

The background features a dark blue field on the left and a grey field on the right, separated by a diagonal line. In the upper right, there are white, stylized waveforms. In the lower right, there is a 3D wireframe mesh with a color gradient from yellow at the top to blue at the bottom. Faint blue circuit-like patterns are visible in the bottom right corner.

# MATLAB EXPO 2017

## Simulink as Your Enterprise Simulation Platform

Dr. Mohamed Anas

# Simulink as an Enterprise Simulation Platform

## Simulating Spacecraft Communications for Deep-Space Missions

Dr. Deepak Mishra, Scientist/Engineer (SF)

Indian Space Research Organization



### Challenge

- Integrating large multi-faceted project
- Simulation at multiple stages and in multiple domains to explore the problem

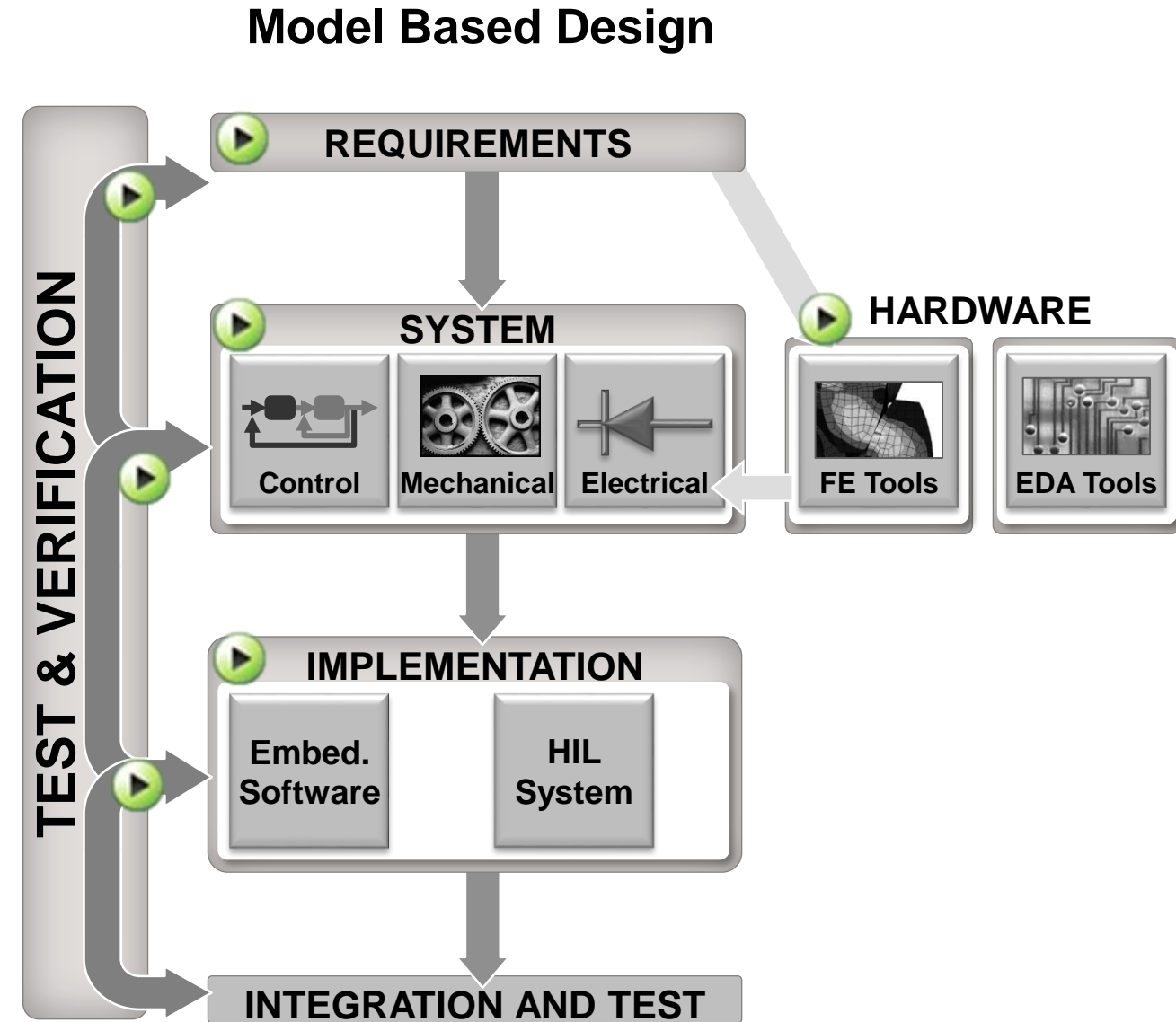
### Solution

- Leverage Simulink as a platform

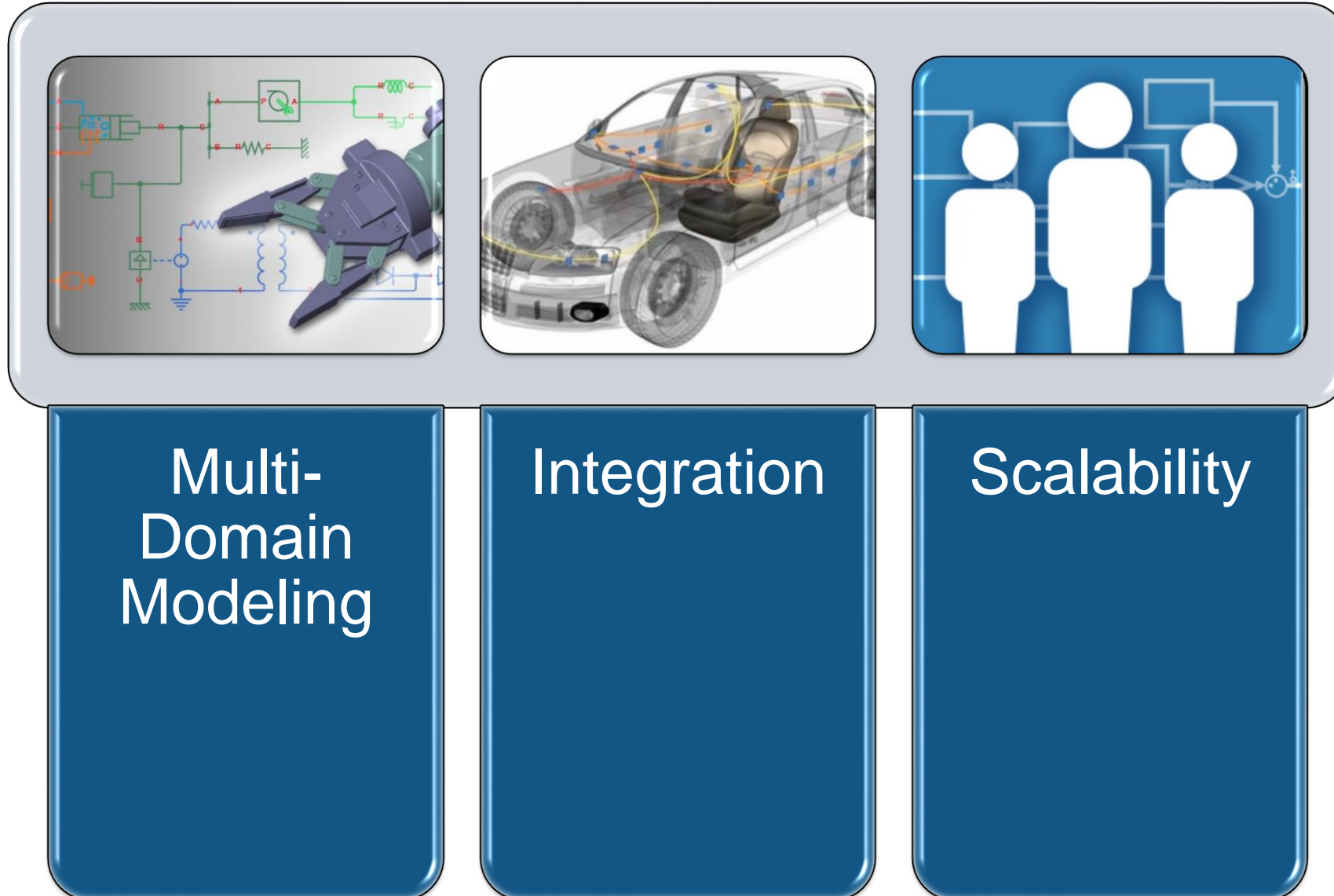
# Enterprise Simulation Platform

- Enterprise - Any size business or project
- Simulation – Evaluating system behavior through computation
- Platform – Scalable environment for multi-disciplinary collaboration

