Industrial Process Controllers and Simulators

Topic 7

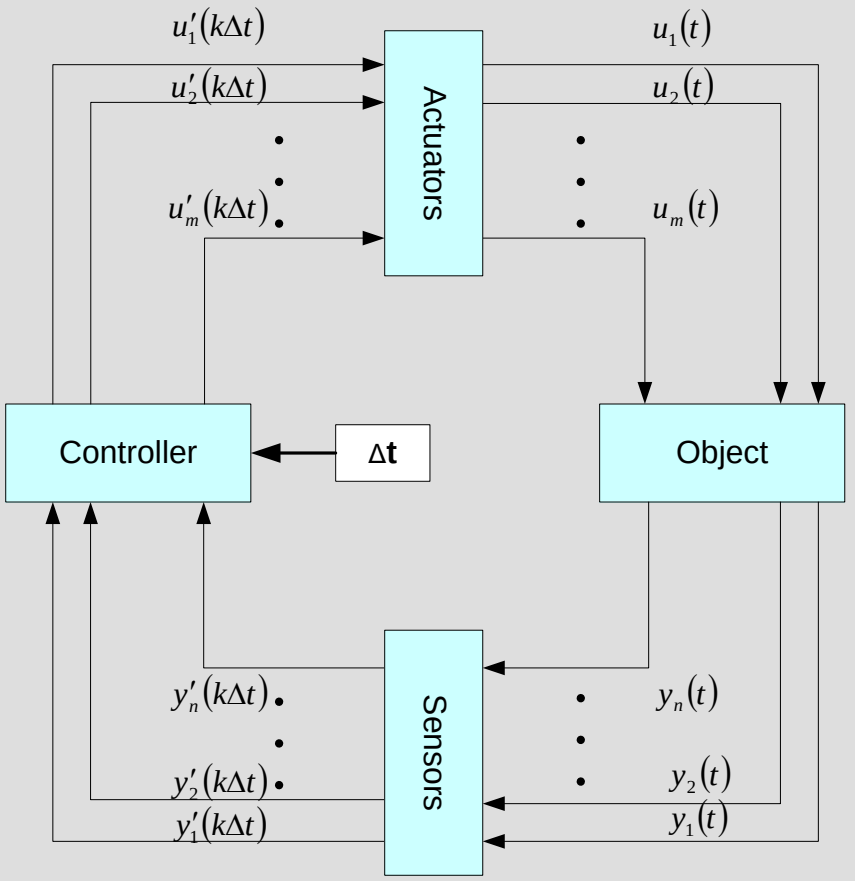
Simulators

General theory

Complex Real-Time Systems

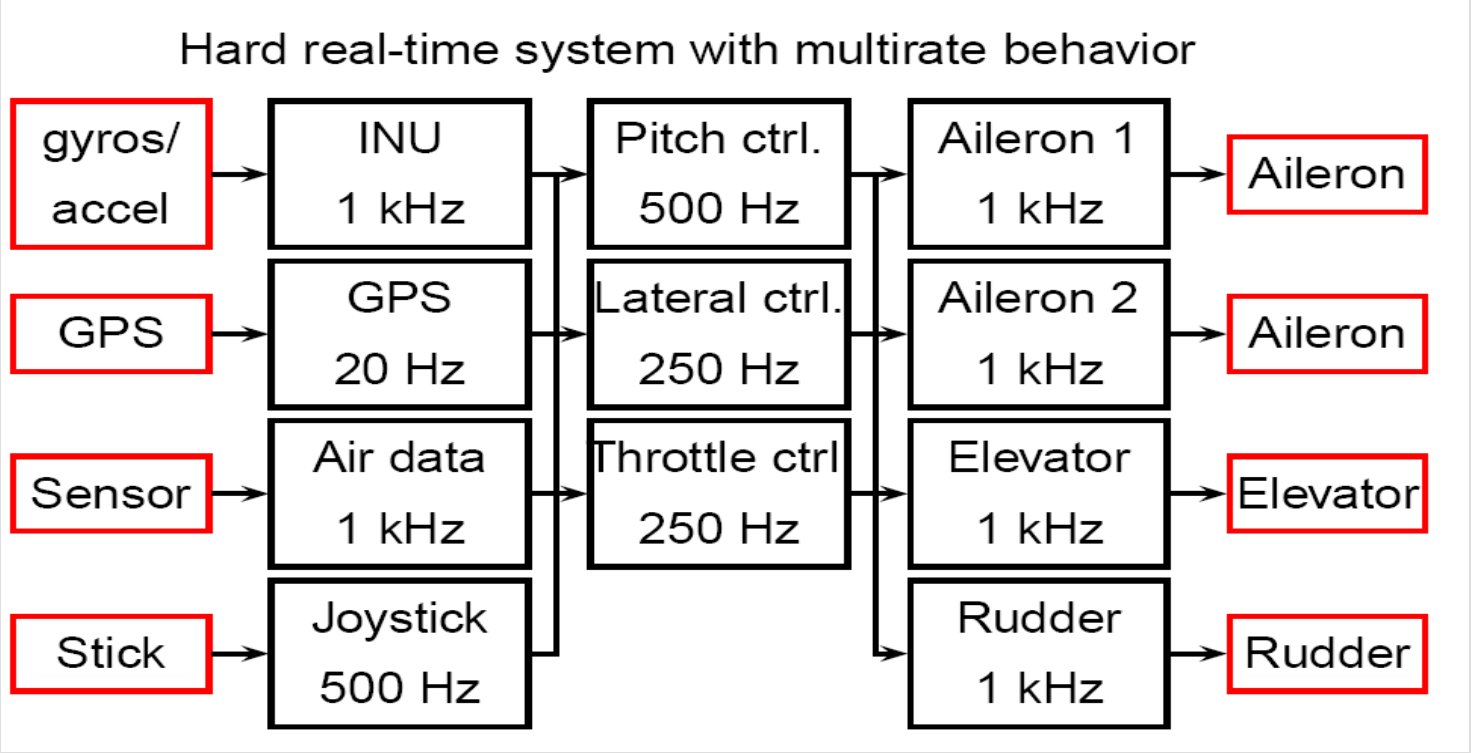
