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

Informatik 11
Embedded Software

RWTHAACHEN
UNIVERSITY

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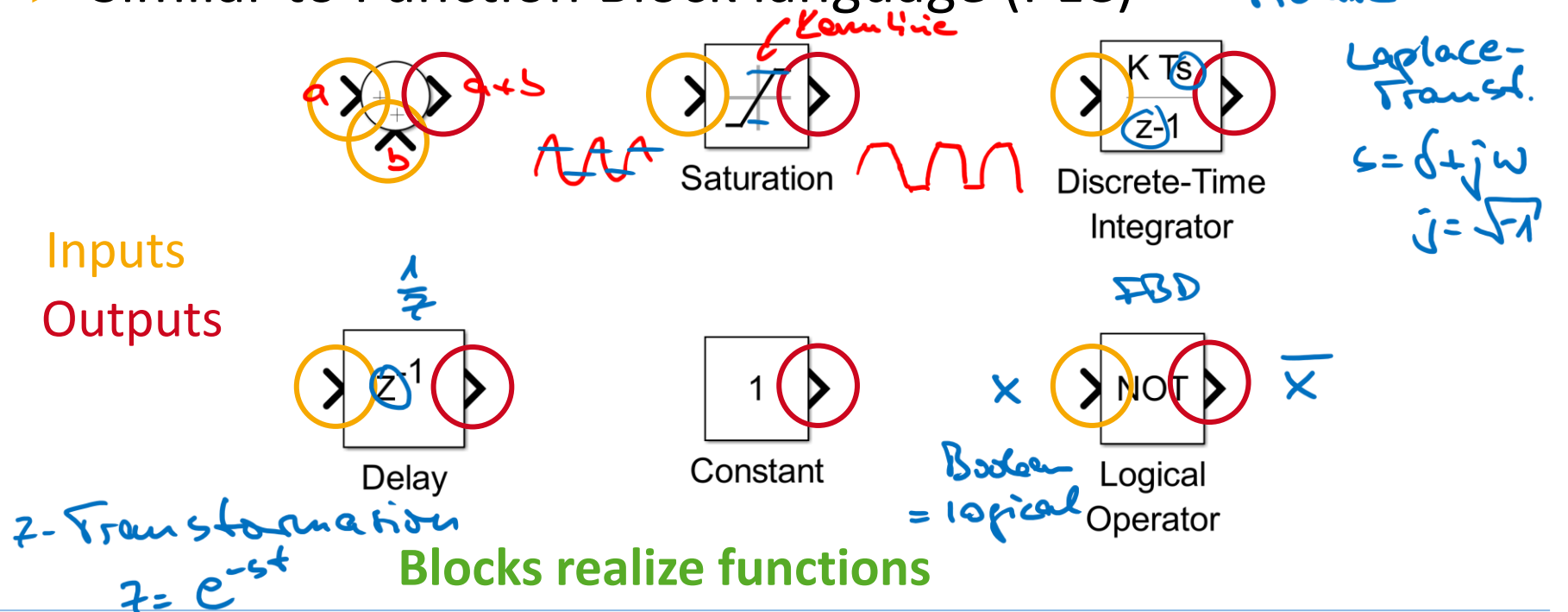
Name ▲	Value
ans	[1 4]
n	5.4772
v	[1 2 3 4]
vs	[2 3 4 5]
x	1
y	2
z	3

Workspace

Command-Window

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

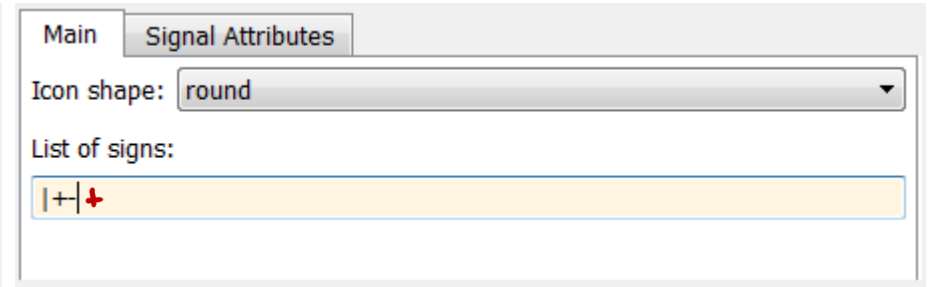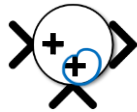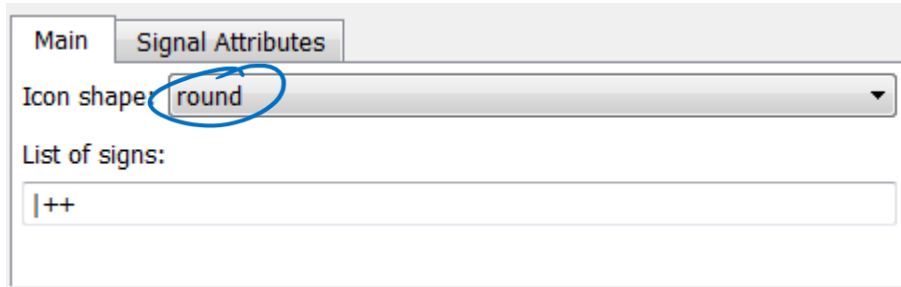# Simulink Blocks

