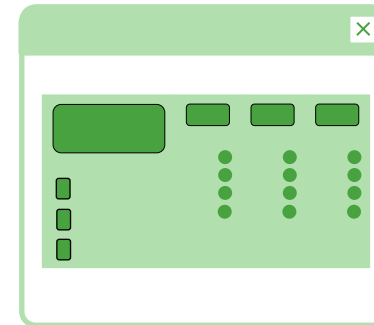
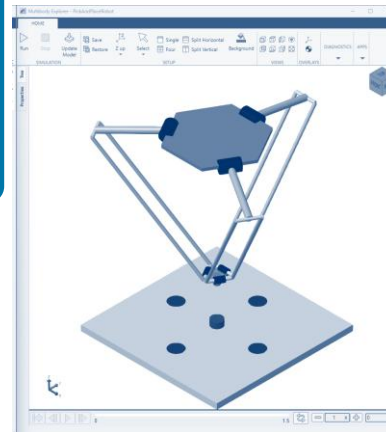
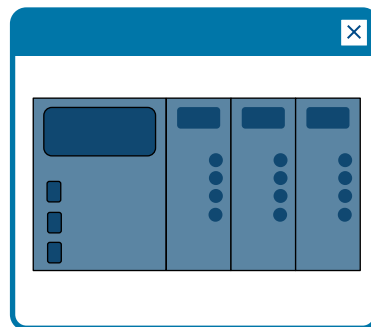
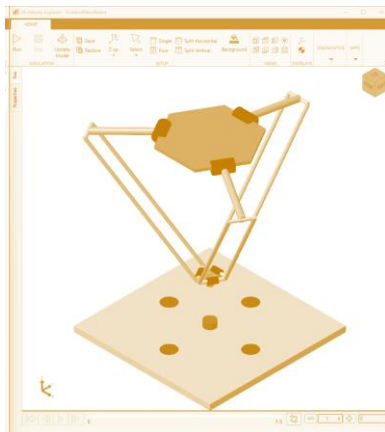
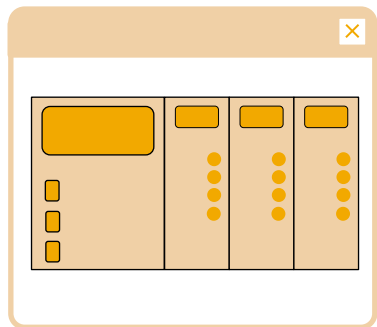
Virtual operator training and testing environment

- Cyclic, deterministic OPC UA exchange
- High-fidelity Simulink/Simscape plant model including fault injection
- Batched signals with shared-memory buffering for reduced jitter for closed-loop validation
- Enables stress-testing control logic and safe fault injection