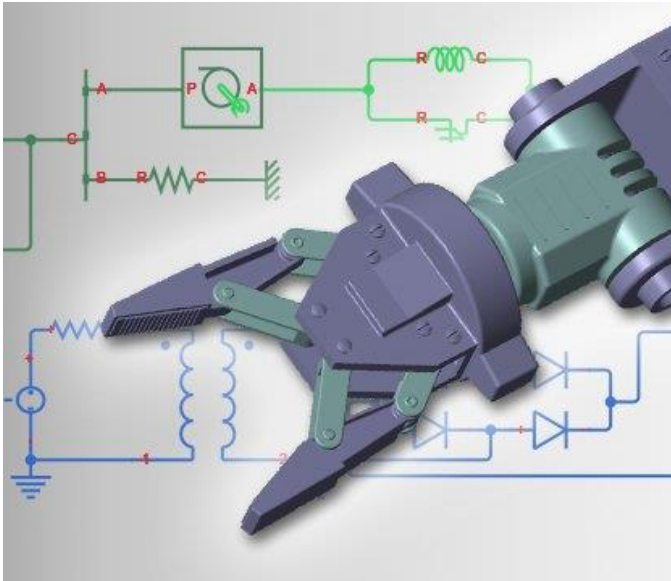
**Simulation**



# Enterprise Simulation Platform Enablers



# Multi-Domain Modeling



# Multi-Domain Modeling in Simulink



Dynamic Systems



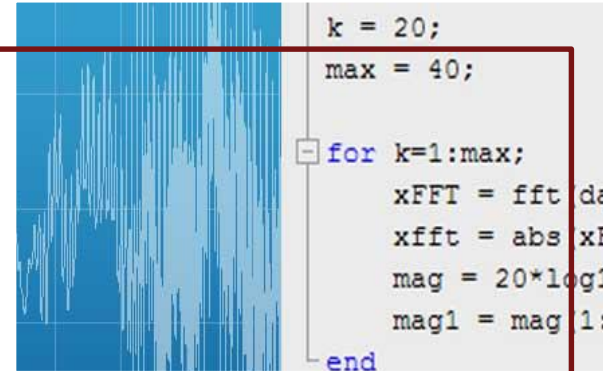
State Machines



Discrete-Event Systems



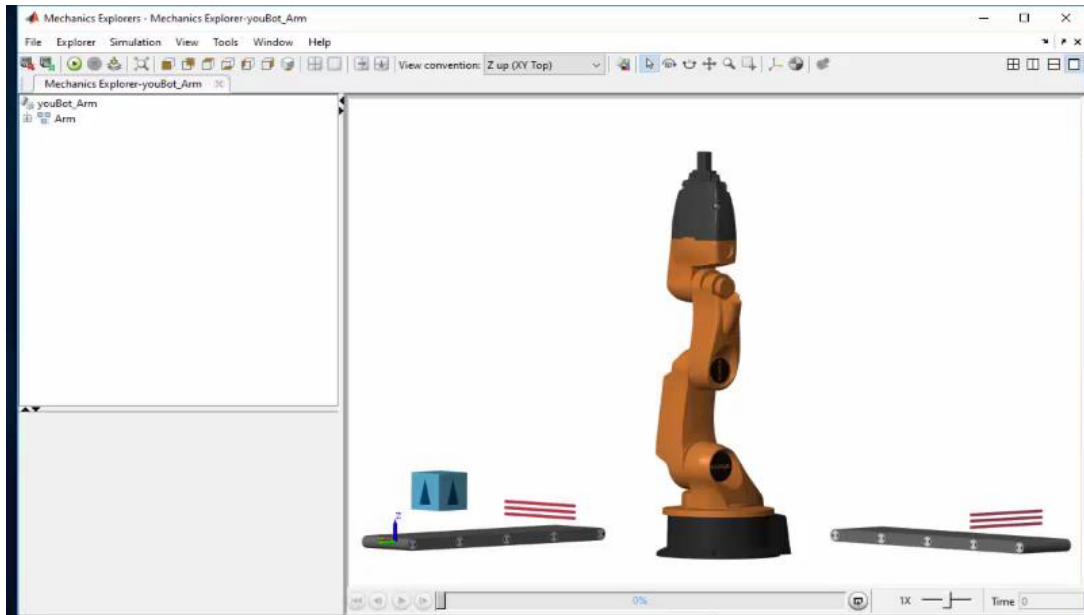
Physical Modeling



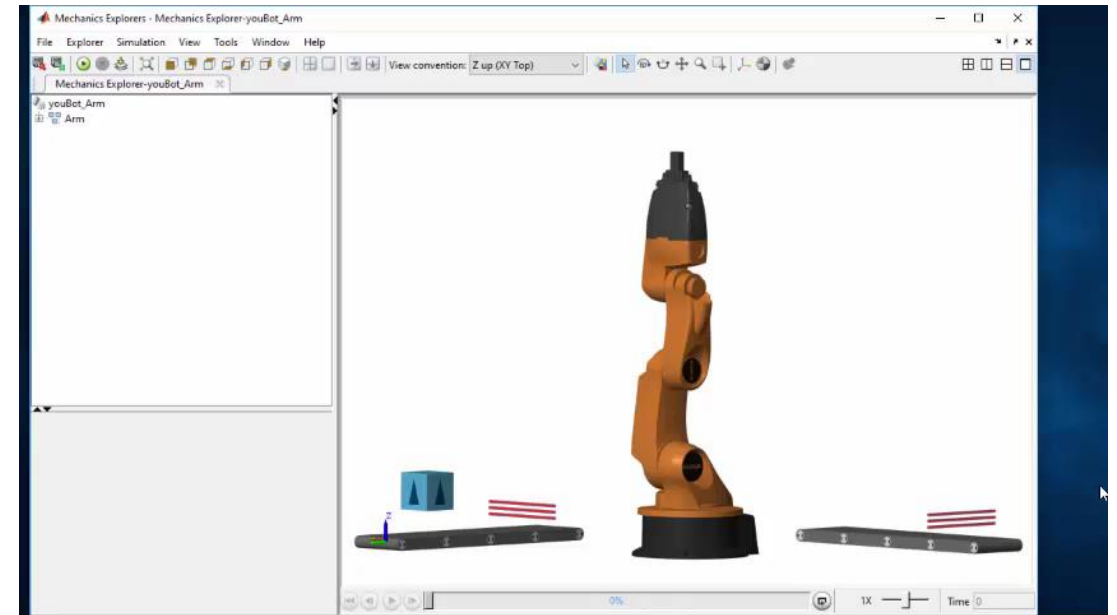
Object-Oriented

# Robot Arm Multi-Domain Simulation

Without Network Model



With Network Model



# Multi-Domain Model

youBot\_Arm - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

youBot\_Arm

**youBot Arm**

1. Plot motor [currents \(code\)](#) and [torques \(code\)](#)
2. Plot joint [angles \(code\)](#) and [forces \(code\)](#)
3. Plot box [trajectory \(code\)](#)
4. [Explore simulation results](#) using [sscexplore](#)
5. Plot optimization results: [Friction](#), [No Friction \(code\)](#)
6. [Compare](#) optimization results ([code](#))
7. Load [model parameters \(code\)](#)
8. [Learn more](#) about this example

**Configure Test** : [Default \(code\)](#)  
 Box Transfer only: [Linear](#); Splines: [Manual](#), [Optim \(friction\)](#), [Optim \(no friction\)](#)  
 Joint Tests: [Pivot](#), [Bicep](#), [Forearm](#), [Wrist](#), [Max Torque](#), [All 35](#)

Run optimization: [Friction](#), [No Friction \(code\)](#)

Ready 123% ode15s



# State Charts and System Dynamics

**youBot Arm**

Home

Belt In

Belt Out

Arm

Belt

Input Control

Network (CAN Bus)

Out1

In1

**Stateflow (chart) youBot\_Arm/Input/Control/Logic - Simulink**

Logic

BeltIn

Robot

Gripper

BeltOut

**youBot Arm**

1. Plot motor [currents \(code\)](#) and [torques \(code\)](#)
2. Plot joint [angles \(code\)](#) and [forces \(code\)](#)
3. Plot box [trajectory \(code\)](#)
4. [Explore simulation results](#) using [sscexplore](#)
5. Plot optimization results: [Friction, No Friction \(code\)](#)
6. [Compare optimization results \(code\)](#)
7. Load [model parameters \(code\)](#)
8. [Learn more](#) about this example

**Configure Test (code)**

Box Transfer only: [Manual, Or](#)

Joint Tests: [Pivot, Biceps, Max](#)

Run optimization: [Friction, No Friction](#)

**youBot\_Arm/.../Control/Joint Commands - Simulink**

Joint Commands

Way

Grip

Integrator

Rate Limiter

Scope

# Multi-Domain Model

**youBot Arm**

1. Plot motor [currents \(code\)](#) and [torques \(code\)](#)
2. Plot joint [angles \(code\)](#) and [forces \(code\)](#)
3. Plot box [trajectory \(code\)](#)
4. [Explore simulation results](#) using [sscexplore](#)
5. Plot optimization results: [Friction](#), [No Friction \(code\)](#)
6. [Compare](#) optimization results ([code](#))
7. Load [model parameters \(code\)](#)
8. [Learn more](#) about this example

**Configure Test** : [Default \(code\)](#)  
 Box Transfer only: [Linear](#); Splines: [Manual](#), [Optim \(friction\)](#), [Optim \(no friction\)](#)  
 Joint Tests: [Pivot](#), [Bicep](#), [Forearm](#), [Wrist](#), [Max Torque](#), [All 35](#)  
 Run optimization: [Friction](#), [No Friction \(code\)](#)

# Physical Modeling

youBot\_Arm - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

youBot\_Arm

youBot\_Arm

Input Control

Arm

Network (CAN Bus)