► 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

(handwritten annotations on blocks:)
- a, b, a+b (Sum block)
- Kennlinie — Saturation
- K Ts / (z-1) — Discrete-Time Integrator
- Übertragungsfkt. / Transferfunctions
- Laplace-Transf.
- $s = \delta + j\omega$
- $j = \sqrt{-1}$
- $\frac{1}{z}$ — z-1 — Delay
- Z-Transformation $z = e^{-st}$
- 1 — Constant
- FBD — NOT — Logical Operator
- x, $\overline{x}$
- Boolean = logical

4

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
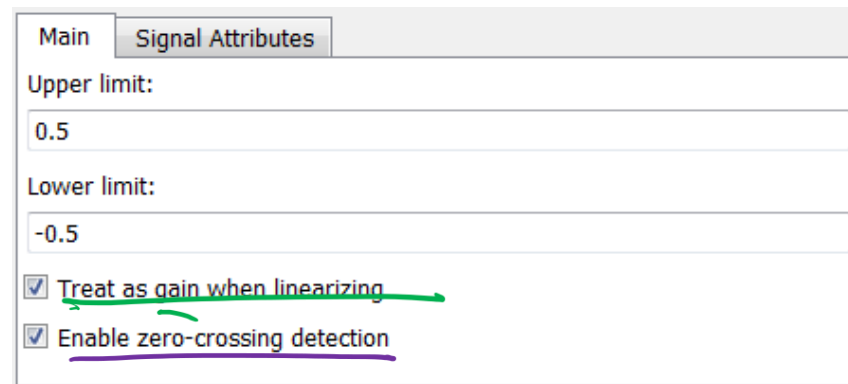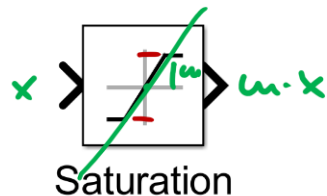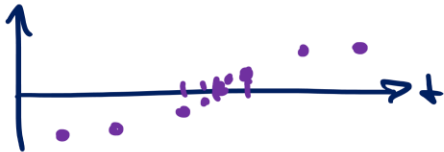Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY
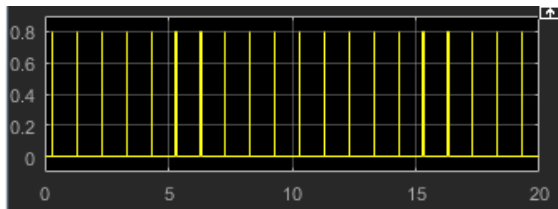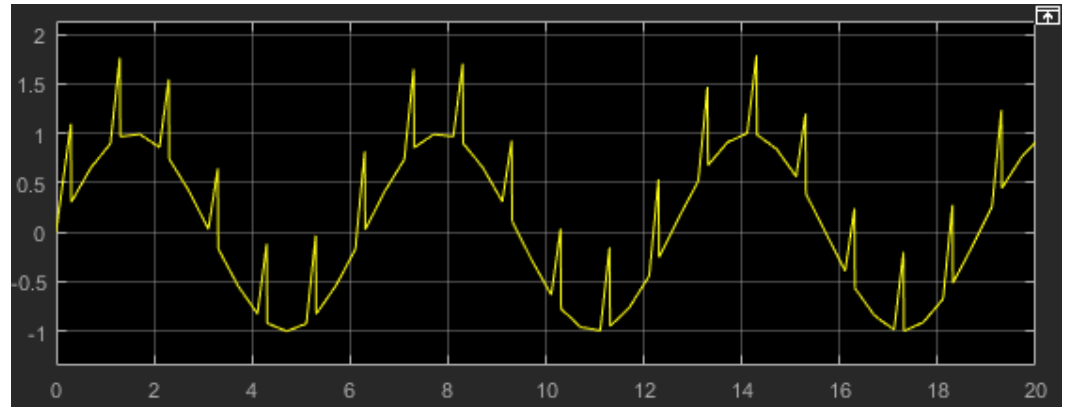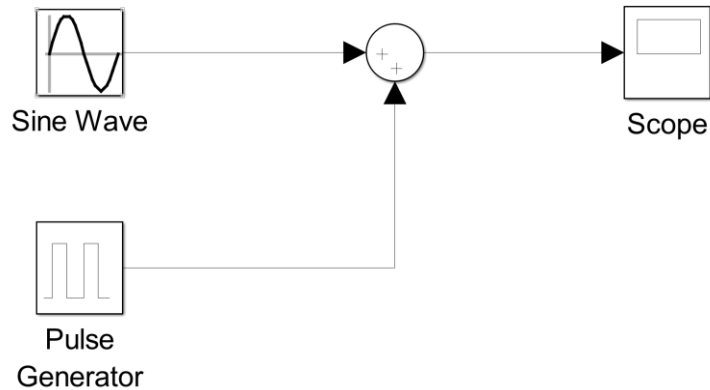
Simulink Block Parameters

▶ Block parameters can be configured

x(t) ▶ Available parameters depend on block type

Signals

▶ Simulink blocks generate/process signals
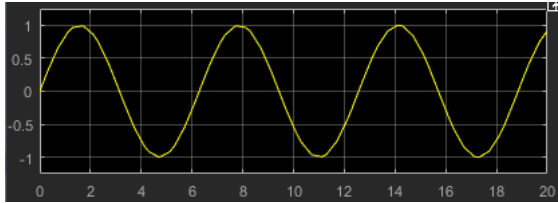
**Signals are variables
(values change over time)**