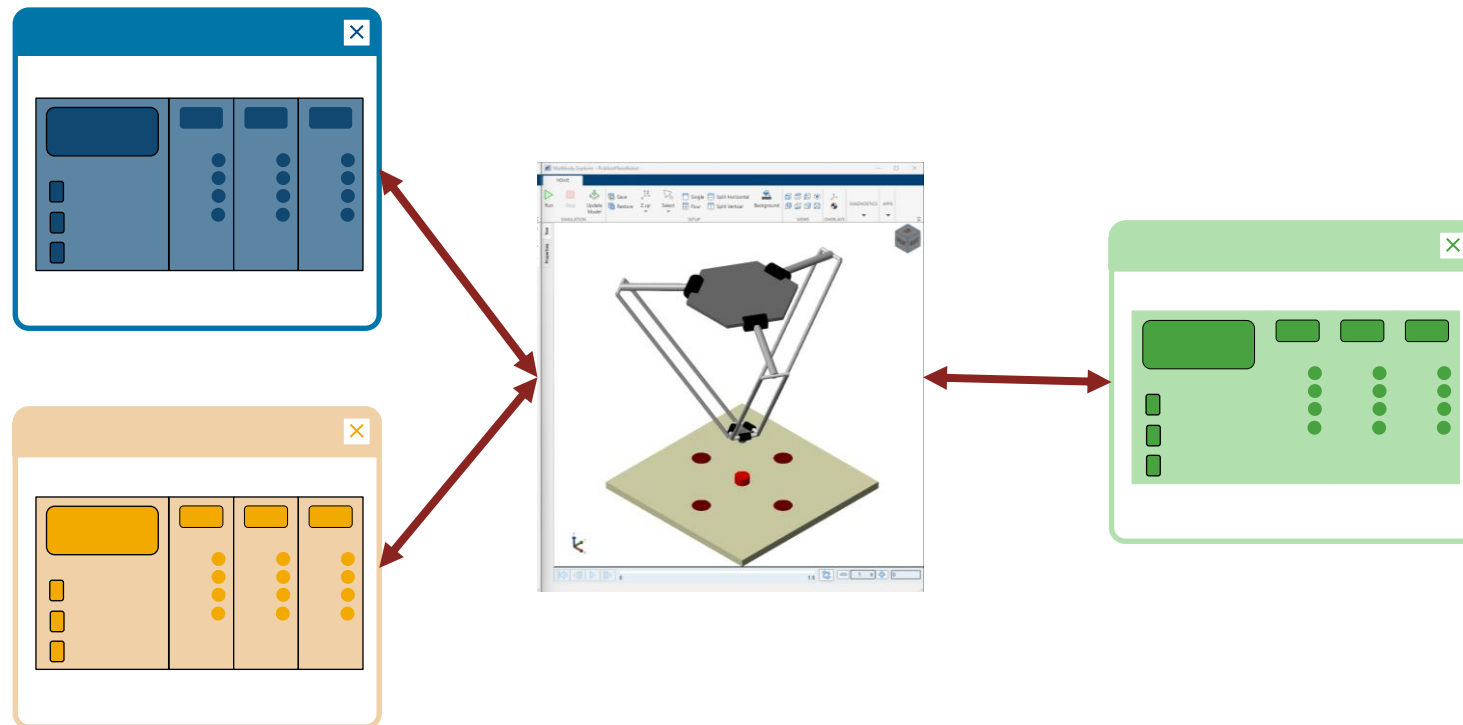
When plant behavior must scale across PLC platforms

- Vendor-specific plant simulation tools are generally tied to a PLC platform
- The same behavior is implemented and maintained multiple times
- Reuse becomes hard as platforms and teams grow



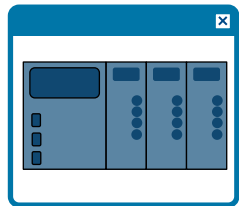
When plant behavior must scale across PLC platforms

Simulink as a target-agnostic platform helps you to reduce efforts and risks associated with multiple implementations of the same plant model



The interface may change — the validated model stays the same

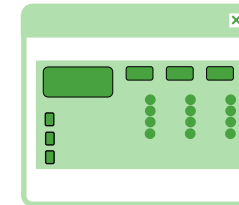
- OPC UA is one practical option for co-simulation – not the only one
- Different projects call for different interface strategies
- The validated control and system model remains unchanged