Arm

youBot\_Arm/Arm - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

Arm

World

Belt

Environment

Finger A

Finger B

Base

Pivot

Bicep

Forearm

Wrist

Gripper

Actuation Motion

Configure Actuation: [Motion](#), [Motor](#)

Arm

116%

ode15s

123%

ode15s

**Configure Test :** [Default](#) ([code](#))  
 Box Transfer only: [Linear](#); Splines: [Manual](#), [Optim \(friction\)](#), [Optim \(no friction\)](#)  
 Joint Tests: [Pivot](#), [Bicep](#), [Forearm](#), [Wrist](#), [Max Torque](#), [All 35](#)

Run optimization: [Friction](#), [No Friction](#) ([code](#))

# Multi-Domain Model

**youBot Arm**

1. Plot motor [currents \(code\)](#) and [torques \(code\)](#)
2. Plot joint [angles \(code\)](#) and [forces \(code\)](#)
3. Plot box [trajectory \(code\)](#)
4. [Explore simulation results](#) using [sscexplore](#)
5. Plot optimization results: [Friction](#), [No Friction \(code\)](#)
6. [Compare](#) optimization results ([code](#))
7. Load [model parameters \(code\)](#)
8. [Learn more](#) about this example

**Configure Test** : [Default \(code\)](#)  
 Box Transfer only: [Linear](#); Splines: [Manual](#), [Optim \(friction\)](#), [Optim \(no friction\)](#)  
 Joint Tests: [Pivot](#), [Bicep](#), [Forearm](#), [Wrist](#), [Max Torque](#), [All 35](#)  
 Run optimization: [Friction](#), [No Friction \(code\)](#)

# Discrete-Event Modeling

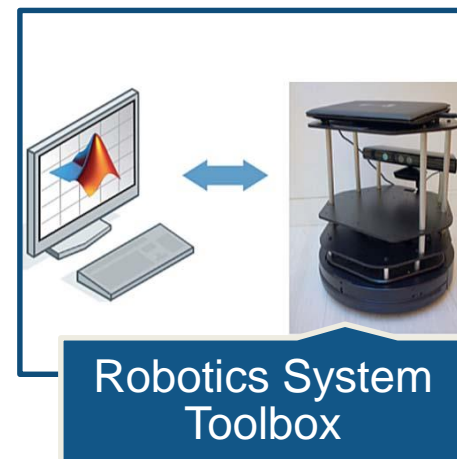
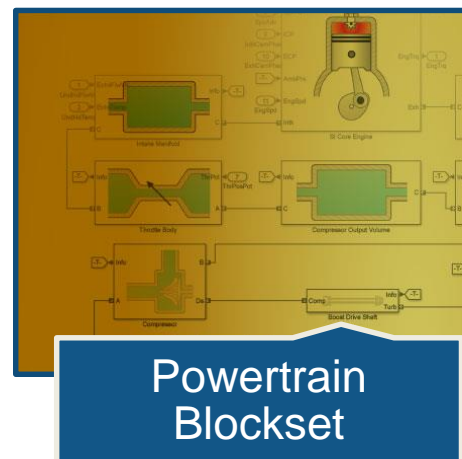
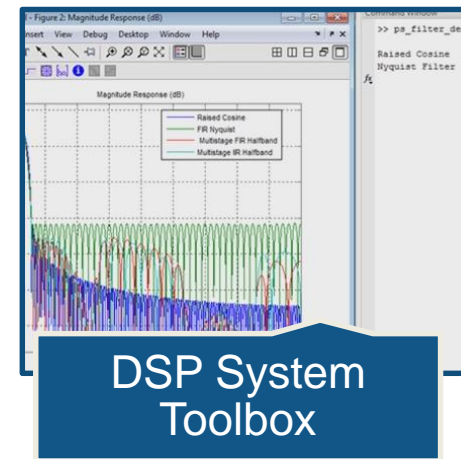
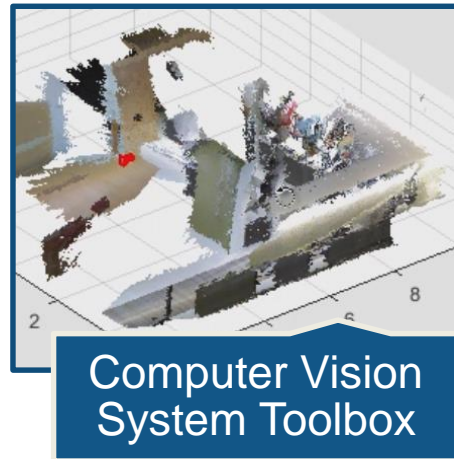
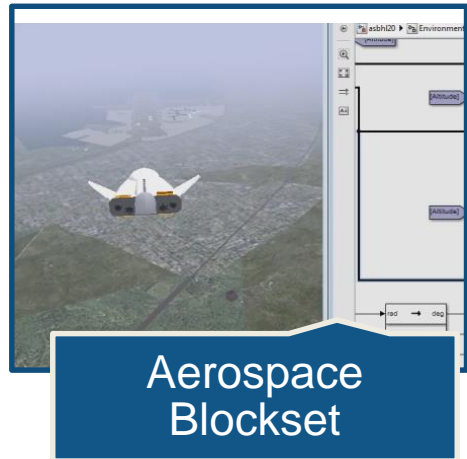
**youBot Arm**

1. Plot motor [currents \(code\)](#) and [torques \(code\)](#)
2. Plot joint [angles \(code\)](#) and [forces \(code\)](#)
3. Plot box [trajectory \(code\)](#)
4. [Explore simulation results](#) using [sscexplore](#)
5. Plot optimization results: [Friction](#), [No Friction \(code\)](#)
6. [Compare](#) optimization results ([code](#))
7. Load [model parameters \(code\)](#)
8. [Learn more](#) about this example

**Configure Test :** [Default \(code\)](#)  
 Box Transfer only: [Linear](#); Splines:  
 Joint Tests: [Pivot](#), [Bicep](#), [Forearm](#),  
 Run optimization: [Friction](#), [No Friction](#)

# Domain-Specific Blocksets and Toolboxes

- Simulink has numerous domain-specific tools, for example:



# Customer Success in Multidomain Modeling

## ABB, Deltamarin, and VTT Simulate and Optimize Ship Energy Flows

### Challenge

- Increase the energy efficiency of large vessels

### Solution

- Use Simulink and Simscape to model, simulate, and optimize ship energy flow

### Results

- Cost- and fuel-saving design improvements
- Testing costs reduced by tens of thousands of euros



# Customer Success in Multidomain Modeling

“Simulink and Simscape enabled us to create a dynamic model of a complex energy system that spans several physical domains. By simulating this model, we can see how a new energy subsystem will perform before it is built, and provide customers with an accurate estimate of their return on investment.”

Juha Orivuori, ABB



## Solution

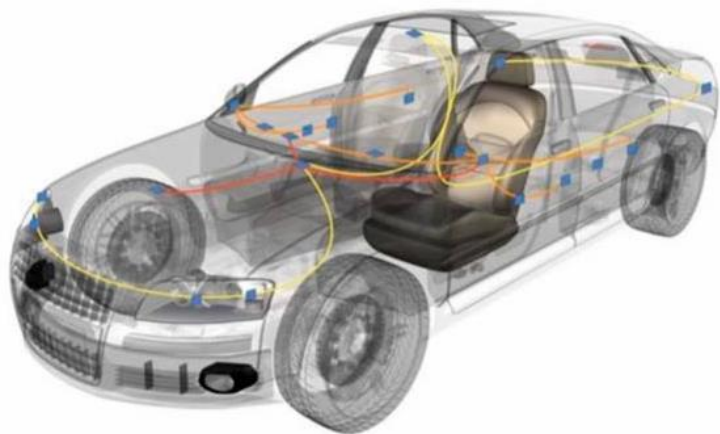
- Use Simulink and Simscape to model, simulate, and optimize ship energy flow

## Results

- Cost- and fuel-saving design improvements
- Testing costs reduced by tens of thousands of euros