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

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
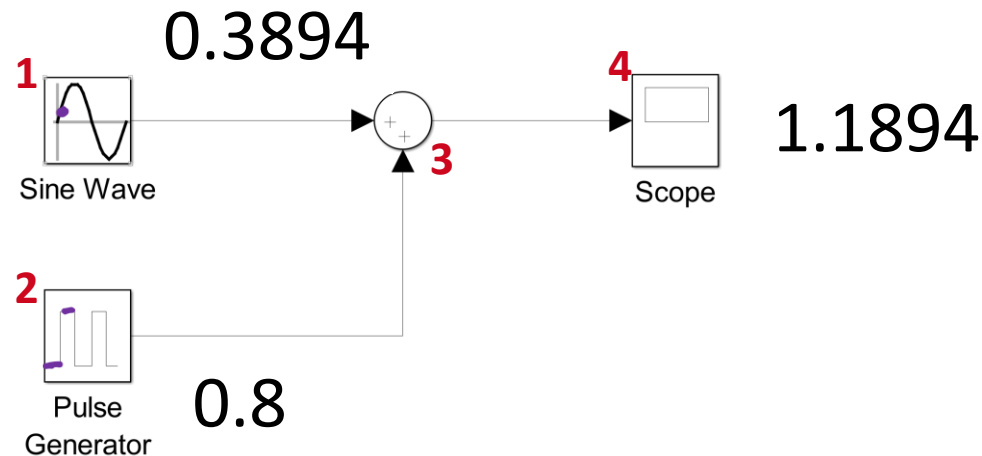
Signal Properties

▶ **Basic Signal Properties**

- Dimension
- Datatype
  - logical(boolean)
  - int8, int16, int32
  - uint8, uint16, uint32
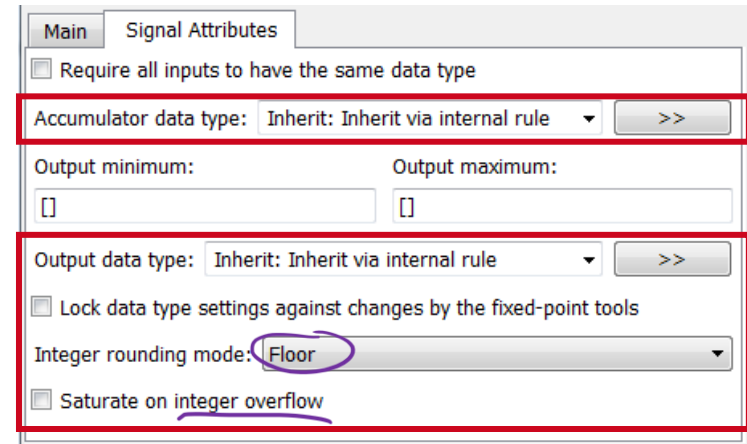 - float32(single), float64(double)
  - fixed-point data types

▶ **Signal properties are configurable by some blocks, e.g. Sum block**

int8

uint8

(1)001,0101

Vorzeichen · Mantisse · $2^{Exponent}$

Main	Signal Attributes

☐ 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

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

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

Bus-Signale

Mux

$c = \begin{pmatrix} a \\ b \end{pmatrix}$

Demux.

$d = \begin{pmatrix} e \\ f \end{pmatrix}$
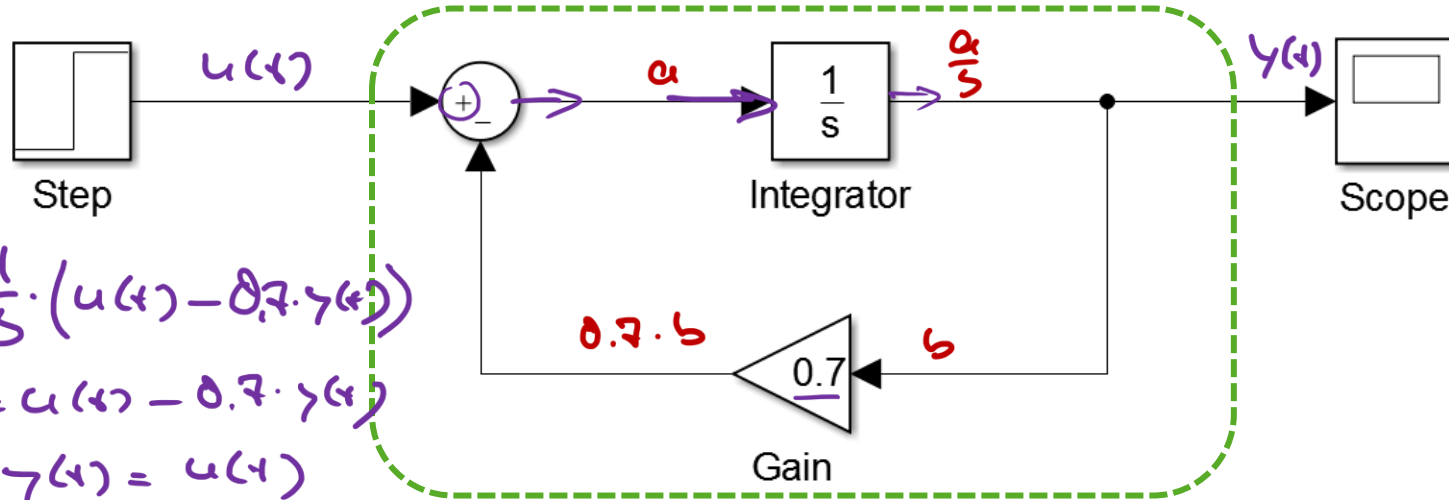
Vector Concatenate