Basic structure:

1. A controlling system
2. A controlled system
3. The environment



a Complex Object

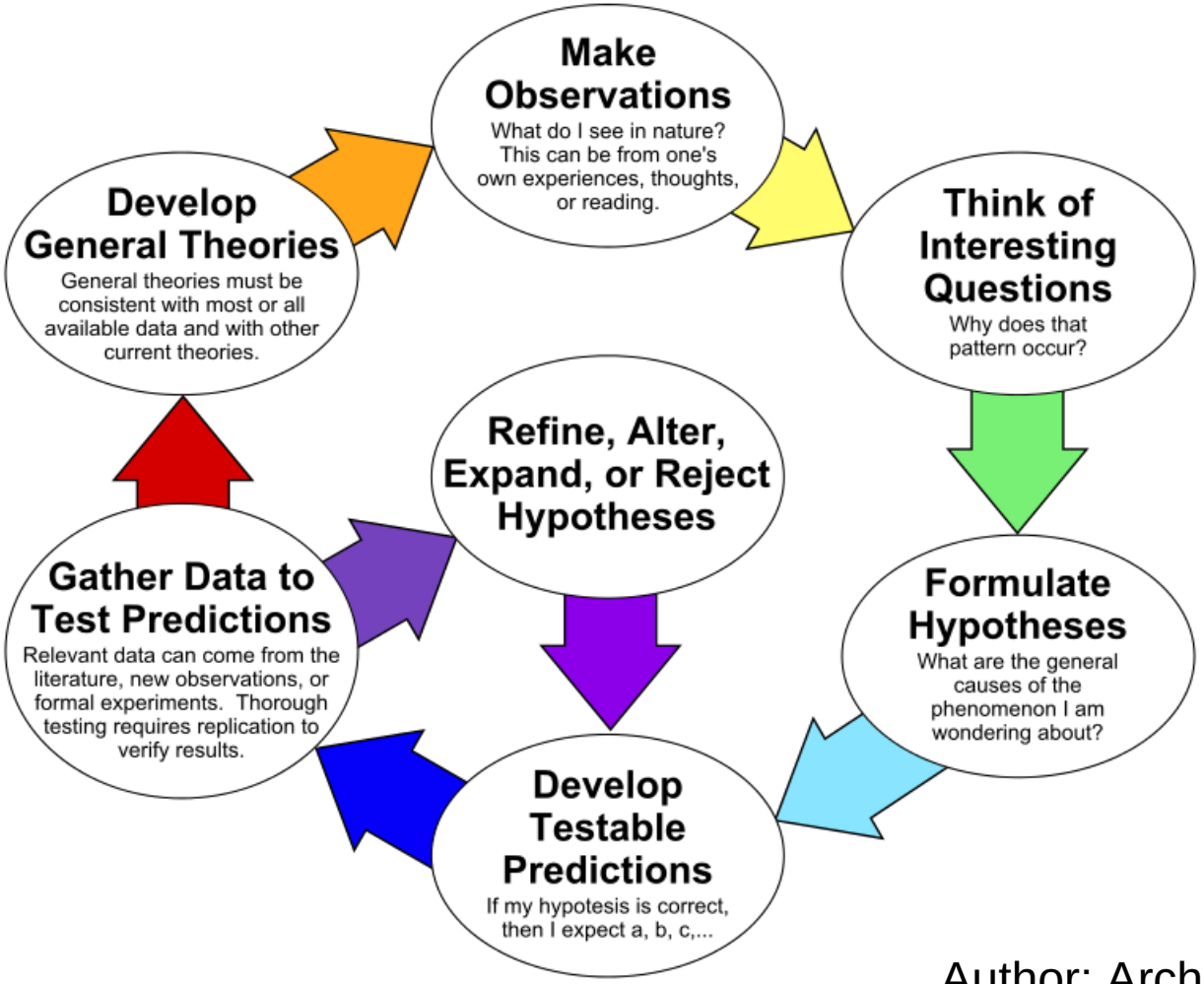
Example:



Source: Prof. St. Edwards

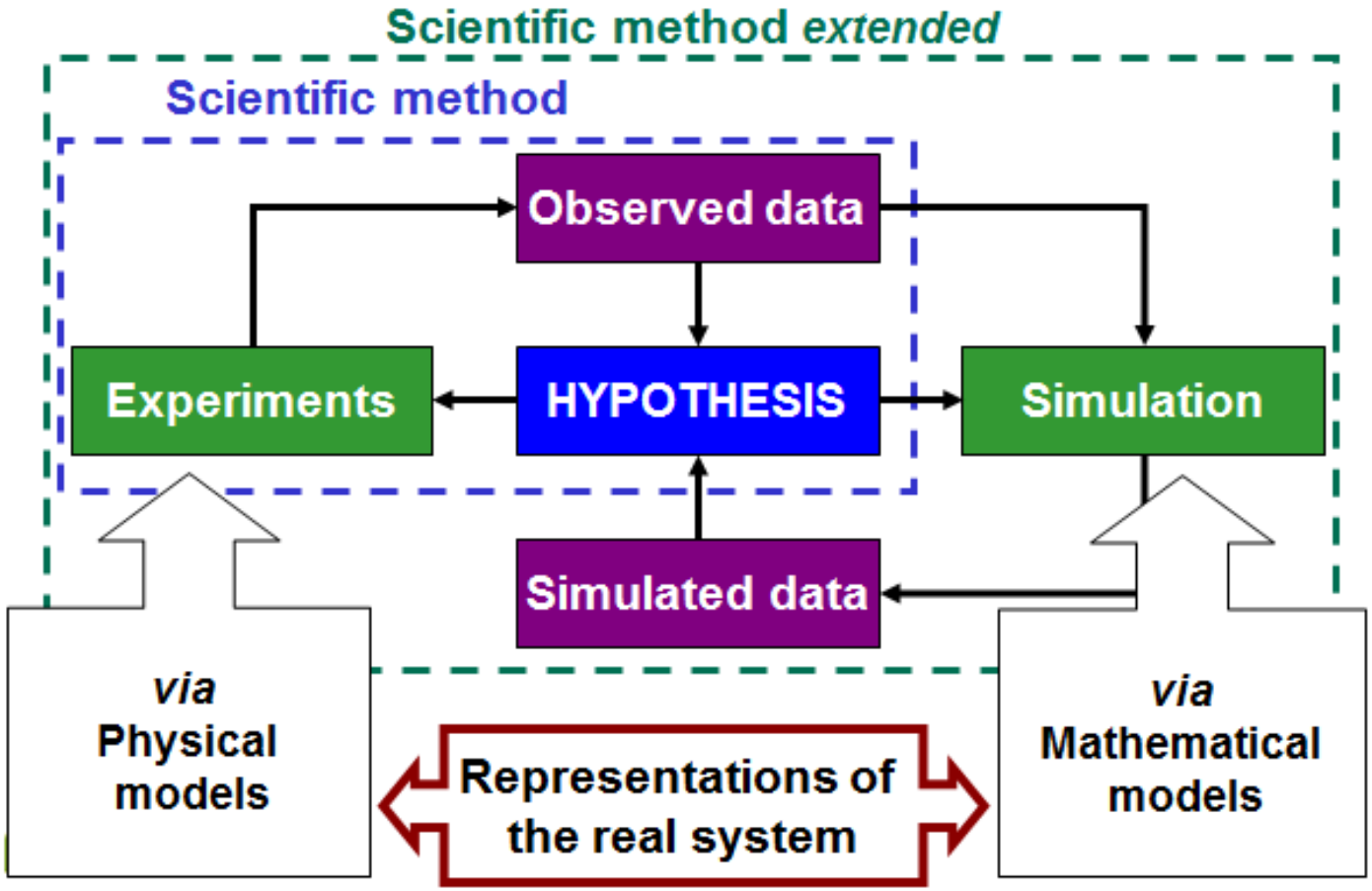
Simulators – a General Theory

The Scientific Method as an Ongoing Process



Author: Archon Magnus

Simulators – a General Theory



Author: Marcello Donatelli

Simulators – a General Theory

- The model is a representation of something
  - The something can be an idea, a concrete object or an abstract object
  - It is a logical representation of something
- Models can take many forms
  - Physical
  - Mathematical
  - Informal explanations

Simulators – a General Theory

- Simulation is the imitation of the operation of a real-world process or system over time
- Simulation is used in many contexts - safety engineering, testing, training, education…
- Simulation approaches:
  - Model-in-the-loop (MIL) – the very first step for every modelling process – both the object model and the controller are executed in one and the same environment – non-real-time
  - Software-in-the-loop (SIL) - both the object model and the controller are executed in one and the same environment – real-time execution
  - Processor-in-the-loop (PIL) - mathematic model is running in real-time, the controller is running on the target platform
  - Hardware-in-the-loop (HIL) – like PIL with I/Os also implemented

Simulators – a General Theory

- Simulation approaches:
  - Model-in-the-loop (MIL) – the very first step for every modelling process – both the object model and the controller are executed in one and the same environment – non-real-time

