Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

12.24196
Introduction to Embedded Systems

Prof. Dr.-Ing. Stefan Kowalewski | Julius Kahle, M. Sc.

Summer Semester 2025

Part 6
Model-based Development & Simulink

Agenda

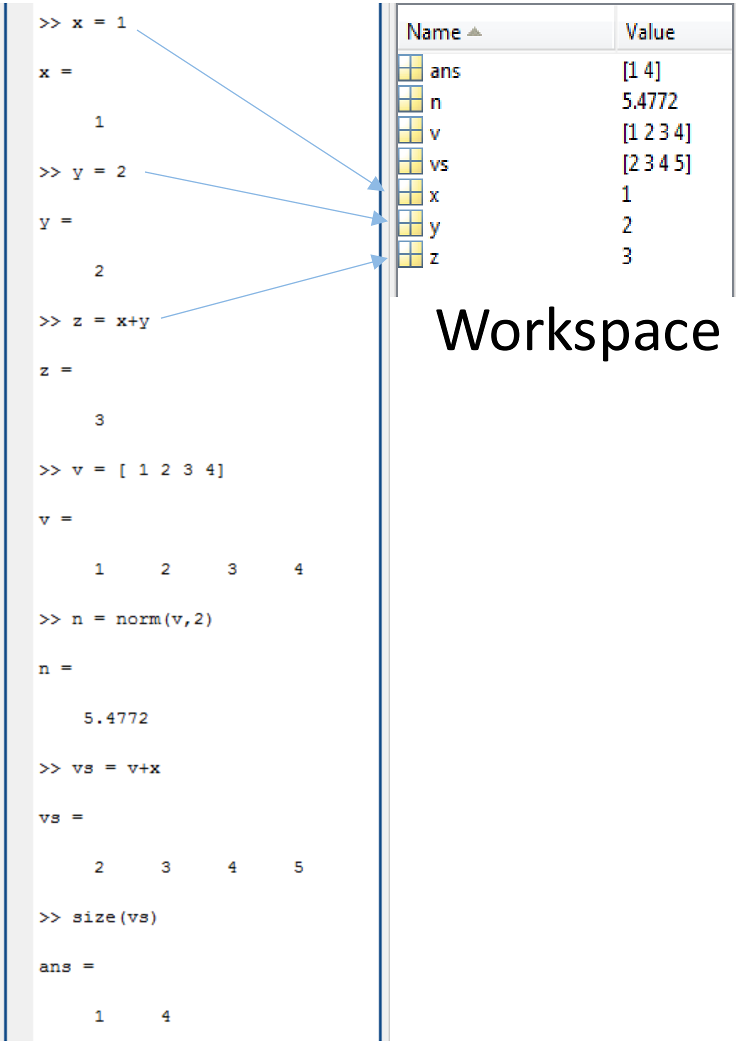
IES Part 8 – Embedded Software Development using Simulink

I. Simulink Fundamentals

II. Model-Based Software Development Process

Matlab

- MATLAB
  - Math-language
    - Everything is a matrix
  - Invented by Cleve Moler
  - Math-tool
  - Distributed by TheMathworks
  - Variety of extensions
    - Optimization
    - Statistics
    - Fuzzy Logic
    - Image Processing
    - Physical Modelling
    - Finances
    - **Simulink**



```
>> x = 1

x =

     1

>> y = 2

y =

     2

>> z = x+y

z =

     3

>> v = [ 1 2 3 4]

v =

     1     2     3     4

>> n = norm(v,2)

n =

    5.4772

>> vs = v+x

vs =

     2     3     4     5

>> size(vs)

ans =

     1     4
```

Name	Value
ans	[1 4]
n	5.4772
v	[1 2 3 4]
vs	[2 3 4 5]
x	1
y	2
z	3

Workspace

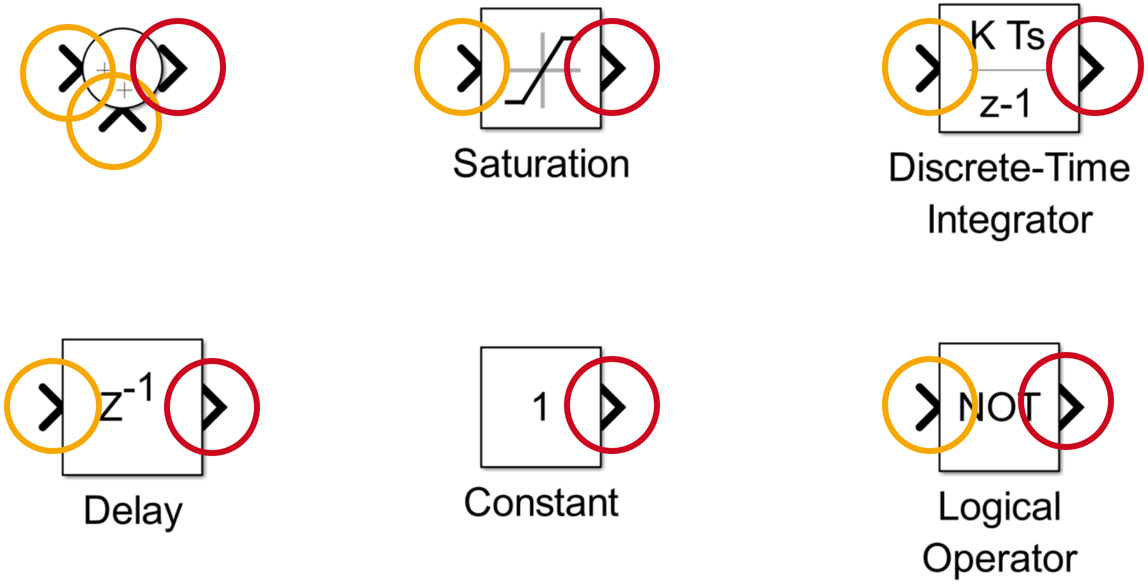
Command-Window

Simulink Blocks

- Simulink is a tool to model and simulate „dynamic systems“
- Operations are expressed by blocks
- Similar to Function Block language (PLC)