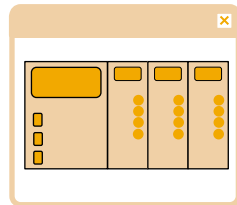
Modbus

MQTT

UDP

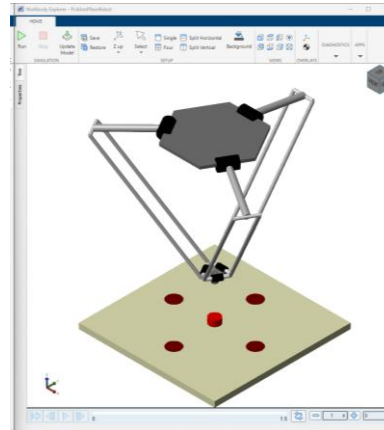


TCP/IP

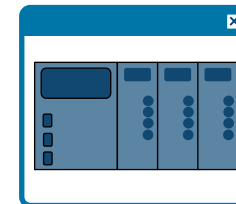


FMU

OPC-UA



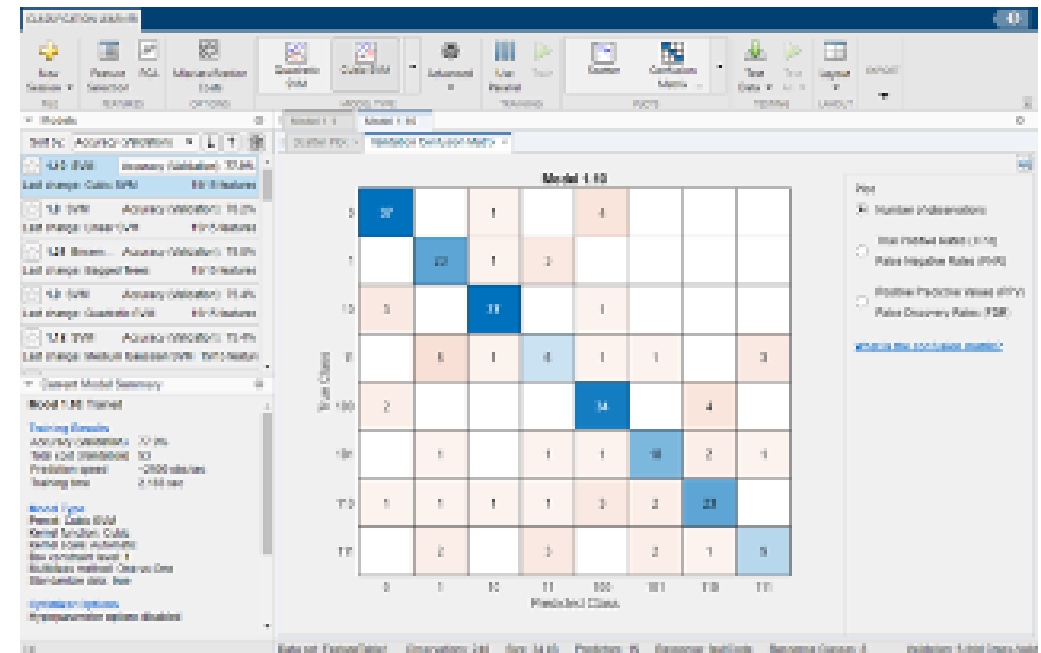
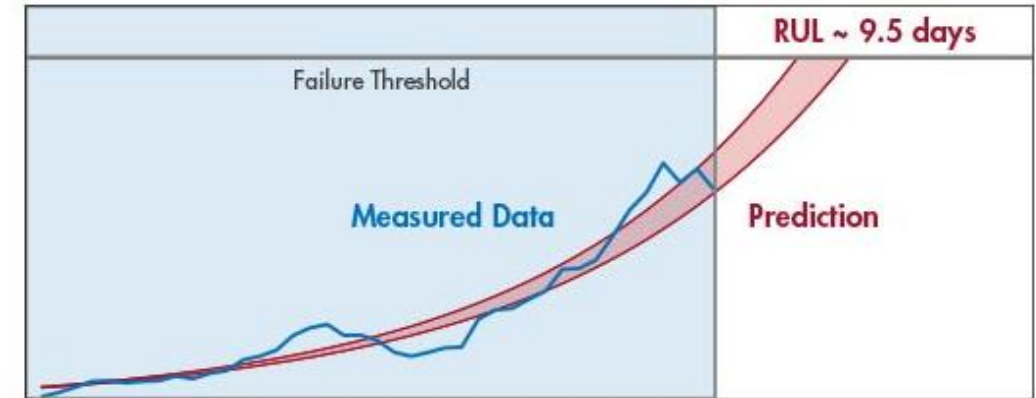
API