$g = \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ e \\ f \end{pmatrix}$

int16   Convert   float32

Data Type Conversion

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY
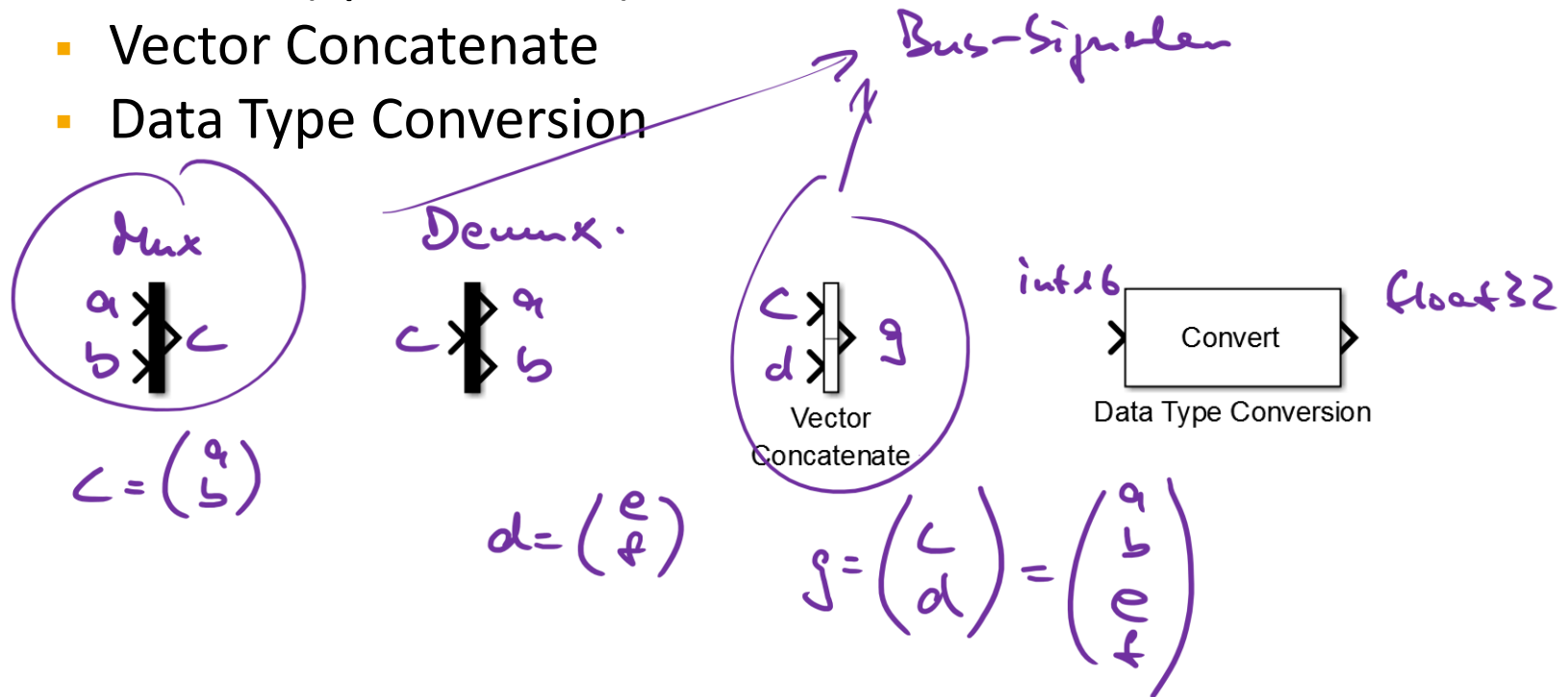
Modeling Possibilities

▶ Example – Low Pass Filter
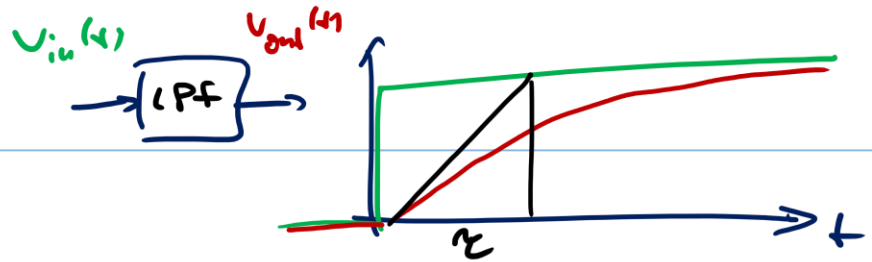
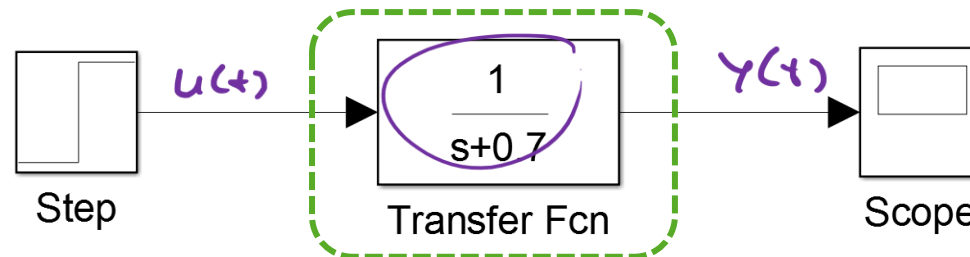$$y(t) = \frac{1}{s} \cdot \left(u(t) - 0.7 \cdot y(t)\right)$$