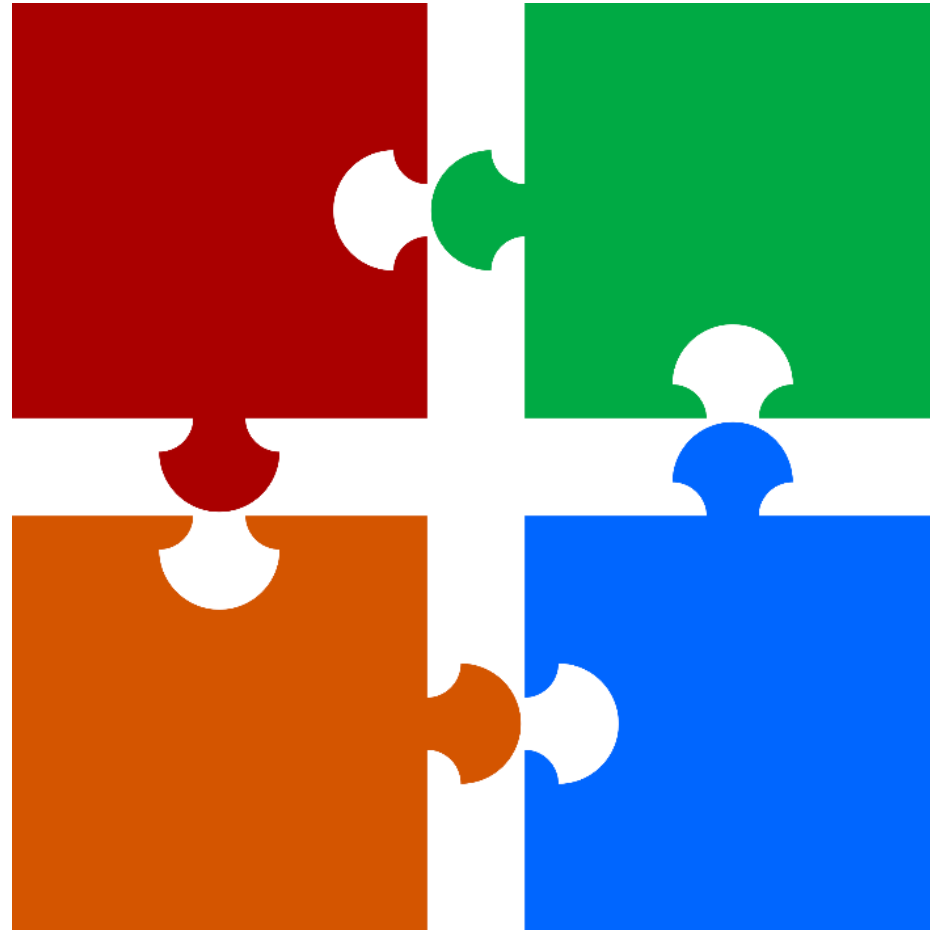


# Simulation Integration



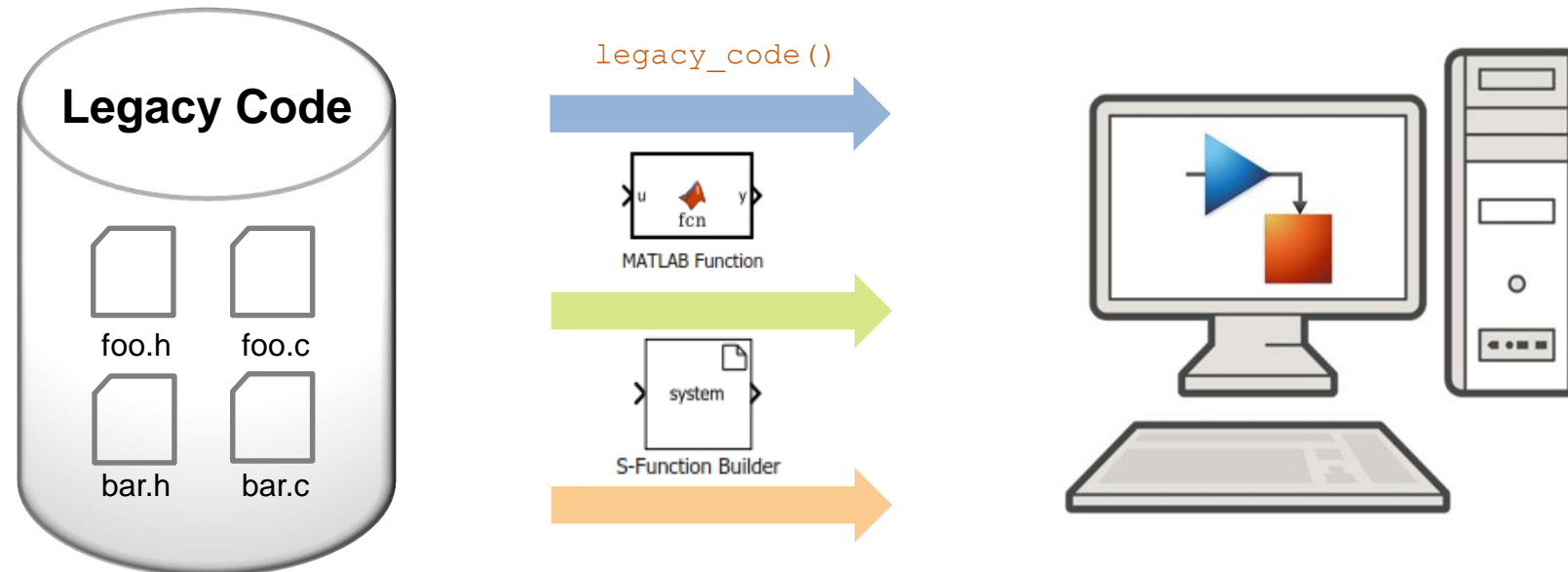
# Disconnected Component Intellectual Property (IP)

- Your IP exists in many forms and in many locations, making integration difficult