Beyond co-simulation: where the digital twin can go next

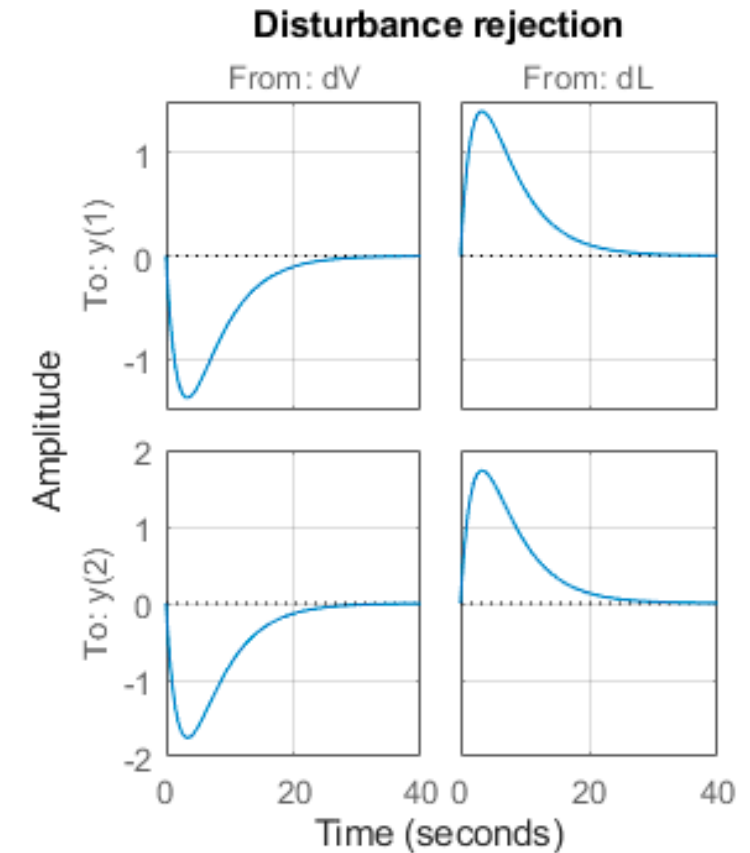
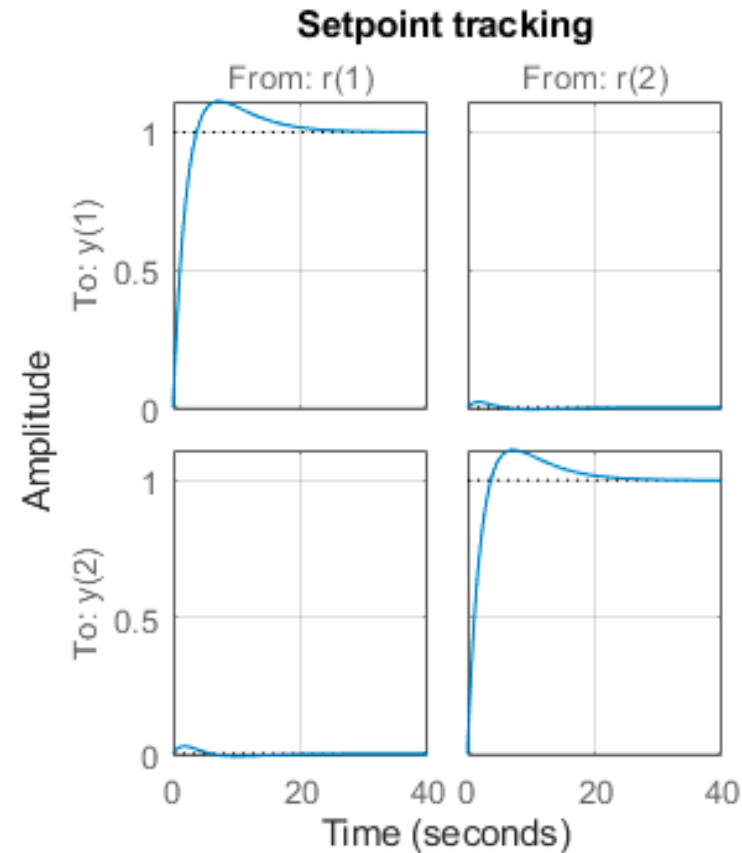
- **Operations and Service**