$$s \cdot y(t) = u(t) - 0.7 \cdot y(t)$$

$$(s + 0.7) \cdot y(t) = u(t)$$

$$y(t) = \frac{1}{s + 0.7} \cdot u(t)$$

▶ Different possibilities to model a behavior

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Subsystems

► Models can be hierarchically composed using Subsystem-blocks
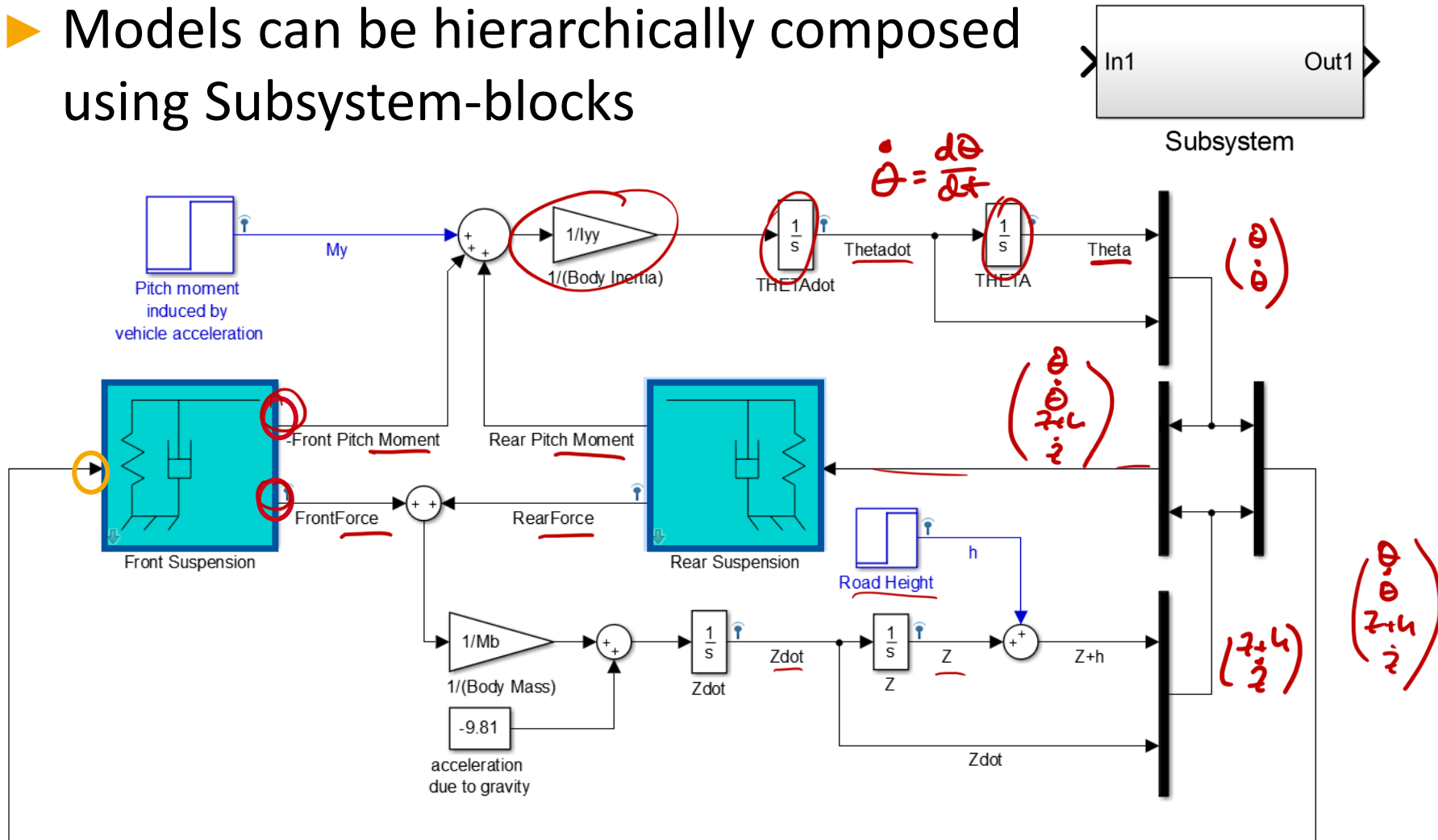
Copyright 1990-2015 The MathWorks, Inc.