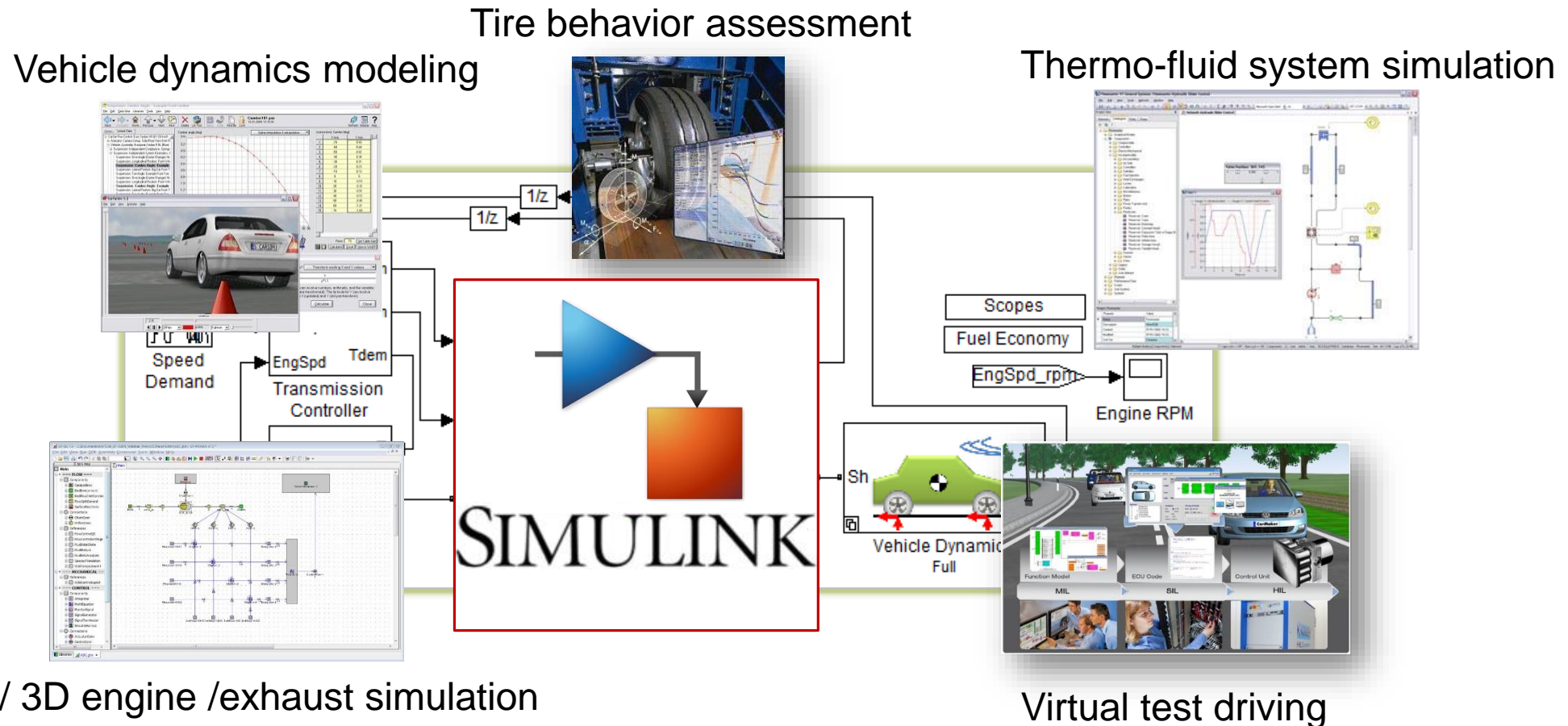
# Integrating Your Code

- Multiple ways to reuse your legacy code with Simulink



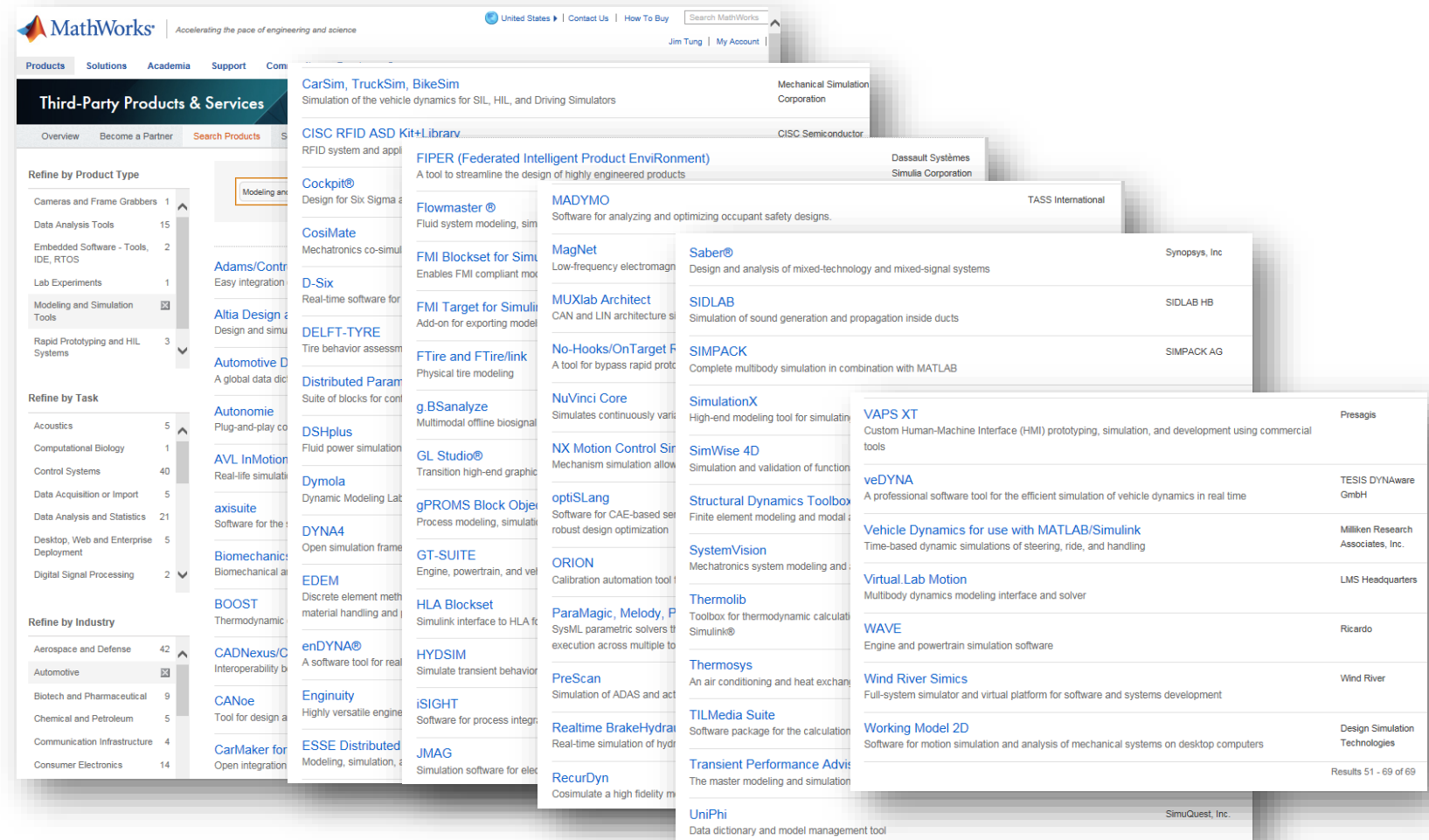
# Integrating Third-Party Simulation Tools

Mature and extensive APIs for third-party tool integration



# Partner Ecosystem

- Numerous partners provide interface to Simulink



# Customer Success in Simulation Integration

Develop Integrated Vehicle Safety Applications

Siddharth D'Silva, Principal Engineer

Autoliv



## Challenge

- Design and validate safety-critical algorithms before implementation

## Solution

- Leverage Simulink as a platform by integrating third-party software

# Customer Success in Simulation Integration

“Seamless integration with third party software solutions enables rigorous development in a safe environment. For application engineers or system engineers, it is very useful that you can export these complex third-party tool functionalities in the form of S-functions and run co-simulation.”

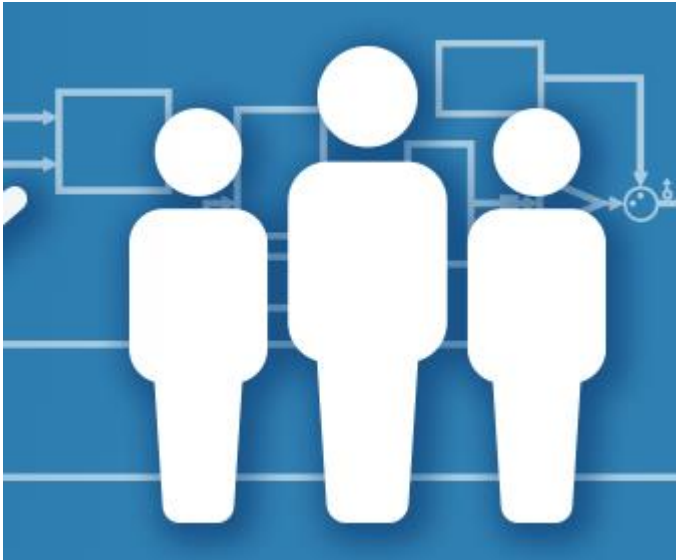
Siddharth D’Silva, Autoliv



## Results

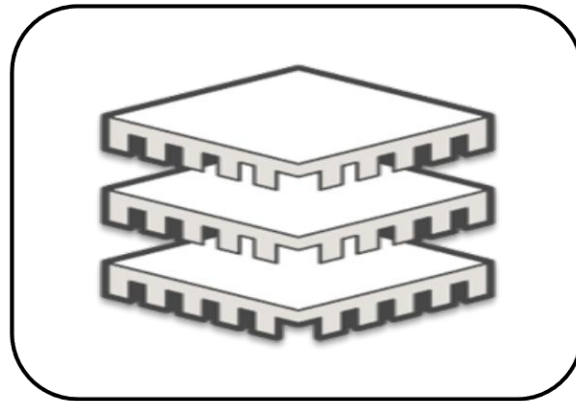
- Industry first integration of stability control inertial sensor into airbag control unit
- Restraint control module software development time reduced by 30%