Inputs

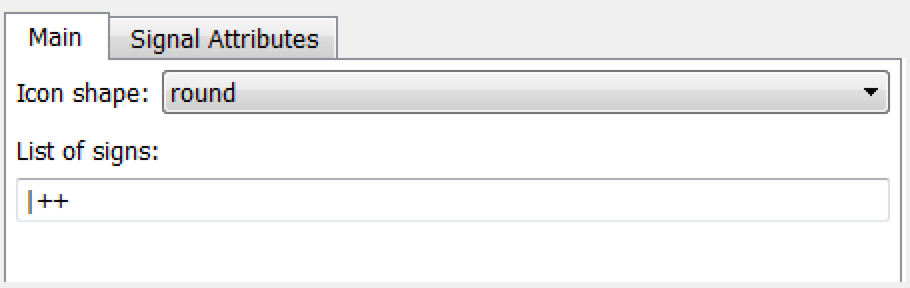
Outputs



Blocks realize functions

Simulink Block Parameters

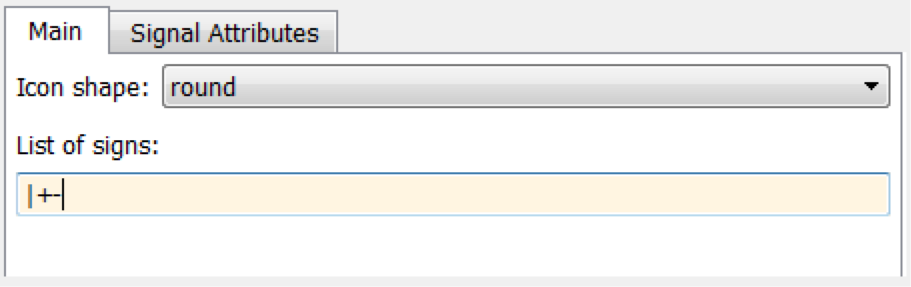
- ▶ Block parameters can be configured



Icon shape: round

List of signs:

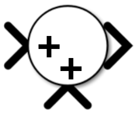
|++

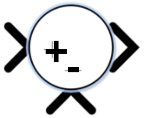


Icon shape: round

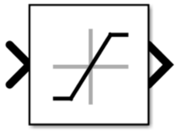
List of signs:

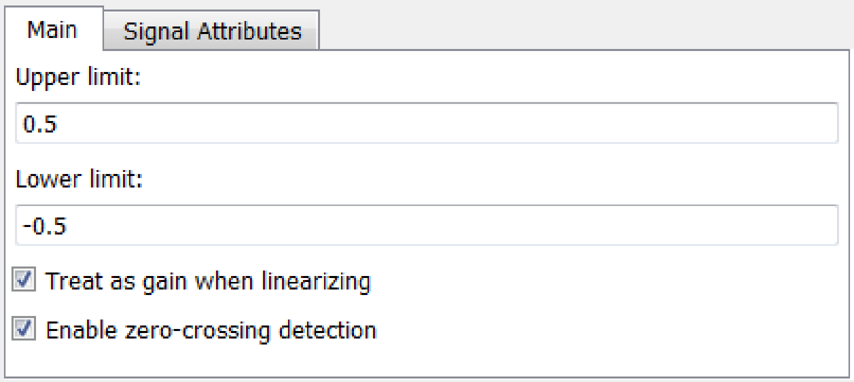
|+-





- ▶ Available parameters depend on block type


Saturation



Upper limit:

0.5

Lower limit:

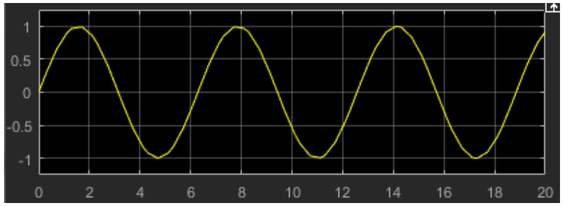
-0.5

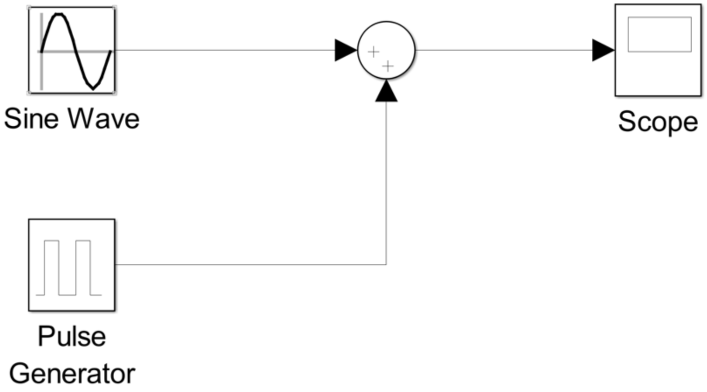
☑ Treat as gain when linearizing

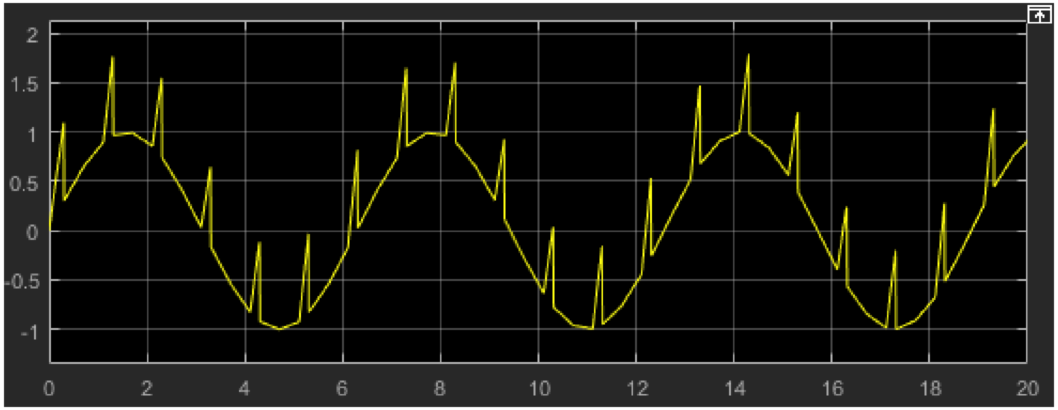
☑ Enable zero-crossing detection

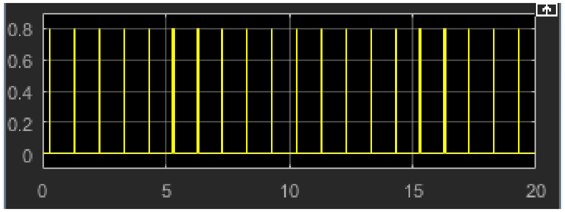
Signals

- Simulink blocks generate/process signals









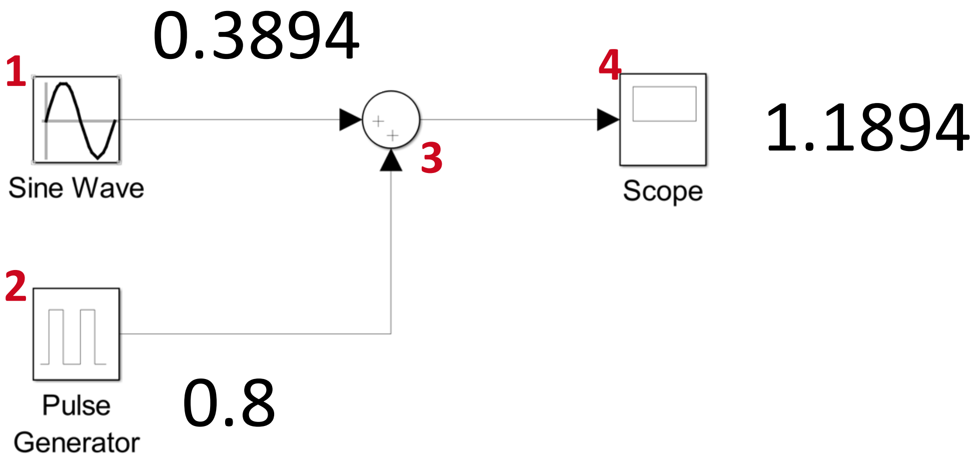
Signals are variables (values change over time)

Sampling

- Signals and Simulink simulation are sample-based
- Blocks are executed according to an execution order

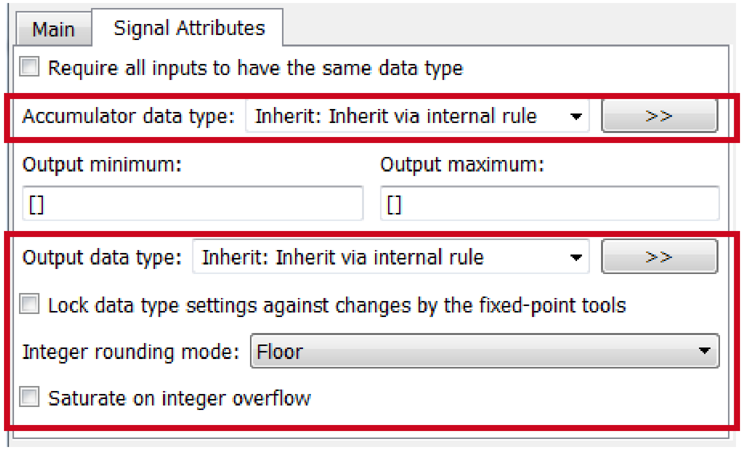
Configured Step-size: 0.1

Sampling time: 0.4



Signal Properties

- Basic Signal Properties
  - Dimension
  - Datatype
    - logical(boolean)
    - int8, int16, int32
    - uint8, uint16, uint32
    - float32(single), float64(double)
    - fixed-point data types
- Signal properties are configurable by some blocks, e.g. Sum block



□ Require all inputs to have the same data type

Accumulator data type: Inherit: Inherit via internal rule

Output minimum: []

Output maximum: []

Output data type: Inherit: Inherit via internal rule

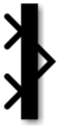
□ Lock data type settings against changes by the fixed-point tools

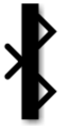
Integer rounding mode: Floor

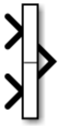
□ Saturate on integer overflow

Signal Property Influencing Blocks

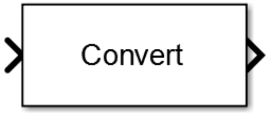
- There are special blocks to influence signal properties
  - Mux (combine signals to vectors)
  - Demux (split vectors)
  - Vector Concatenate
  - Data Type Conversion