- Condition monitoring
- predictive maintenance
- Synthetic data generation



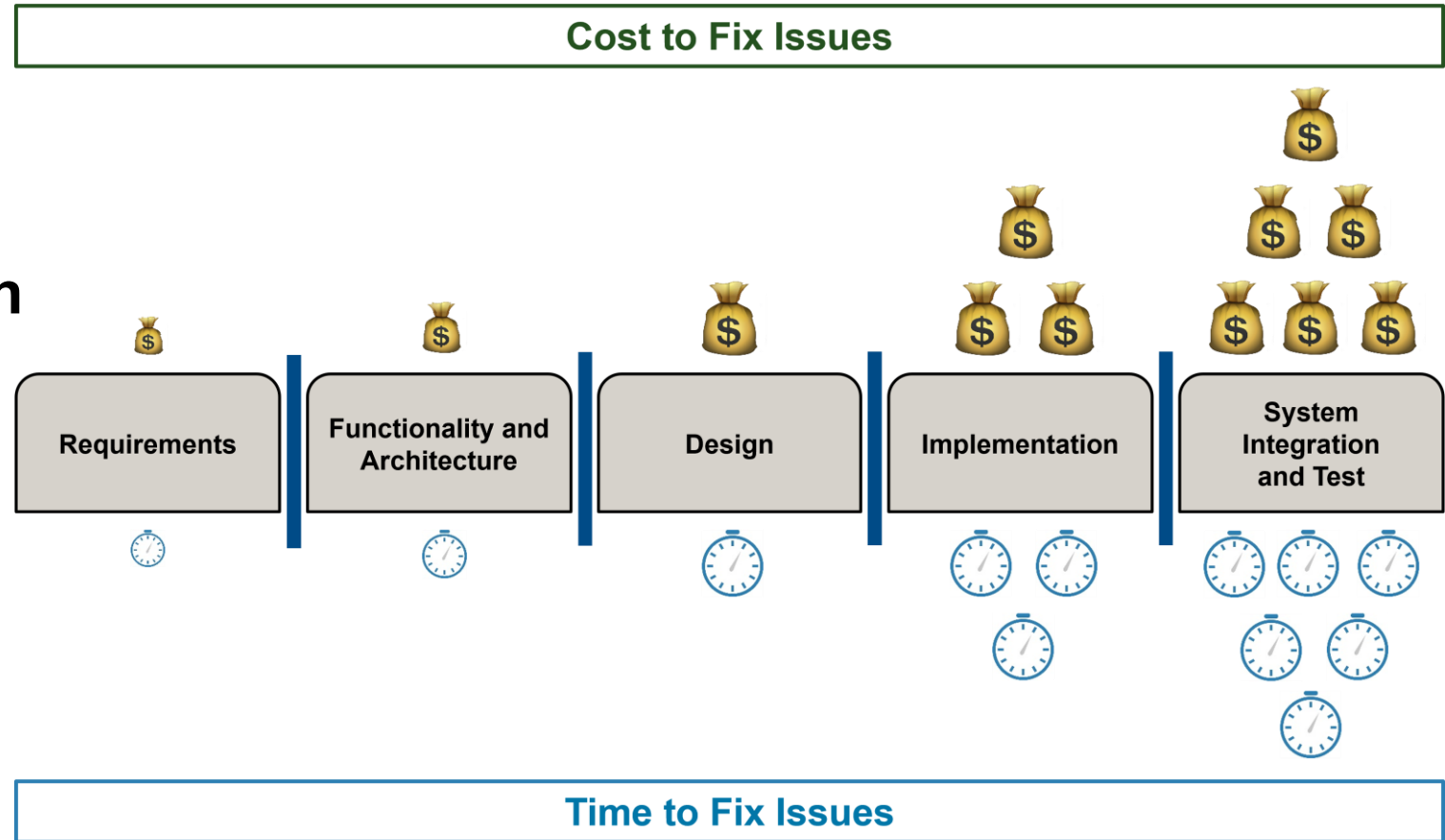
Beyond co-simulation: where the digital twin can go next