# Scalability

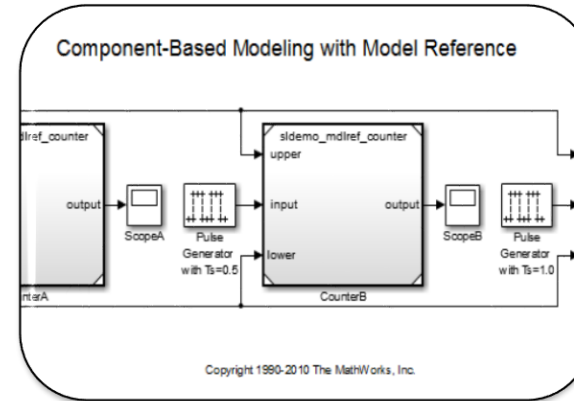




# Scalability Challenges



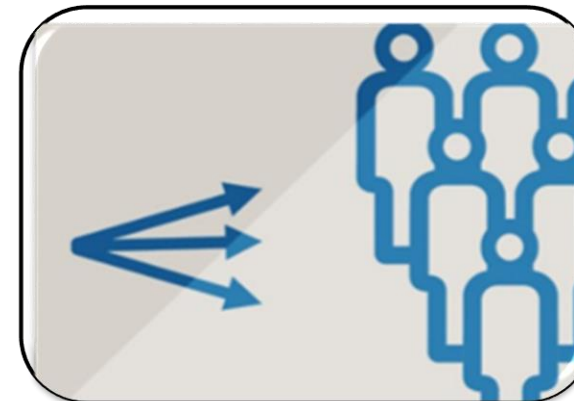
Performance



Componentization



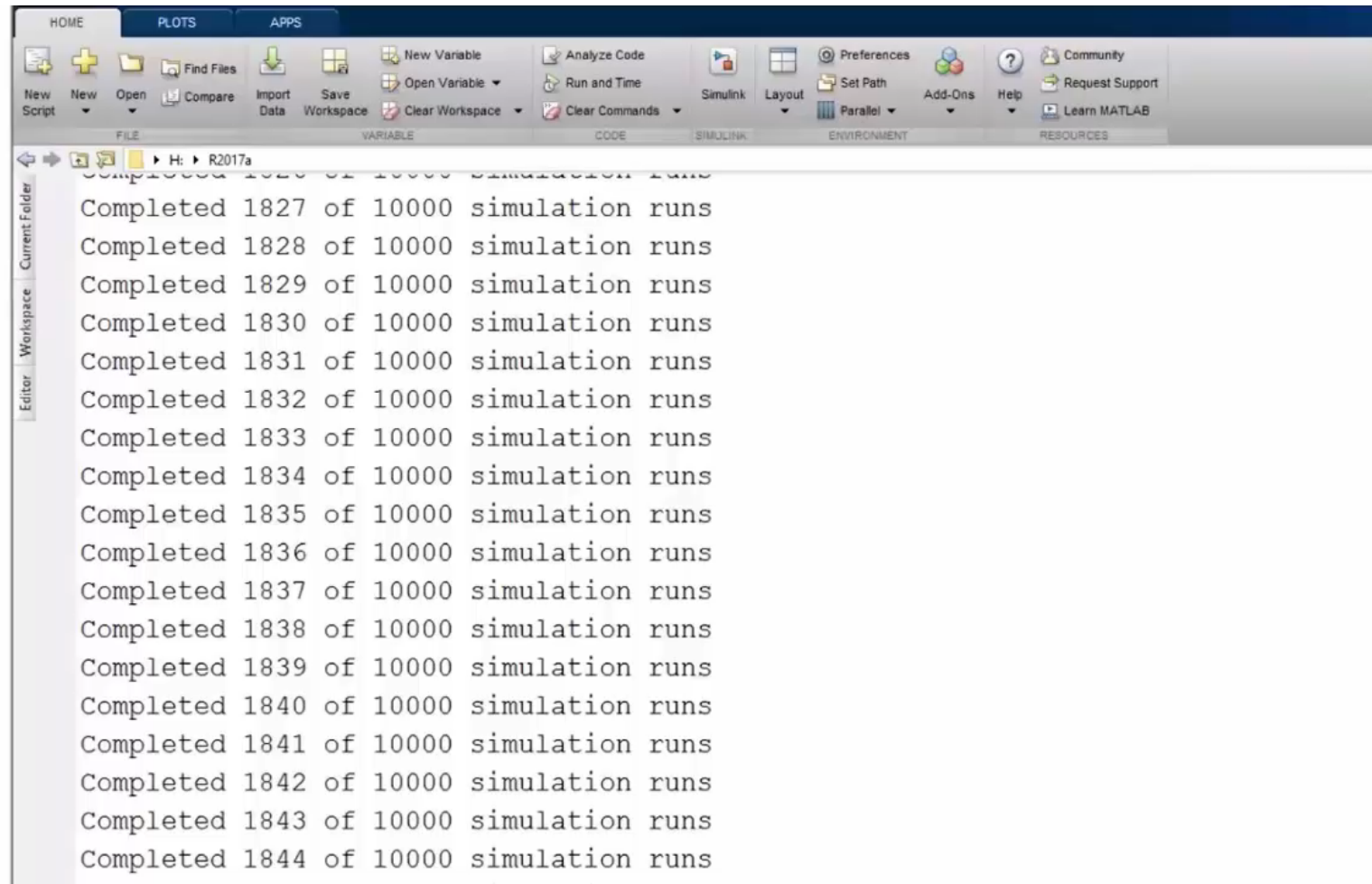
Team Workflows



Sharing

# Performance Scalability

- Easy scalability to multicore or cluster/cloud computation environment

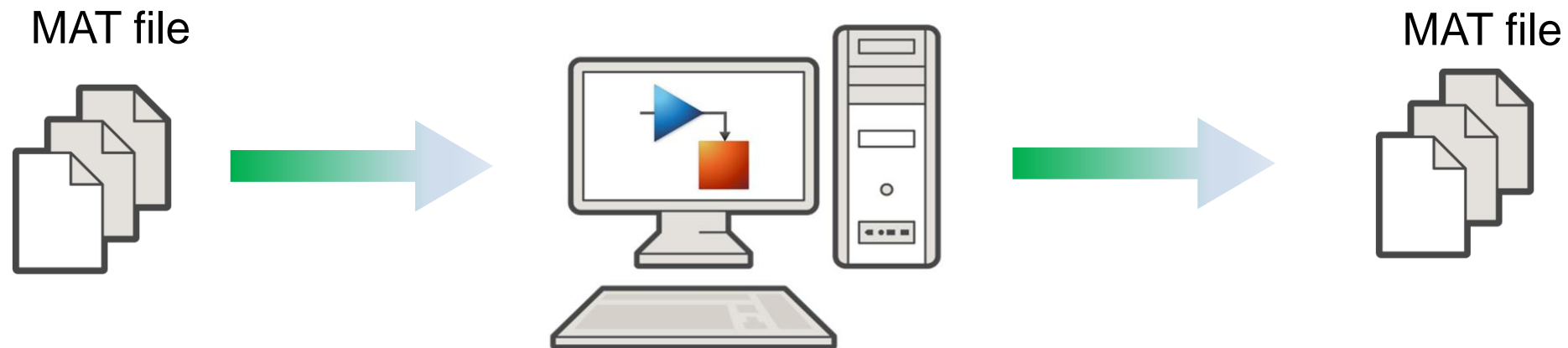


The screenshot displays the MATLAB software interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar is a toolbar with various icons for file operations, workspace management, and simulation control. The main workspace area shows a list of simulation runs, each marked as 'Completed'. The list is organized into columns: 'Completed', 'Run ID', 'Total Runs', and 'Simulation Name'. The runs are numbered sequentially from 1827 to 1844, each representing 1 of 10,000 simulation runs.

Completed	Run ID	Total Runs	Simulation Name
Completed	1827	of 10000	simulation runs
Completed	1828	of 10000	simulation runs
Completed	1829	of 10000	simulation runs
Completed	1830	of 10000	simulation runs
Completed	1831	of 10000	simulation runs
Completed	1832	of 10000	simulation runs
Completed	1833	of 10000	simulation runs
Completed	1834	of 10000	simulation runs
Completed	1835	of 10000	simulation runs
Completed	1836	of 10000	simulation runs
Completed	1837	of 10000	simulation runs
Completed	1838	of 10000	simulation runs
Completed	1839	of 10000	simulation runs
Completed	1840	of 10000	simulation runs
Completed	1841	of 10000	simulation runs
Completed	1842	of 10000	simulation runs
Completed	1843	of 10000	simulation runs
Completed	1844	of 10000	simulation runs

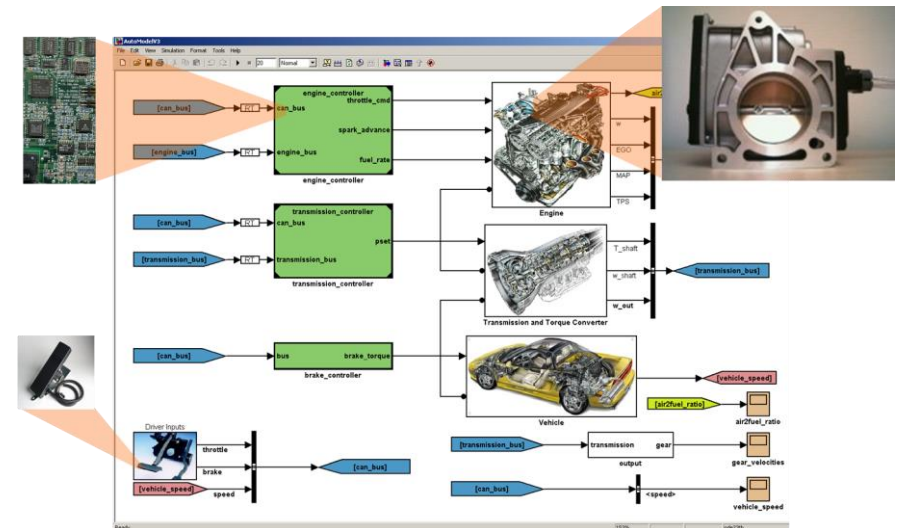
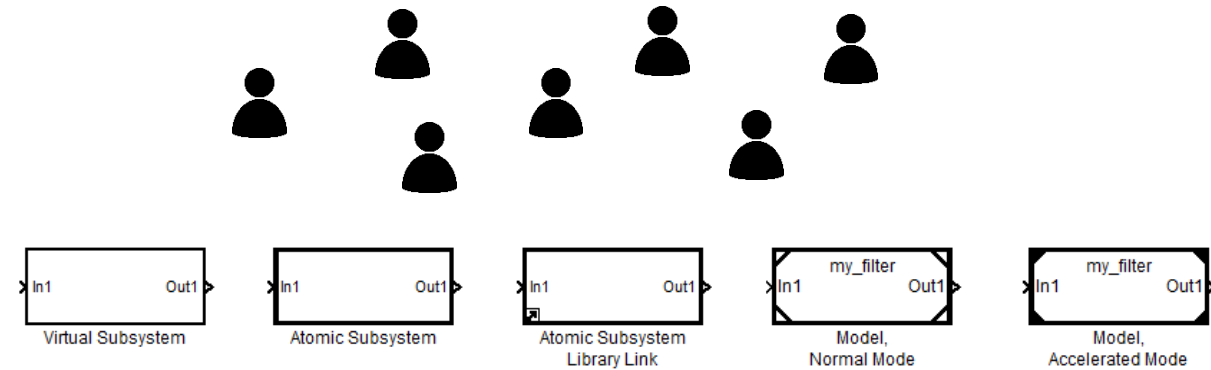
# Performance Scalability