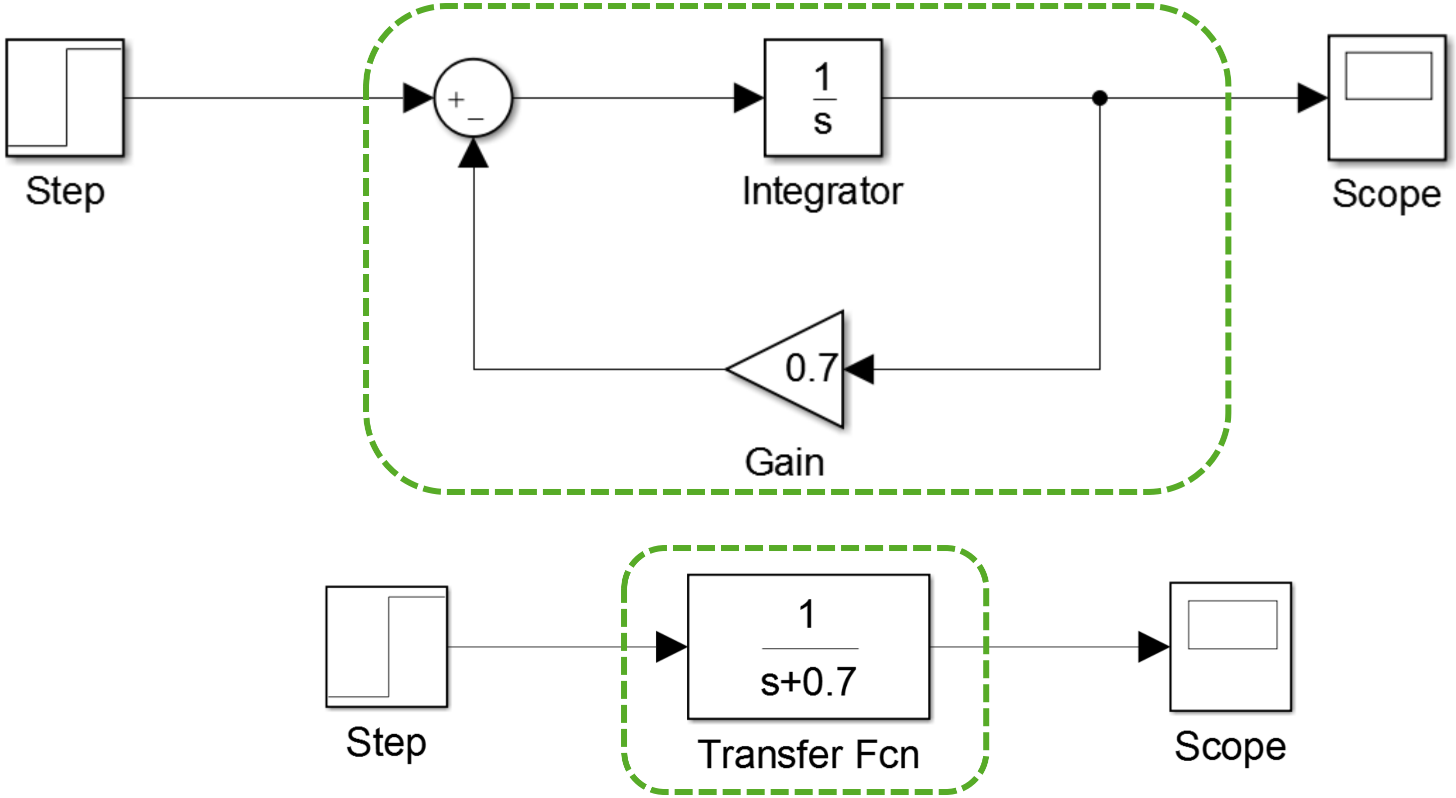
Vector
Concatenate



Data Type Conversion

Modeling Possibilities

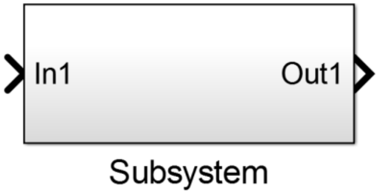
- Example – Low Pass Filter

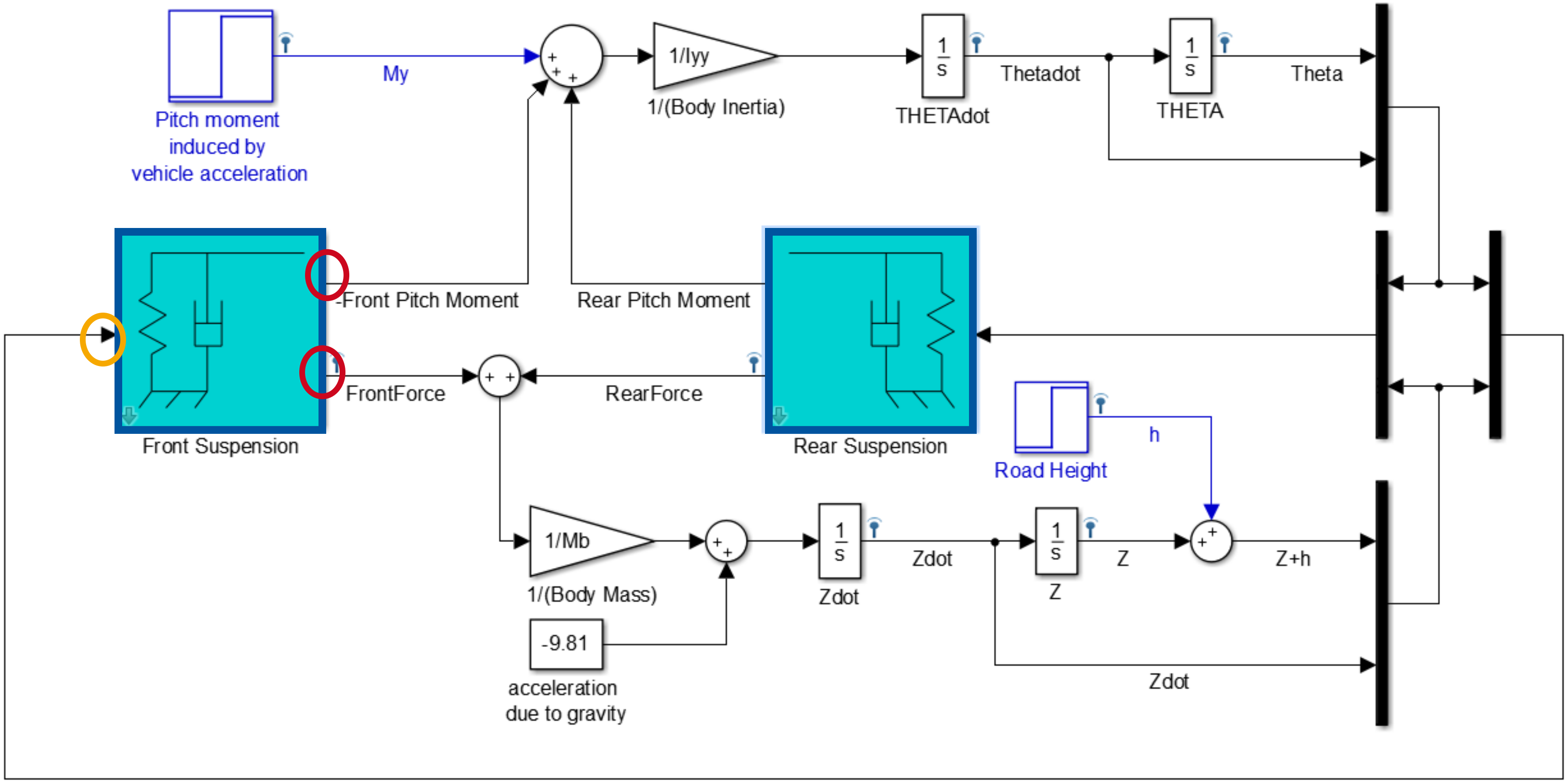


- Different possibilities to model a behavior

Subsystems

- Models can be hierarchically composed using Subsystem-blocks

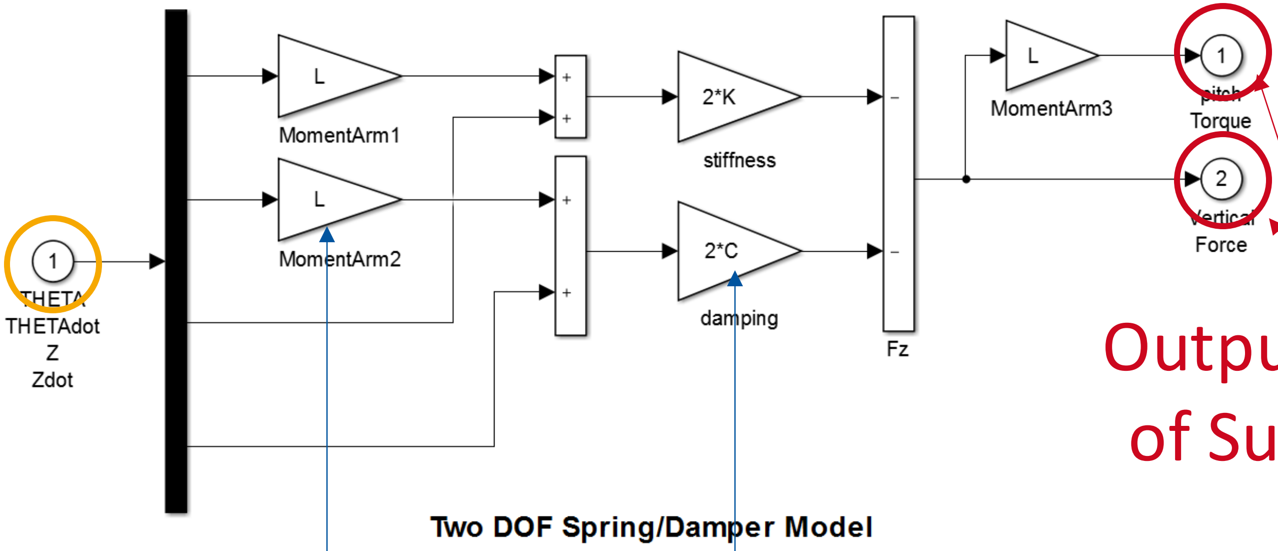