Subsystem Implementation

► Front Suspension Subsystem implementation

Two DOF Spring/Damper Model

Input signal to Subsystem

MATLAB Workspace Constants

Output signals of Subsystem

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

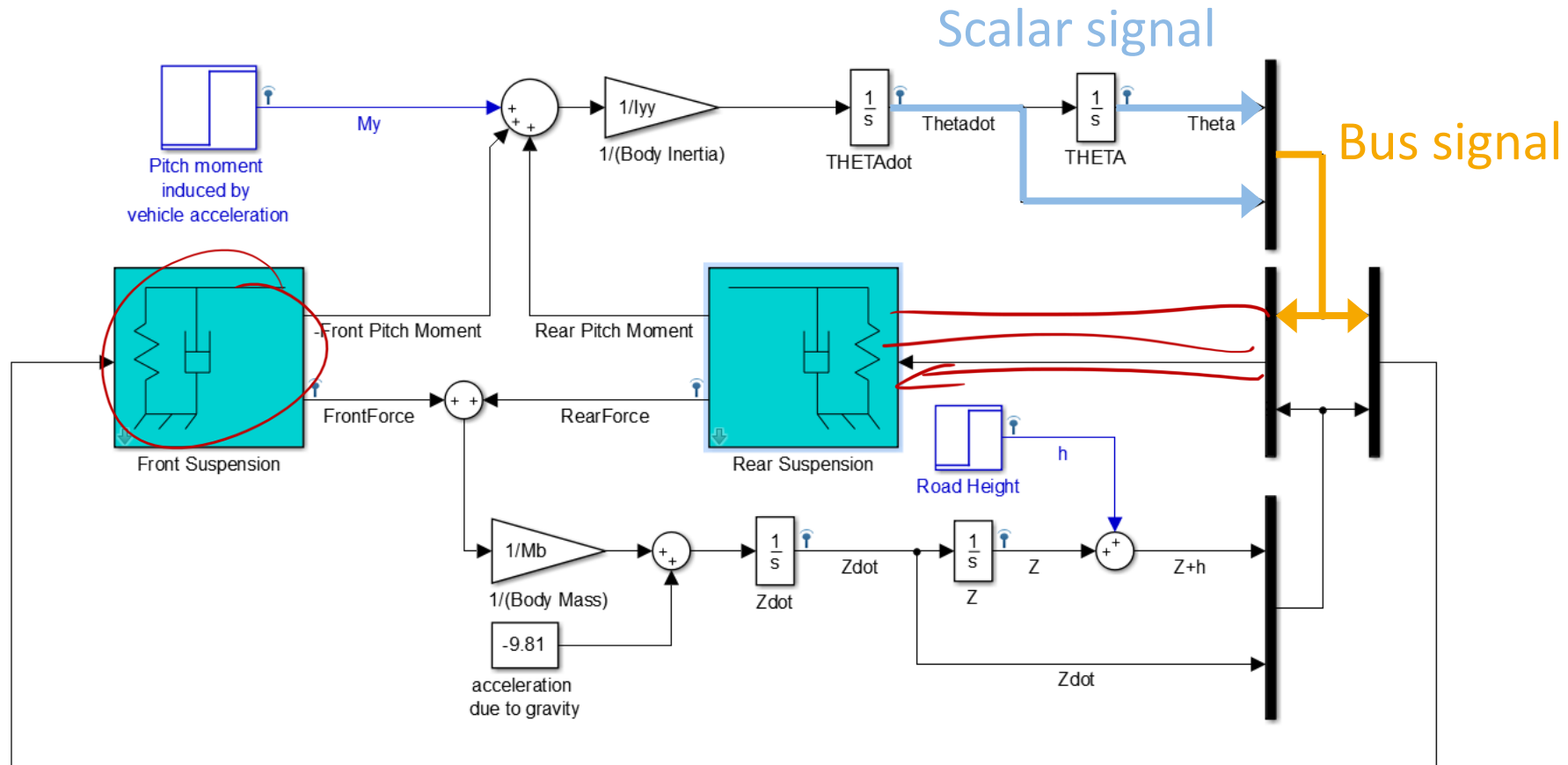# Abstraction Principles

► 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 |

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Bus Signals

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Advanced Simulink Blocks

▶ Extension by mixture with other modelling/programming paradigms

- Mfunctions
  - Write MATLAB programs within Simulink
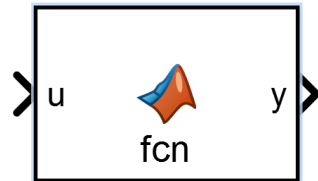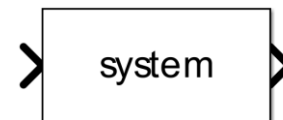

MATLAB Function

```
function y = fcn(u)

    % very sophisticated matlab code

    y = 2*u;
```
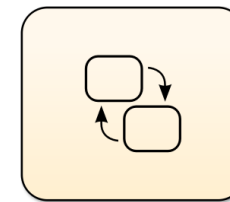
- Sfunctions
  - Call (compiled) code from Simulink