- Big data workflow
  - Processing large amount of simulation inputs / outputs



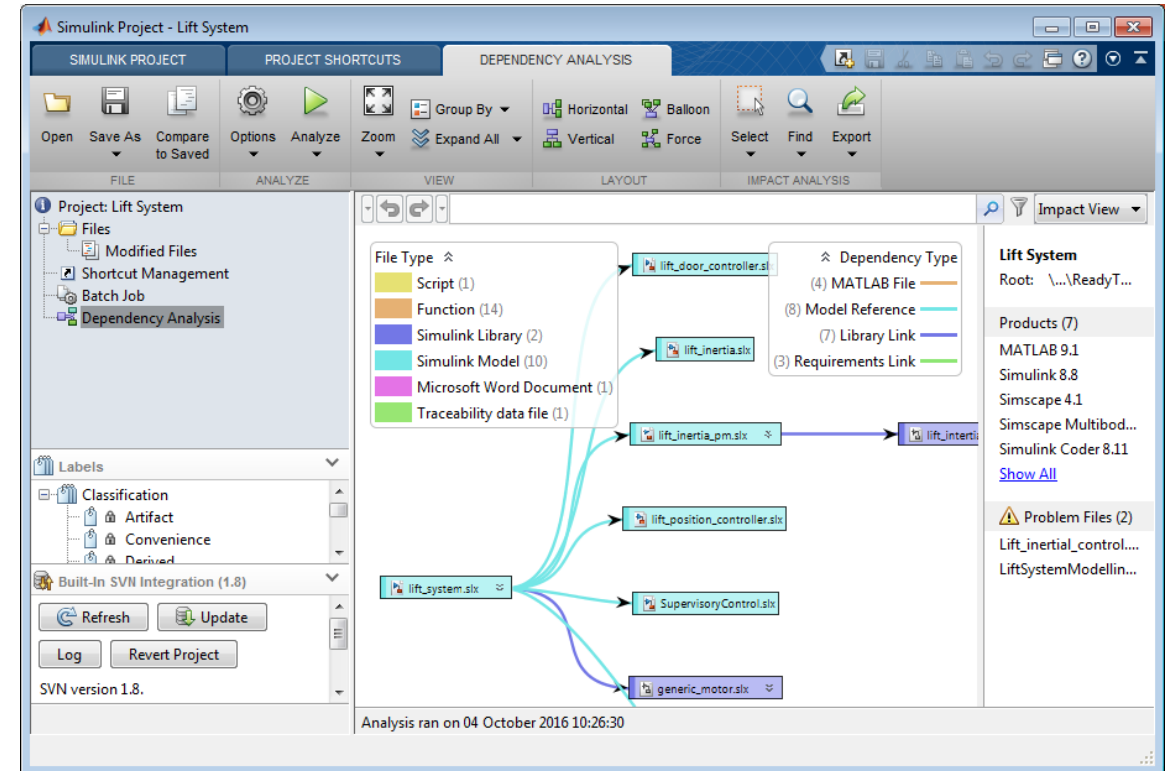
# Complex Design Development through Componentization

- Supporting team workflows
  - Faster modular development
  - More effective verification
  - Increased reusability
  
- Improving performance
  - Incremental loading and code generation
  - Simulation speed
  - Memory usage



# Capabilities Enabling Team Workflows

- Source control
- Design comparison and merging
- Dependency analysis
- Task automation

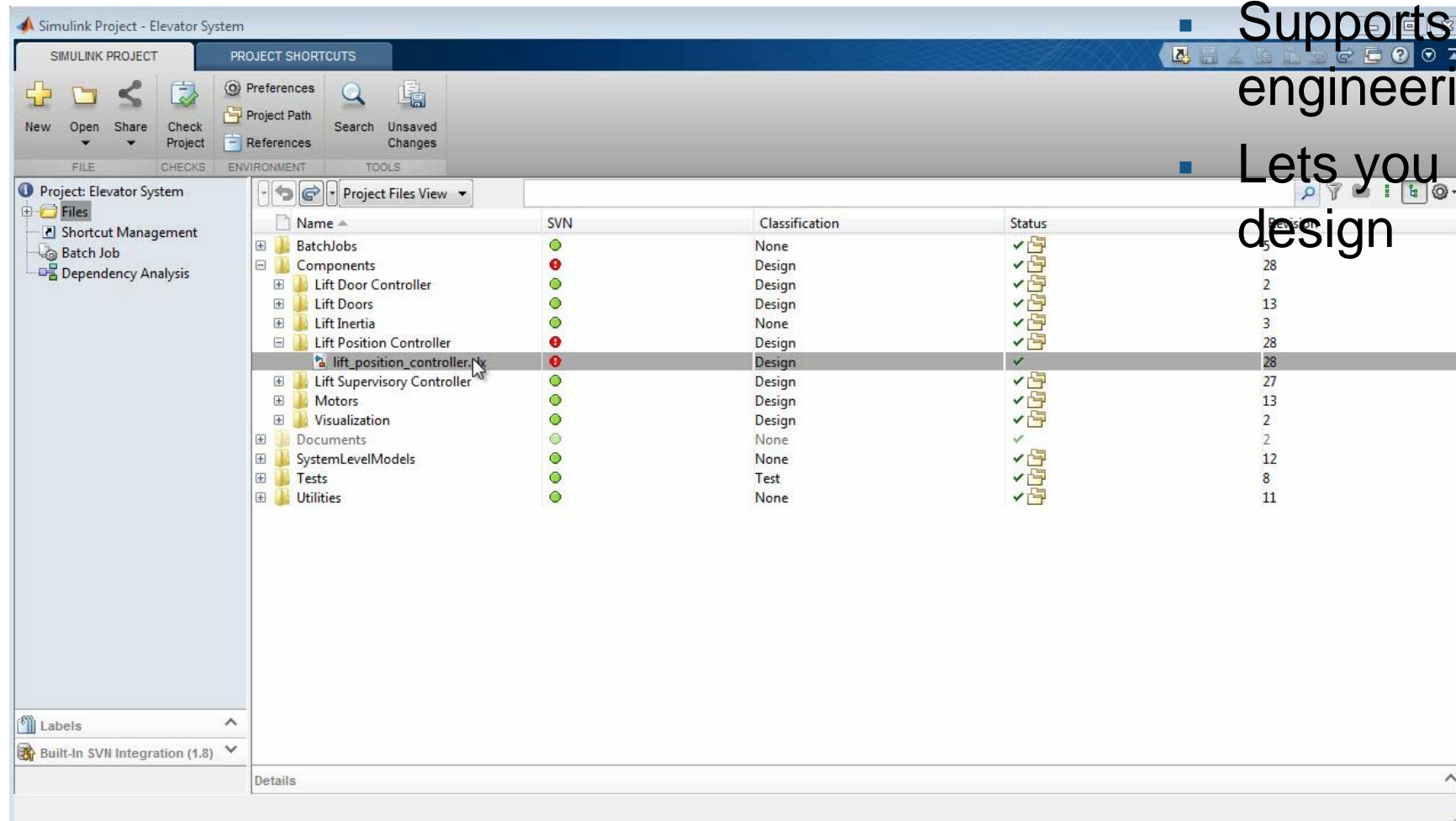


# Source Control Integrations

- Microsoft Team Foundation Server (TFS) integration available now from MathWorks File Exchange



# Integrating Work from Different Engineers via Merge



- Supports concurrent engineering
- Lets you concentrate on design

# Dependency Analysis – Modular Development

The screenshot shows the Simulink Project - Elevator System interface. The main window displays a table of project files and their dependencies. The table has four columns: Name, Path, Status, and Classification. The Status column contains green checkmarks and folder icons, indicating the dependency status of each file. The Classification column lists the type of each file, such as Test, Design, or None.

Name	Path	Status	Classification
Lift	\$\Tests	✓	Test
Lift Door Controller	\$\Components	✓	Design
Lift Door Controller	\$\Tests	✓	Test
Lift Doors	\$\Components	✓	Design
Lift Doors	\$\Tests	✓	Test
Lift Inertia	\$\Components	✓	None
Lift Motor	\$\Tests	✓	Test
Lift Position Controller	\$\Components	✓	Design
Lift Position Controller	\$\Tests	✓	Test
Lift Supervisory Controller	\$\Components	✓	Design
Lift Supervisory Controller	\$\Tests	✓	Test
Motors	\$\Components	✓	Design
SystemLevelModels	\$\	✓	None
Tests	\$\	✓	Test
Utilities	\$\	✓	None
Visualization	\$\Components	✓	Design
Visualization	\$\Tests	✓	Test
basic_animation.slx	\$\Components\Visualization	✓	Design
ElevatorTemplate.sltx	\$\Utilities	✓	Other
exportToR2016a.m	\$\BatchJobs	✓	Design
generateBillOfMaterials.m	\$\BatchJobs	✓	Design
generateICD.m	\$\Utilities	✓	Design
generic_motor.slx	\$\Components\Motors	✓	Design
history.m	\$\Utilities	✓	Design
lift_door.req	\$\Components\Lift Doors	✓	Design
lift_door.slx	\$\Components\Lift Doors	✓	Design
lift_door_controller.slx	\$\Components\Lift Door Controller	✓	Design