Copyright 1990-2015 The MathWorks, Inc.

Subsystem Implementation

- Front Suspension Subsystem implementation



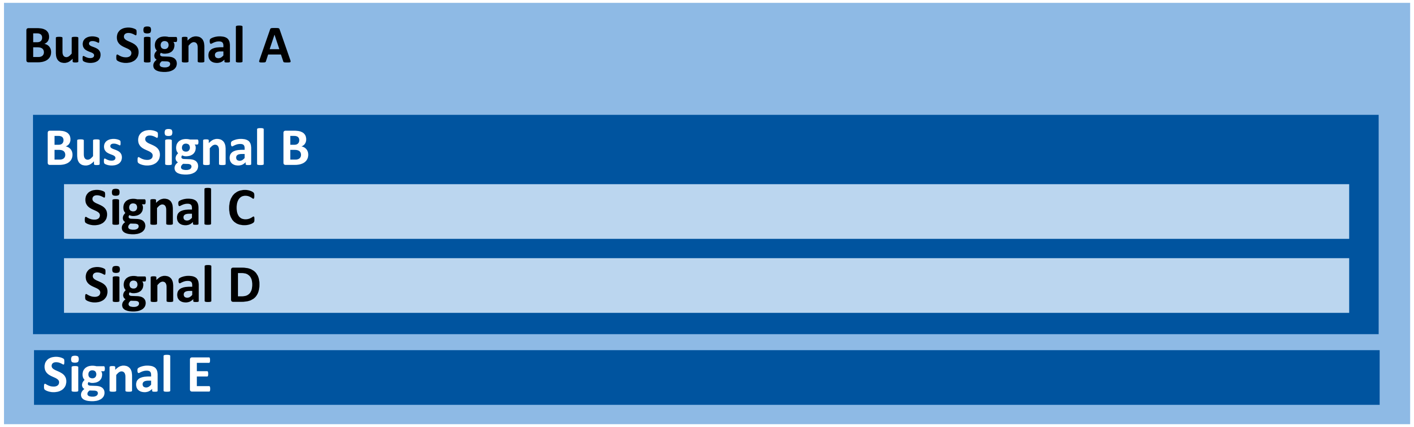
Input signal to Subsystem

MATLAB Workspace Constants

Output signals of Subsystem

Abstraction Principles

- Subsystems structure model
- Subsystems can be
  - Reused
  - Parametrized
  - Extracted to Libraries
- Signals can be hierarchically grouped (Bus-signals)