- Operations and Service
- **Control design**
 - Tune parameters
 - Try alternative algorithms



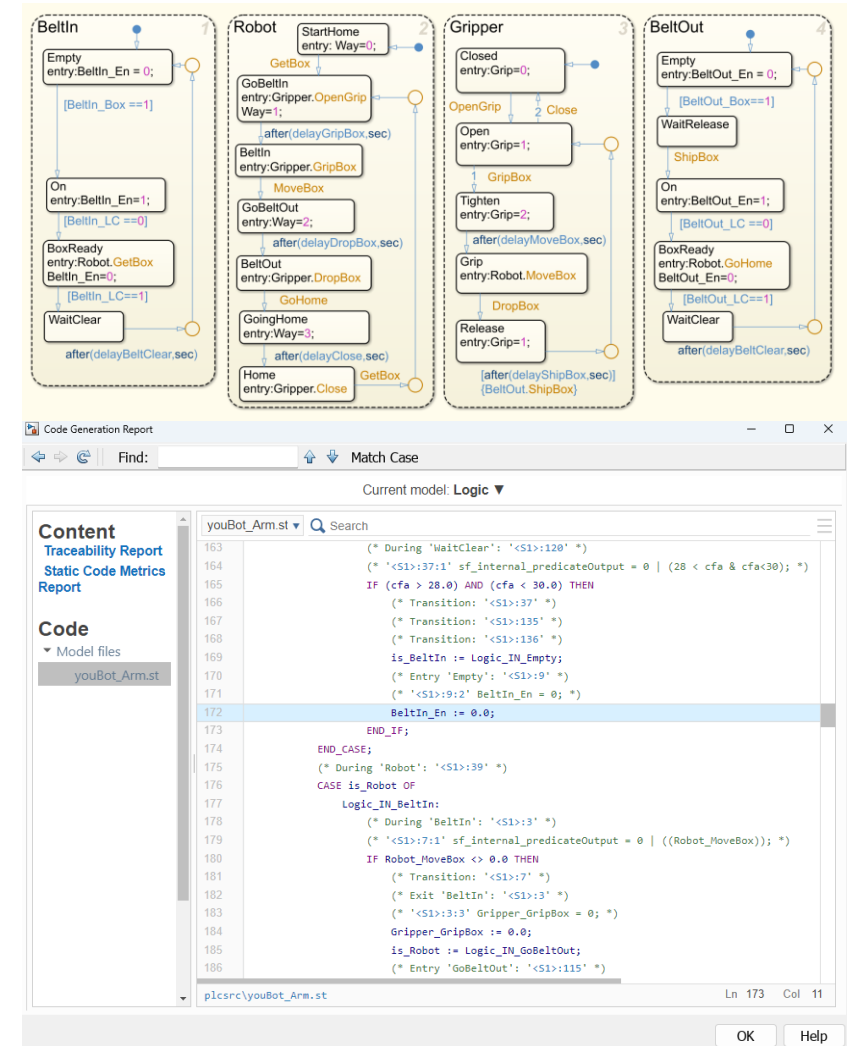
Beyond co-simulation: where the digital twin can go next

- Operations and Service
- Control design
- **Verification and Validation**
 - Automated tests
 - Regression testing
 - Certification