# Dependency Analysis – Modular Development

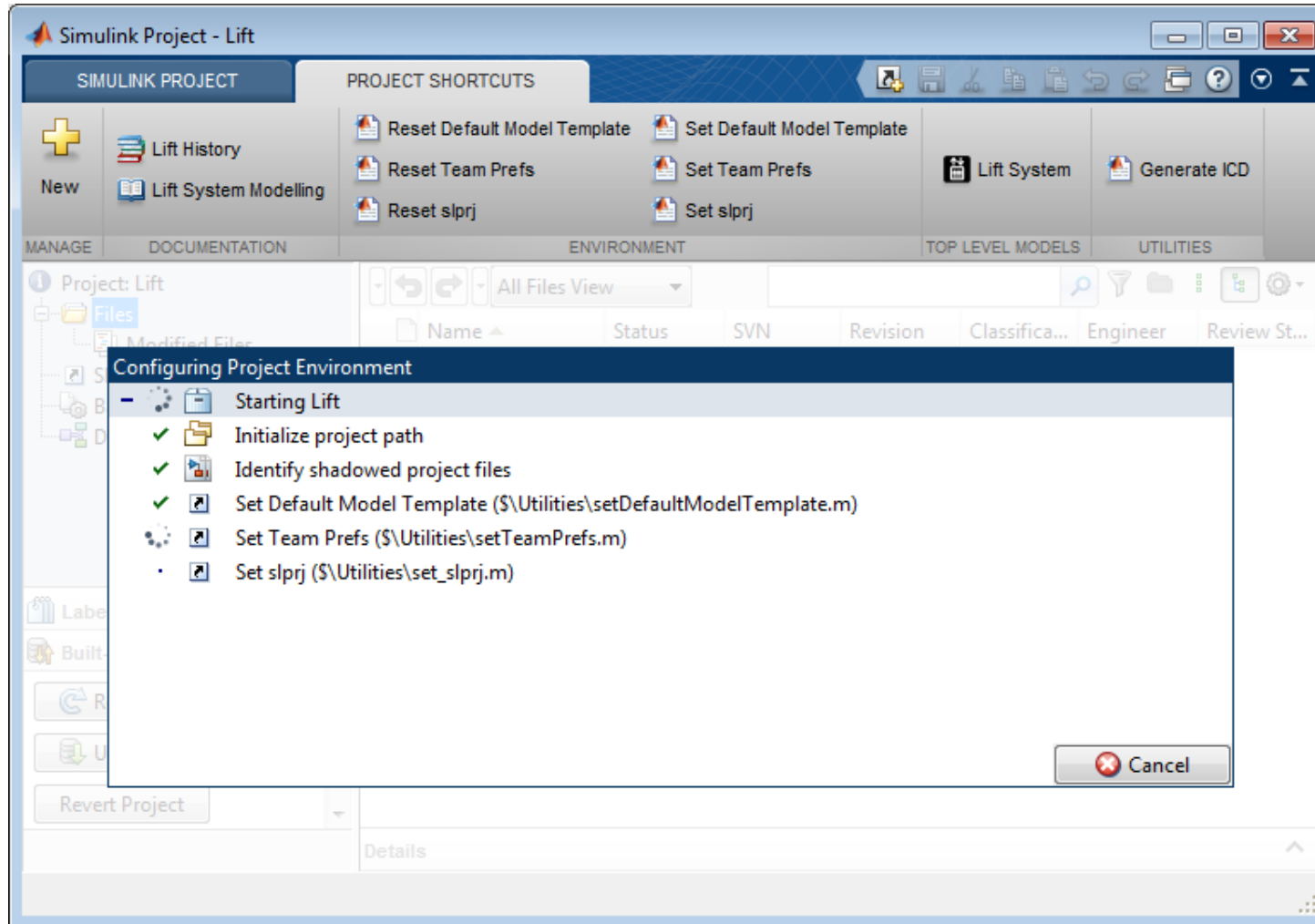
The screenshot displays the Simulink Dependency Analysis interface. The main workspace shows a dependency graph where the root node is 'lift\_system.slx'. It branches into several sub-modules: 'generic\_motor.slx', 'lift\_door\_controller.slx', 'lift\_inertia.slx', 'lift\_inertia\_pm.slx', and 'lift\_position\_controller.slx'. 'lift\_inertia\_pm.slx' has a further dependency on 'lift\_inertia\_utils.slx'. A warning icon is visible on the 'LiftSystemModelling.docx' node. The right-hand sidebar provides a list of products used by the model, including MATLAB 9.1, Simulink 8.8, Simscape 4.1, Simscape Multibody 4.9, and Simulink Coder 8.11. Below this, it indicates one problem file: 'LiftSystemModelling.docx'.

**List products required**

**Show model structure**

**Highlight issues**

# Task Automation – Configuring Project Environment



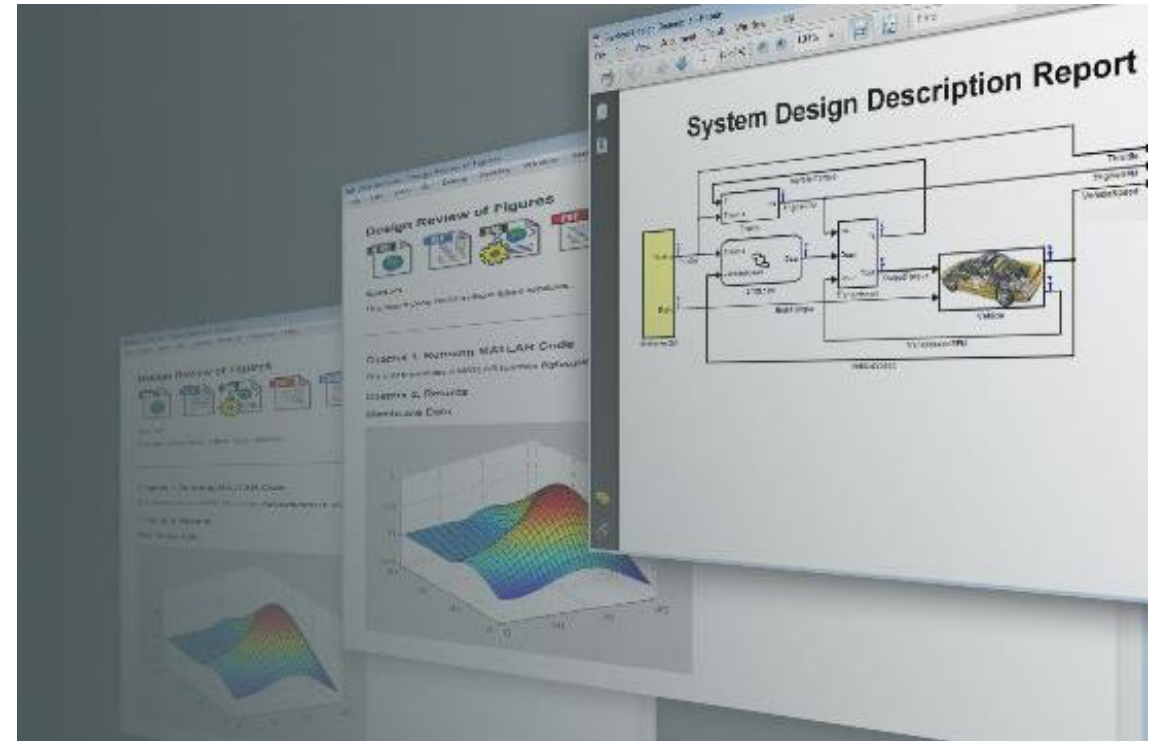
- Robustly configure the team environment
- For everyone
- Automatically

# Sharing Outside Your Team

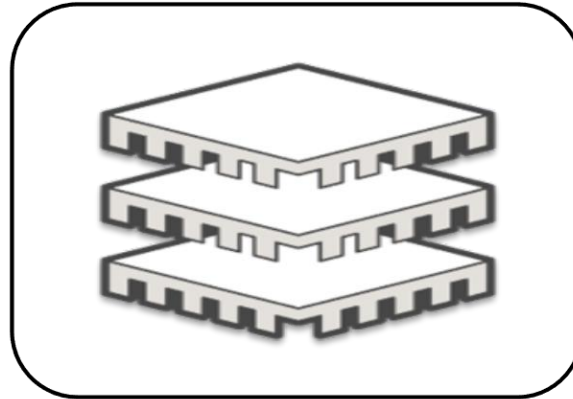
Quick File  
Packaging

Model Protection  
(IP Management)

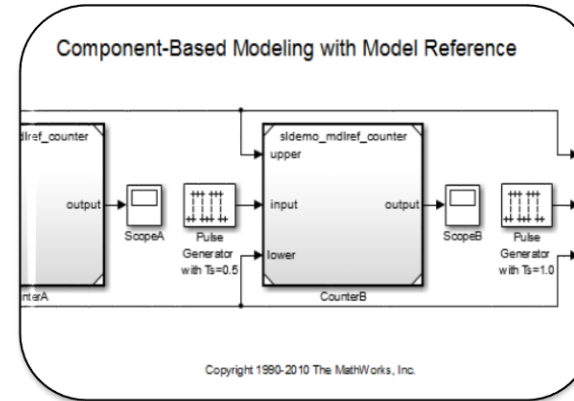
Reporting and  
Documentation



# Simulink Addressing Scalability Challenges



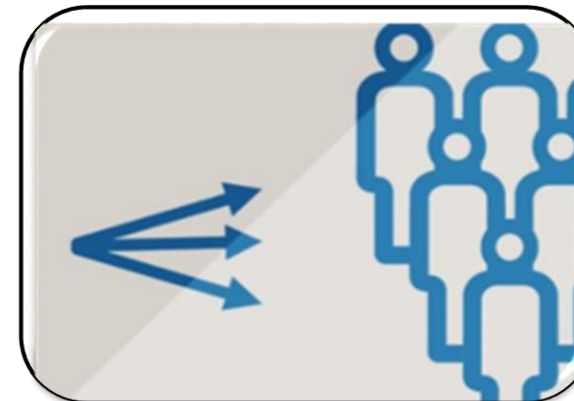
Performance



Componentization



Team Workflows



Sharing

# Simulink as Enterprise Simulation Platform

*“There is no such tool, which gives the simulation environment as well as the hardware verification and validation. In a single environment, I am getting these together. **That is why I use MATLAB and Simulink.**”*

Dr. Deepak Mishra,  
Indian Space Research Organization