S-Function

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

Chart

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Stateflow

► **Stateflow example**

Transition guard

Initial state
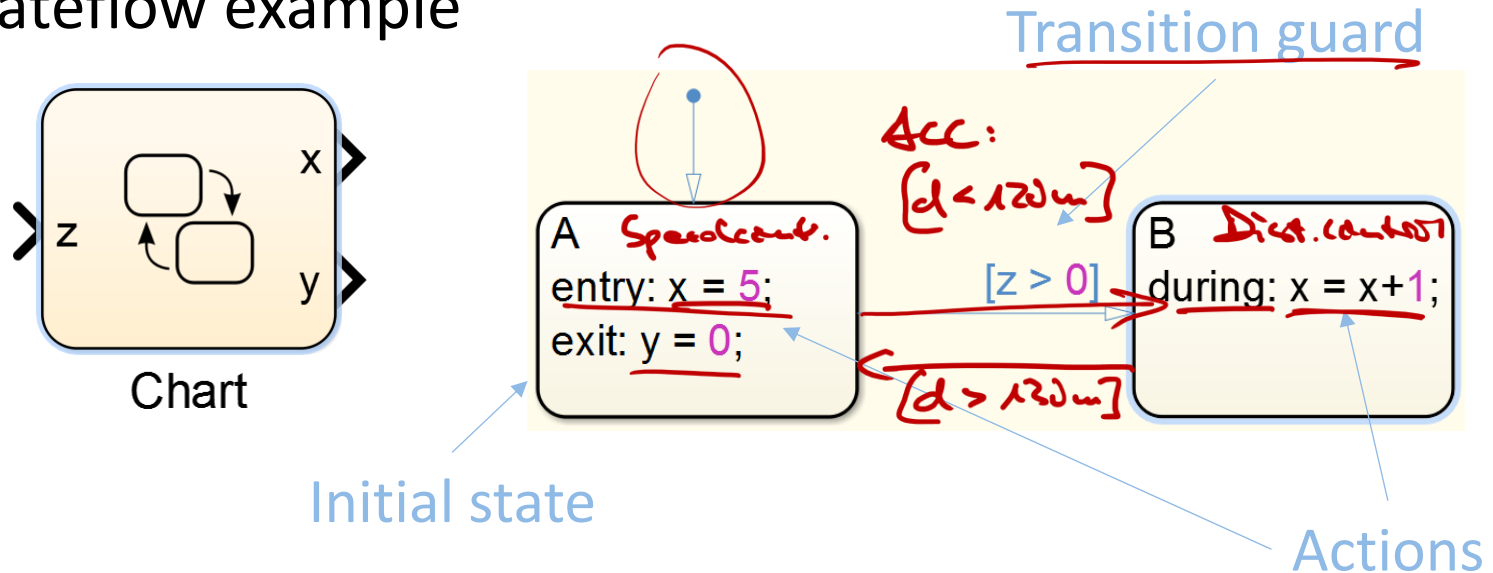
Actions

► 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

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Agenda

IES Part 8 – Embedded Software Development using Simulink

Informatik 11
Embedded Software

RWTHAACHEN
UNIVERSITY

Simulink in Context

requirements analysis

specification

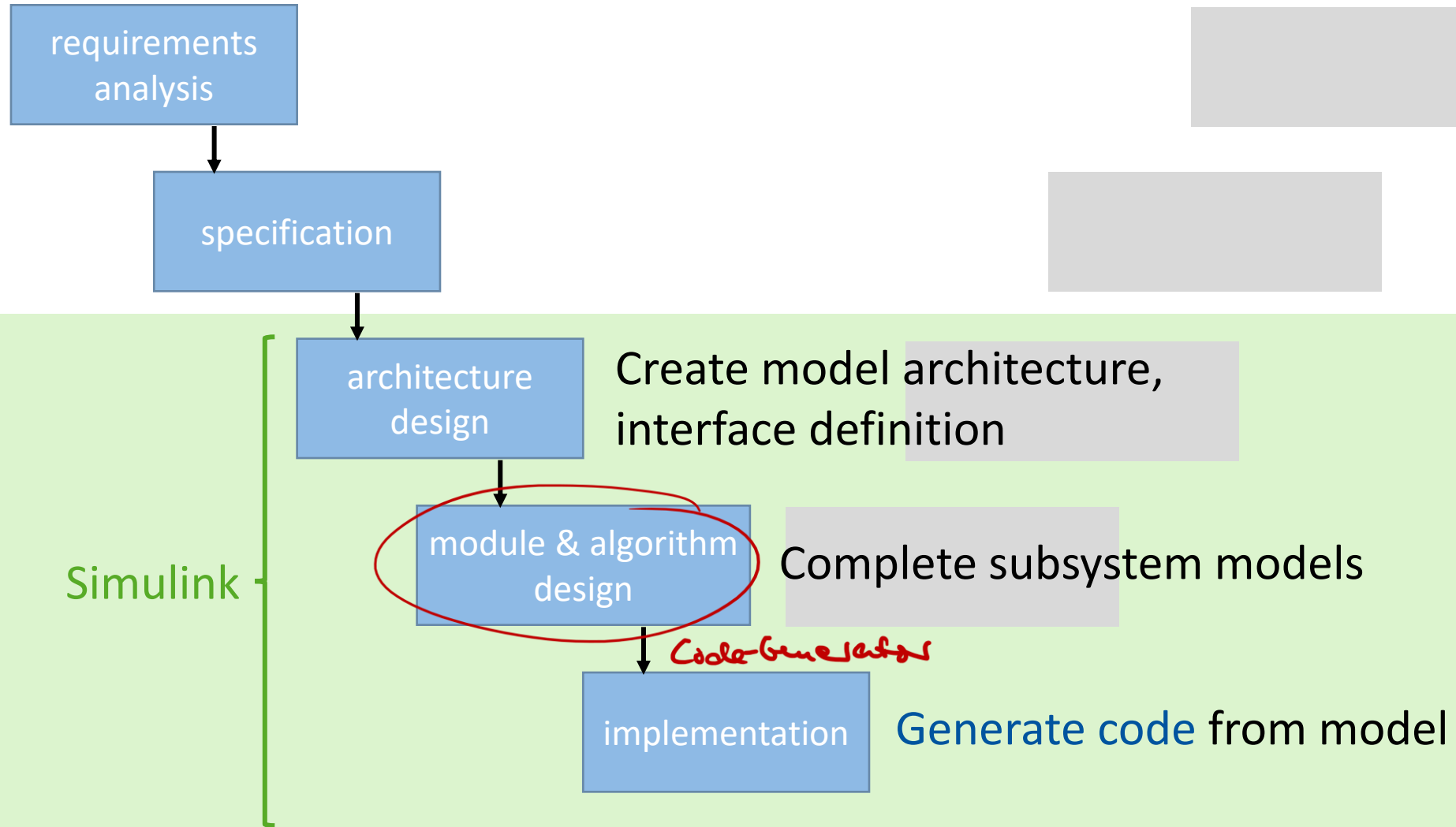architecture design

Create model architecture, interface definition

module & algorithm design

Complete subsystem models

Simulink

Code-Generator

implementation

Generate code from model

18
Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software
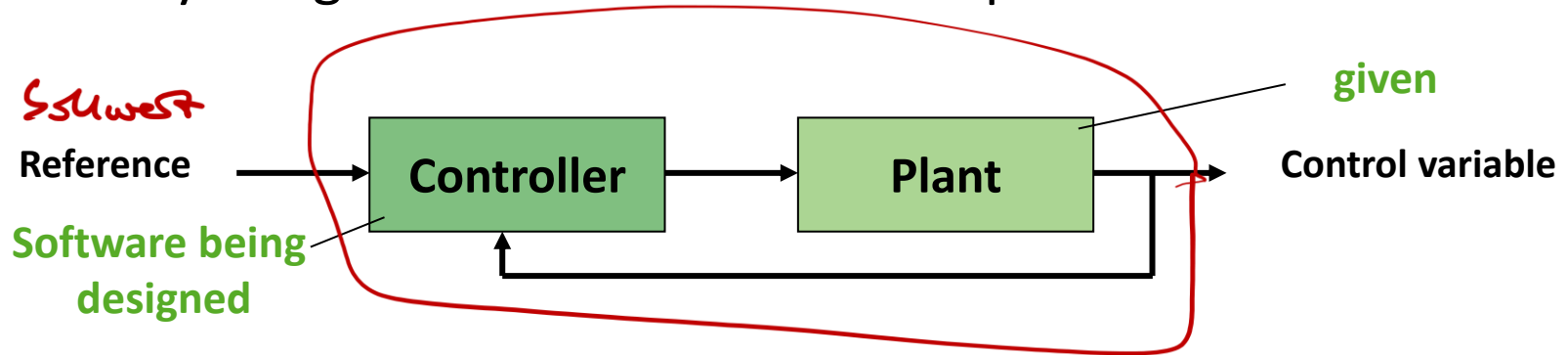