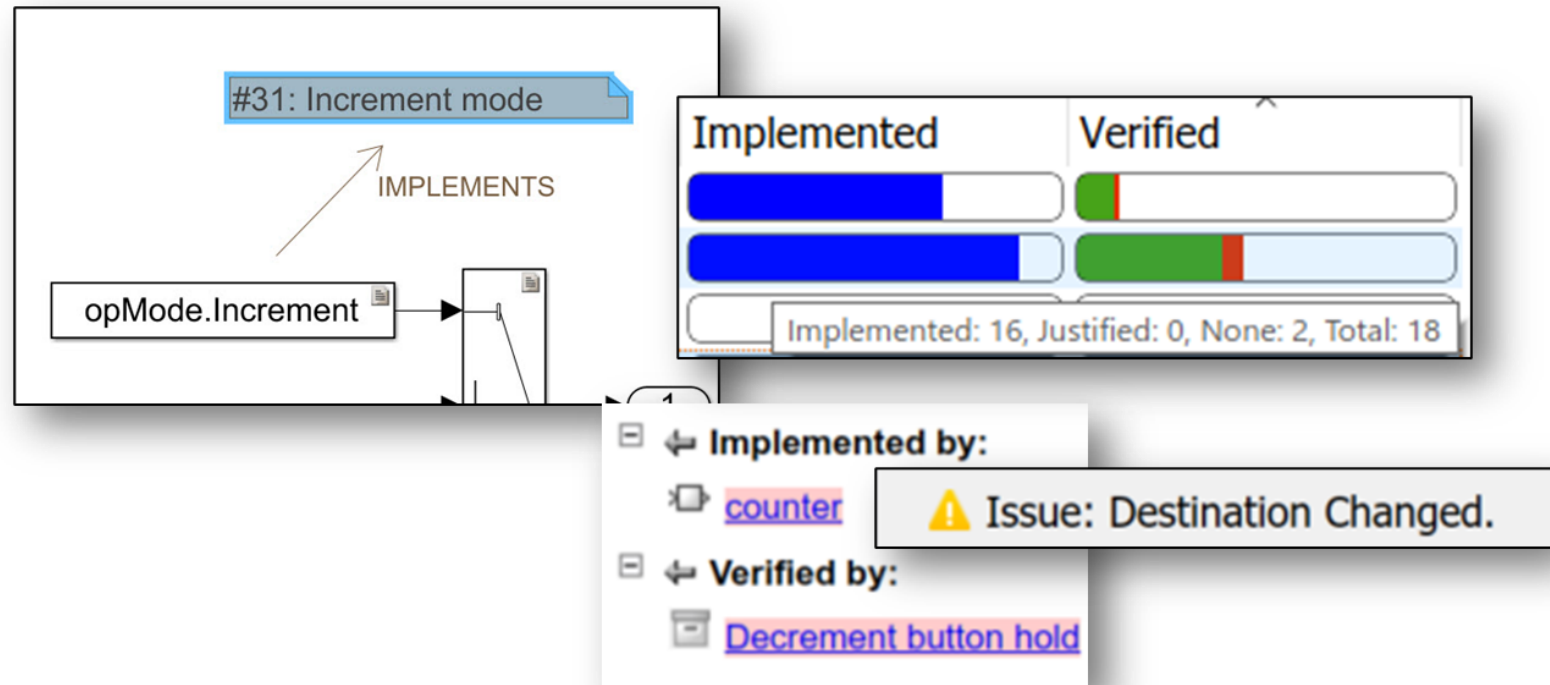
Beyond co-simulation: where the digital twin can go next

- Operations and Service
- Control design
- Verification and Validation
- **Software engineering**
 - Platform-independent code generation for standardized components and advanced algorithms



Beyond co-simulation: where the digital twin can go next

- Operations and Service
- Control design
- Verification and Validation
- Software engineering
- **Systems engineering**
 - Requirement tracing
 - Architectural design
 - Component sizing
 - What-if analysis



Key Takeaways

Verifying control logic on Soft PLCs using co-simulation let you

- Gain early confidence instead of late validation
- Lower integration risk and faster commissioning
- Create consistency across design, verification, and deployment