



Multi-paradigm Modeling & Simulation and System Architecture

Rongjun (Ron) Zhang, Ph.D.

**Sr. Chief Principal Engineer-Controls
Advanced Products Engineering
Oshkosh Corporation**



Background: Vehicle System Design

- **Modern military vehicles are designed to increase mobility, reduce logistic footprints and personnel casualties in the battlefield**
- **Sophisticated vehicle features such as hybrid powertrain, autonomous operation, advanced prognostics and diagnostics etc, impose stringent requirements on vehicle control software design**
- **Exercises such as model-based design, modular architecture, separation of hardware-application, flexible task partitioning are critical to robust software design**

Background: Tools needed

- Simulation platform to conduct physics based modeling and simulation
- Impact of Embedded hardware and communication protocol can be investigated
- Embedded code generation for hardware targets

System Modeling

Control Engineering

Detailed control algorithm and logic

Plant model reflects the system principal and major functionalities

Automatic code generation for the control algorithm

Software/computer Engineering

Operation system information

Execution of communication protocols

Hardware input/outputs and hardware drivers

Simulation tool needed to design and develop an efficient hardware topology



Ideal Platform

- Merging software and controls engineering efforts
- Sharing single simulation environment
- Control and software simulation conducted at different domain and time scale
- Automatic code generation for embedded targets

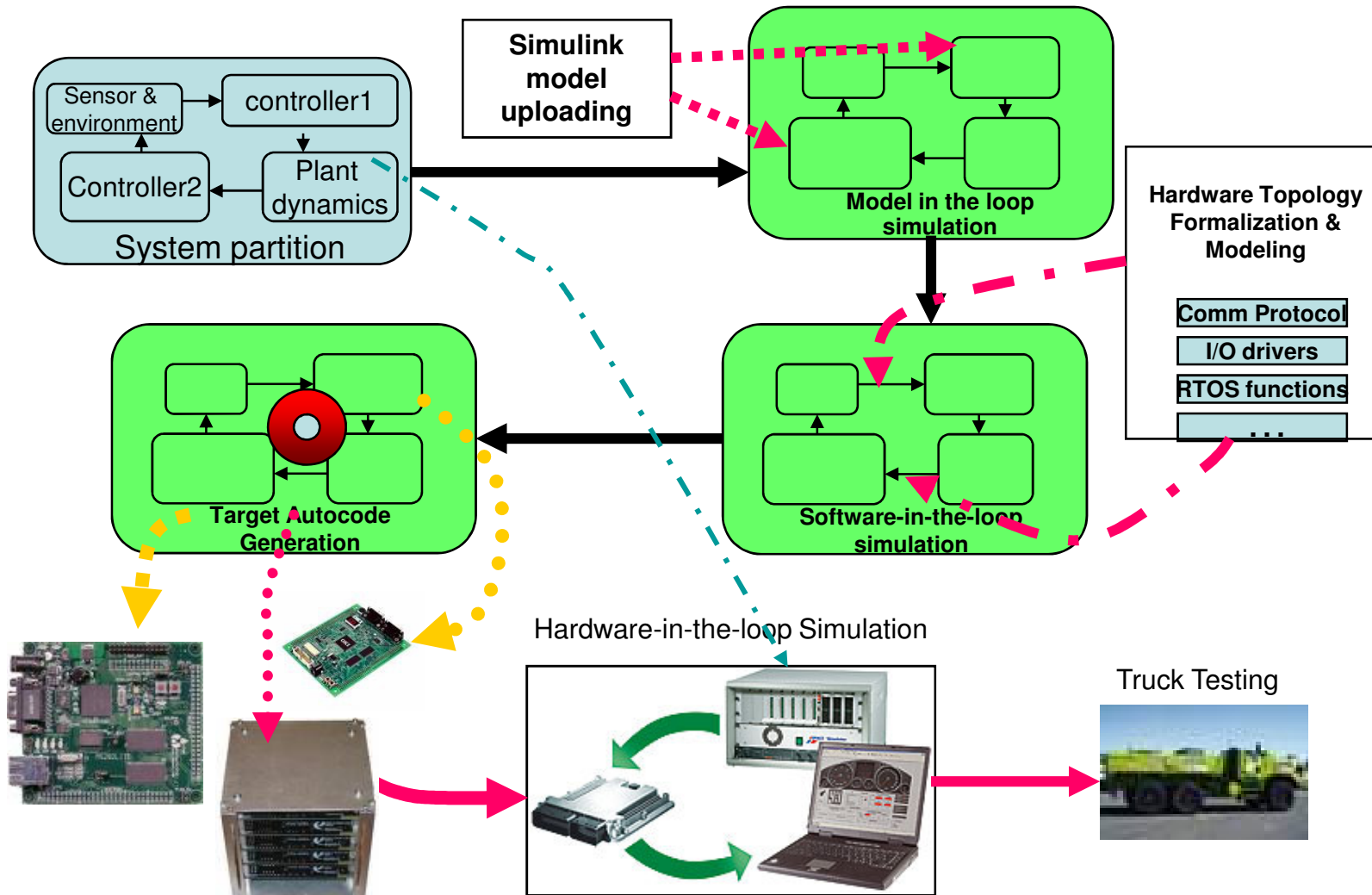
Multi-paradigm Simulation

- **Architecture**
 - **Reconfigurable: allow relocating the application algorithms into different controllers**
- **System level simulation**
 - **Model-in-the-loop simulation**
 - **Physics based modeling and simulation**
 - **Software-in-the-loop simulation**
 - **Hardware topology integrated**
- **Hardware topology simulation**
 - **Influence of the communication protocol/bus traffic**
 - **I/O, driver, digitization interference**
 - **RTOS tasks scheduling effects**

Real-time Implementation

- **Embedded controller implementation**
 - Restart from C/C++? No!
 - Automatic code generation? Yes!
 - With hardware topology? Yes!
 - Heterogeneous controllers? Yes!
 - Networked? Yes!
 - On-line calibration and signal monitoring? Yes!
- **Hardware-in-the-loop simulation**
 - Actual production controller hardware: beyond rapid-prototype controllers

Integrated Simulation and Automatic Code Generation Environment (ISACGE)



Questions?