Bus Signal A

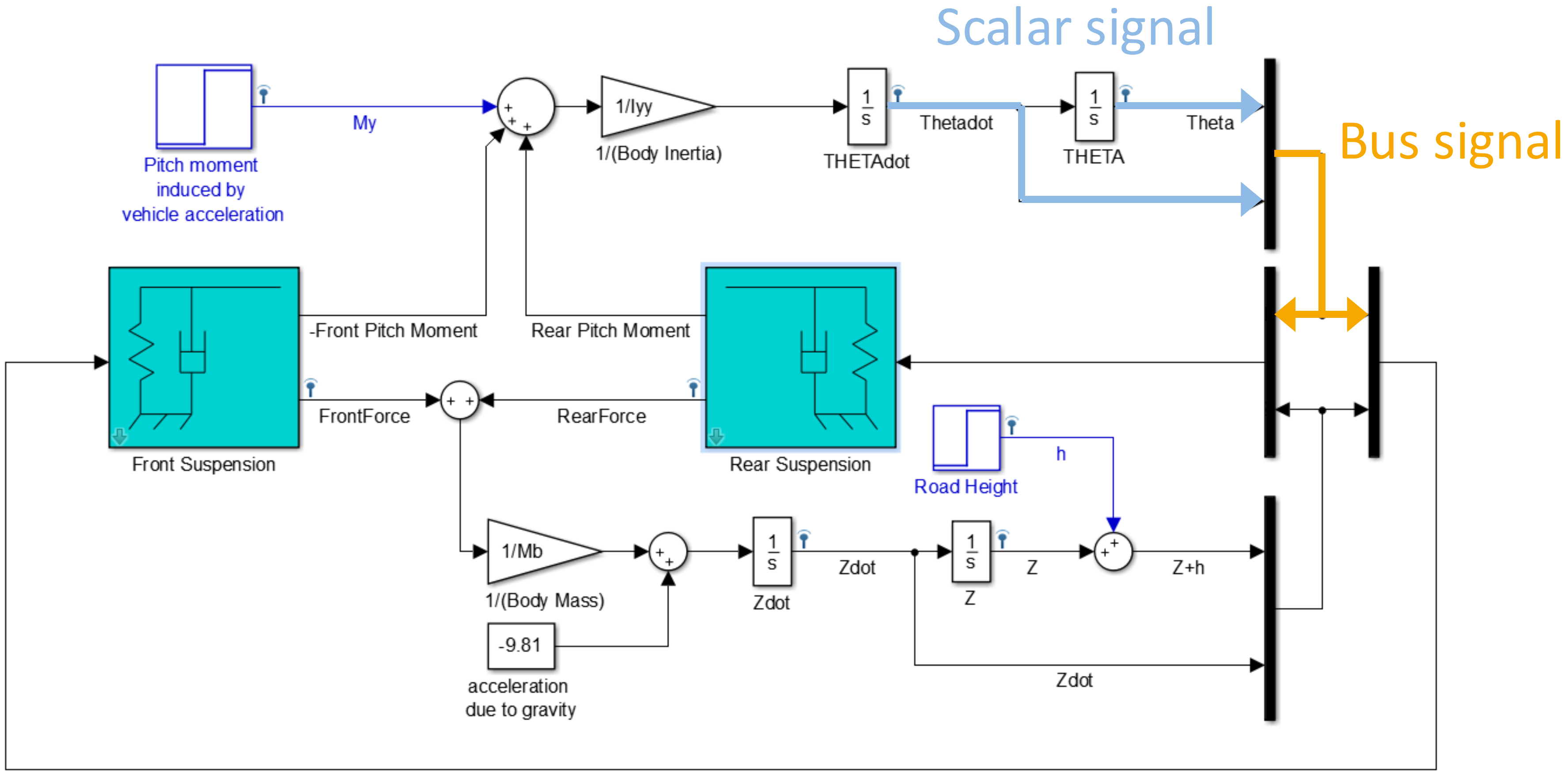
Bus Signal B

Signal C

Signal D

Signal E

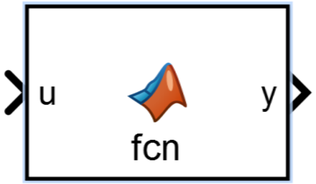
Bus Signals



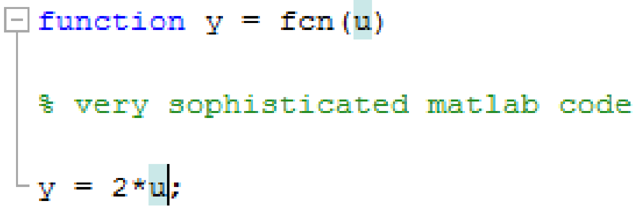
Copyright 1990-2015 The MathWorks, Inc.

Advanced Simulink Blocks

- Extension by mixture with other modelling/programming paradigms
  - Mfunctions
    - Write MATLAB programs within Simulink



MATLAB Function

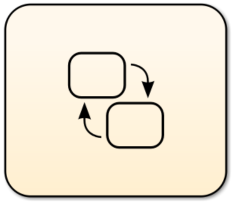


```
function y = fcn(u)

% very sophisticated matlab code

y = 2*u;
```

  - Sfunctions
    - Call (compiled) code from Simulink

S-Function

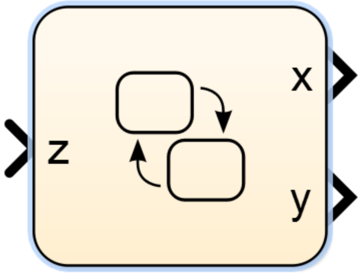
  - Stateflow (Toolbox)
    - Model control structure as state chart



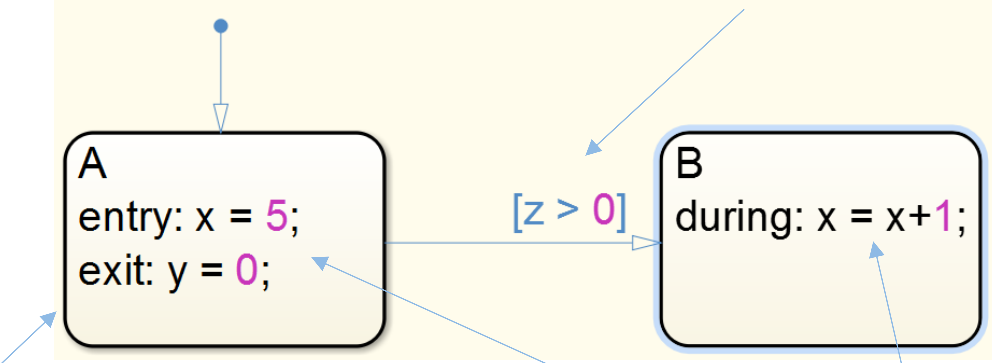
Chart

Stateflow

- Stateflow example



Chart



Transition guard

Initial state

Actions

- Entry: actions performed when entering a state
- During: actions performed when remaining in a state for the considered sampling time
- Exit: actions performed when exiting a state