RWTH AACHEN UNIVERSITY

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

Informatik 11
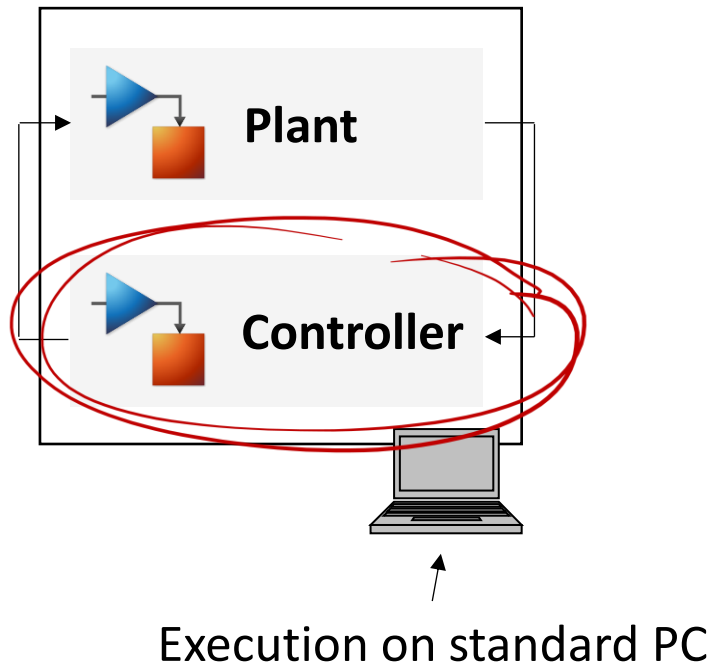Embedded Software

RWTH AACHEN UNIVERSITY

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

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

Informatik 11
Embedded Software

RWTHAACHEN
UNIVERSITY

MIL and SIL Tests

Model-in-the-loop (MIL)

Software-in-the-loop (SIL)

Plant

Controller

Plant

S-Function

C Controller

Generated code

Execution on standard PC

PIL and HIL Tests

Processor-in-the-loop (PIL)

Hardware-in-the-loop (HIL)

HIL simulator plattform
(Real-time capable)

Plant

Controller

Generated code

Execution on ECU

Introduction to Embedded Systems | Summer 2025 | Part 6 – Model-based Development & Simulink
Prof. Dr.-Ing. Stefan Kowalewski

**Informatik 11
Embedded Software**

RWTH AACHEN UNIVERSITY