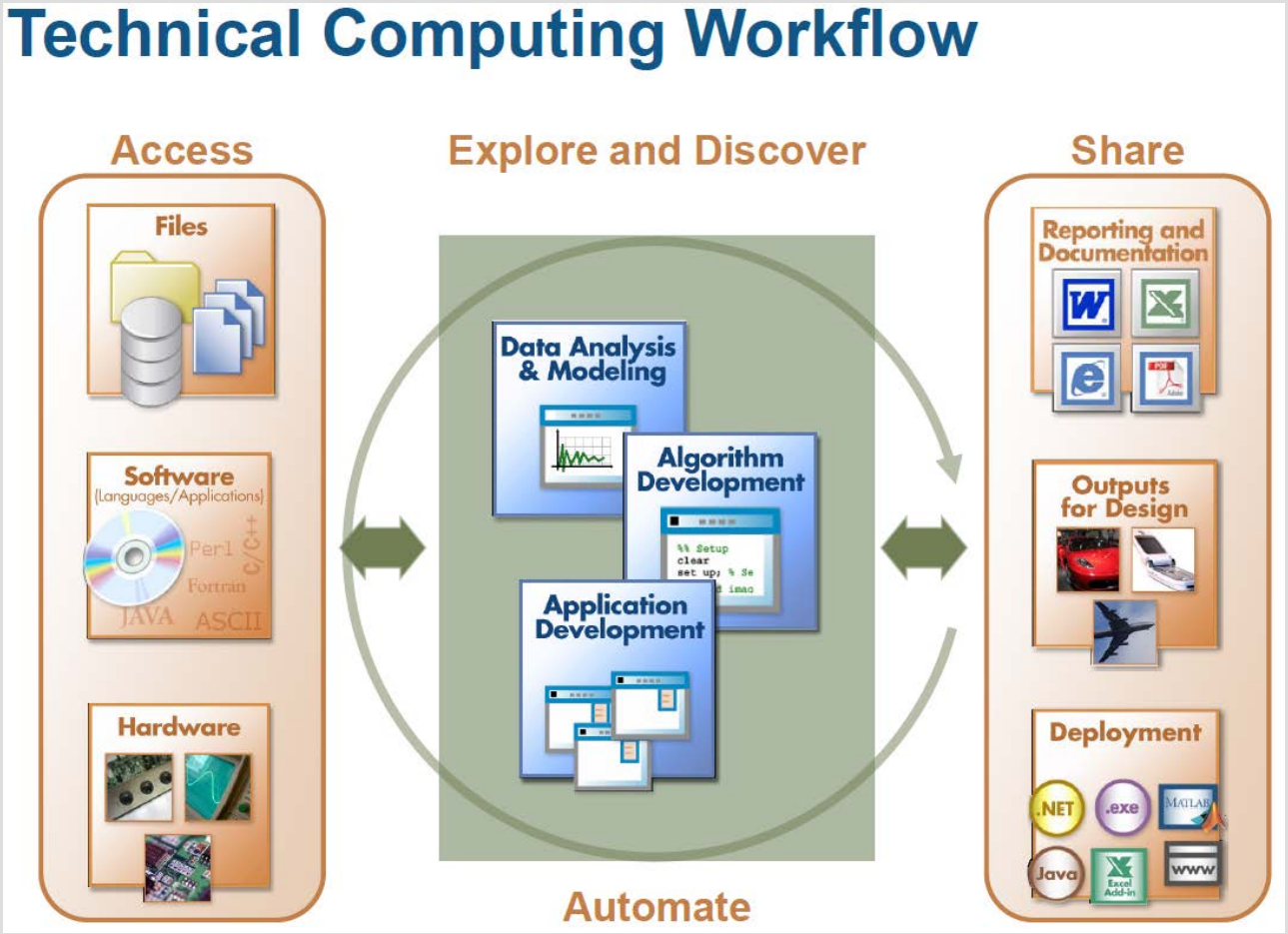
Example: MATLAB

Key Features:

- High-level language of technical computing
- Development environment for engineers, scientists
- Interactive tools for design, problem solving
- Mathematical function libraries
- Graphics and data visualization tools
- Custom GUIs
- External Interfaces: C, C++, Fortran, Java, COM, Excel, .NET