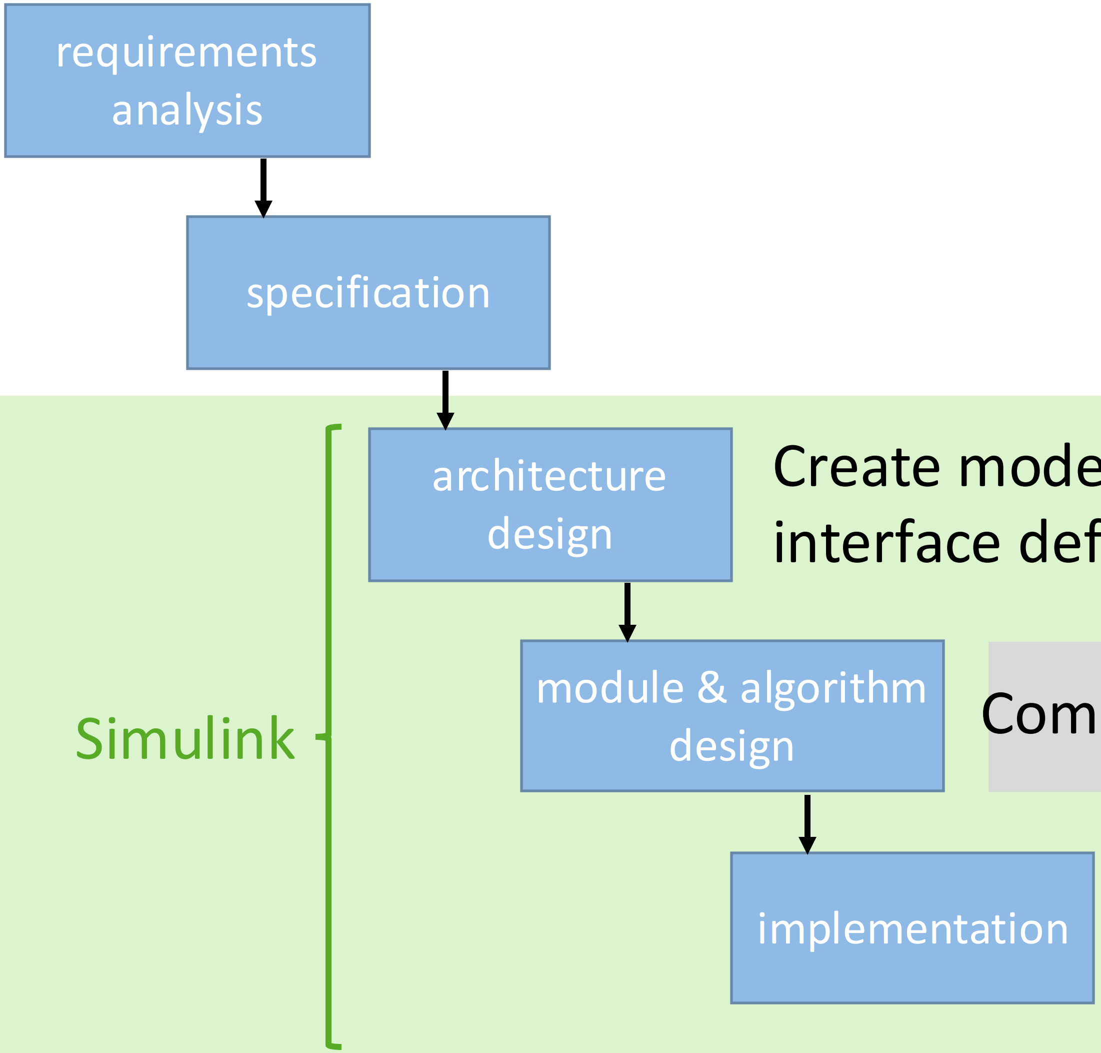
Agenda

IES Part 8 – Embedded Software Development using Simulink

I. Simulink Fundamentals

II. Model-Based Software Development Process

Simulink in Context



Create model architecture, interface definition

Complete subsystem models

Generate code from model

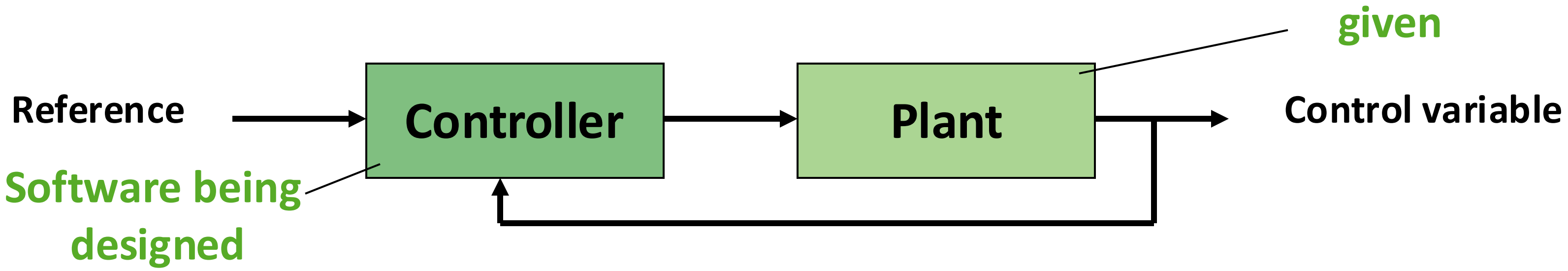
Code Generation

- Multiple code generators available
  - PLC Coder (Mathworks)
    - Structured Text
    - Ladder Diagram
  - Embedded Coder (Mathworks)
    - C code
    - C++ code
  - TargetLink (dSpace)
    - C code
    - But: requires use of TargetLink blockset

- Code generators vary in
  - Configuration possibilities
  - Quality regarding generated code

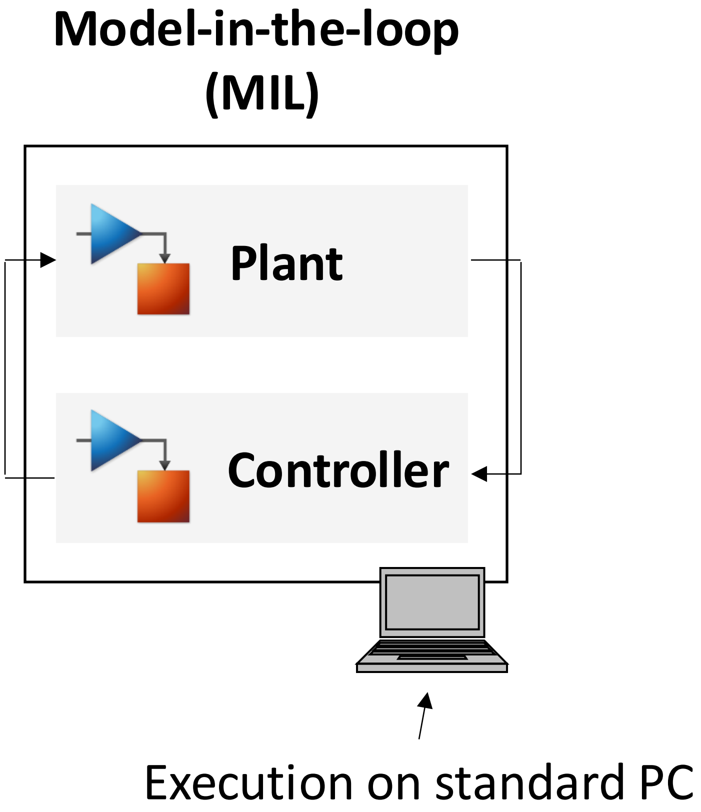
Rapid Prototyping

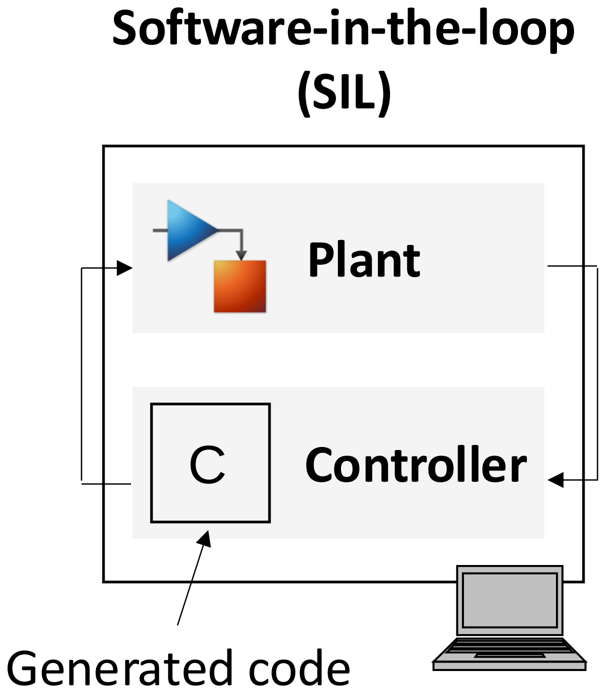
- Encourages rapid prototyping
  - Models are executable from the beginning
  - Easy integration of controllers with plant models



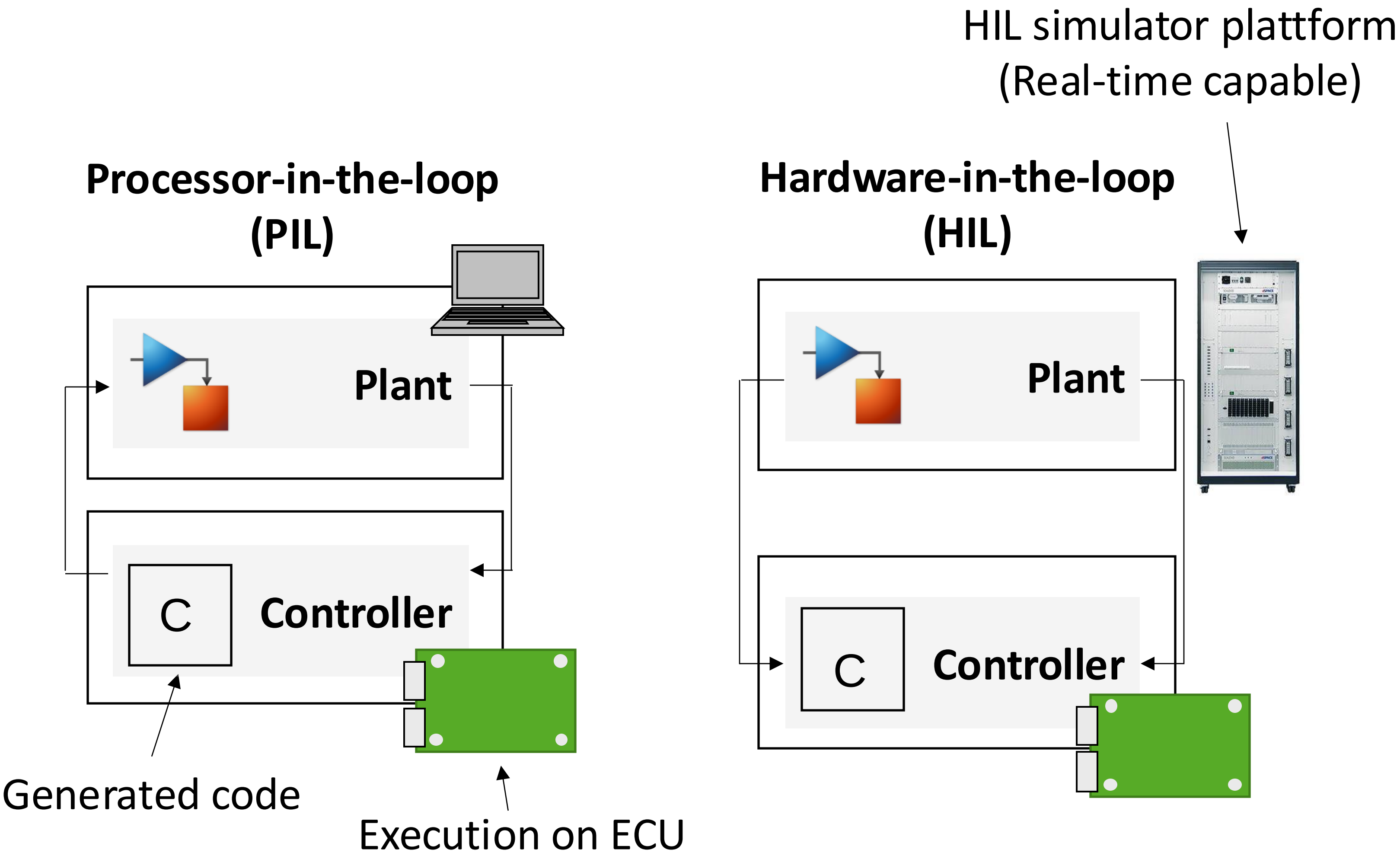
  - Code can be generated at any time
  - Early testing possible

MIL and SIL Tests





PIL and HIL Tests


HIL simulator plattform
(Real-time capable)
Processor-in-the-loop
(PIL)
Plant
C
Controller
Generated code
Execution on ECU
Hardware-in-the-loop
(HIL)
Plant
C
Controller