Simulators – a General Theory

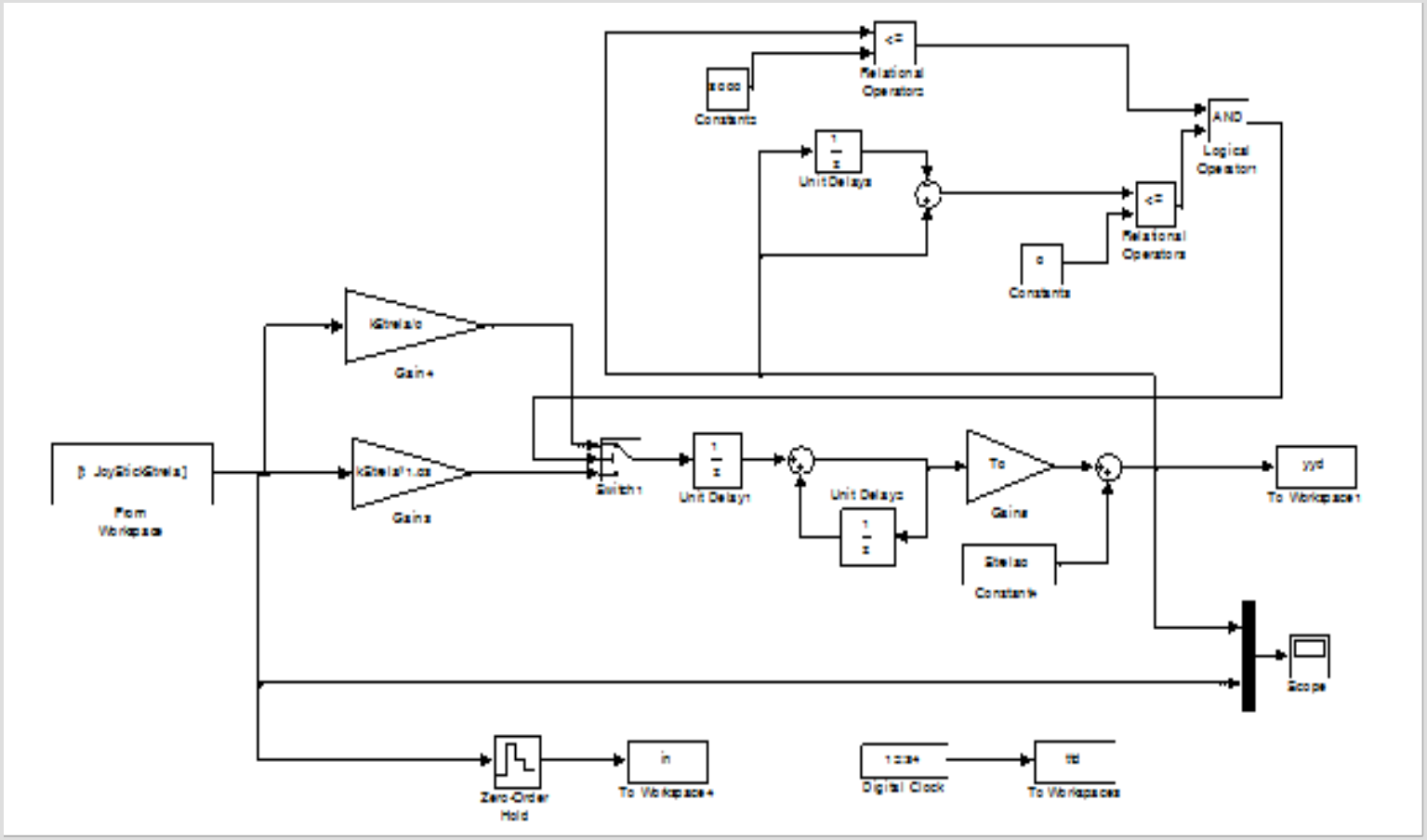
Example: MATLAB



Source: MathWorks®

Simulators – a General Theory

Example: MATLAB - Harbor Crane grapple discrete model



Simulators – a General Theory

Simulink

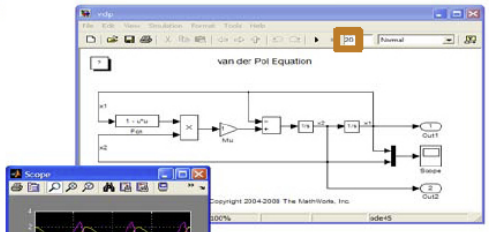
Simulink is a software package for modeling, simulating, and analyzing dynamical systems

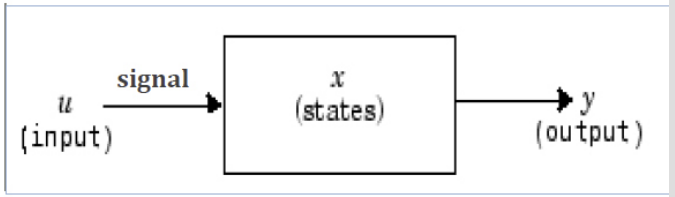
- Block diagram editing
- Nonlinear simulation
- Hybrid (continuous and discrete) models
- Asynchronous (non-uniform sampling) simulation
- Fully integrated with MATLAB, MATLAB toolboxes and blocksets.

Simulators – a General Theory

How Simulink Works

- Engine provides variable-step and fixed-step ODE solvers
- Block Diagram representation of dynamic systems
- Blocks define governing equations
- Signals are propagated between blocks over time

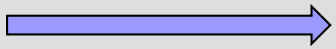




Simulators – a General Theory

Integration

- Matlab
  - Powerful program that allows complex engineering analysis
  - Screens
    - command
    - data
    - history
    - files
  - Where to get help
  - Script Files
  - Functions



- Simulink
  - Visual block oriented simulation tool
  - Library
  - Model window
  - Graphics
  - Interface with Matlab

Simulators – a General Theory

- Simulation approaches:
  - Hardware-in-the-loop (HIL) – mathematic model is running in real-time, I/O devices are implemented, sensors and actuators are modelled or real, controller runs in real-time on the target platform

Hardware-in-the-Loop simulation is a technique that is used for the development and testing of control systems which are used for the operation of complex machines and systems. With HIL simulation the physical part of a machine or system is replaced by a simulation.

Benefits:

- Increase safety
- Enhance quality
- Save time
- Save money
- Human factor
- Education

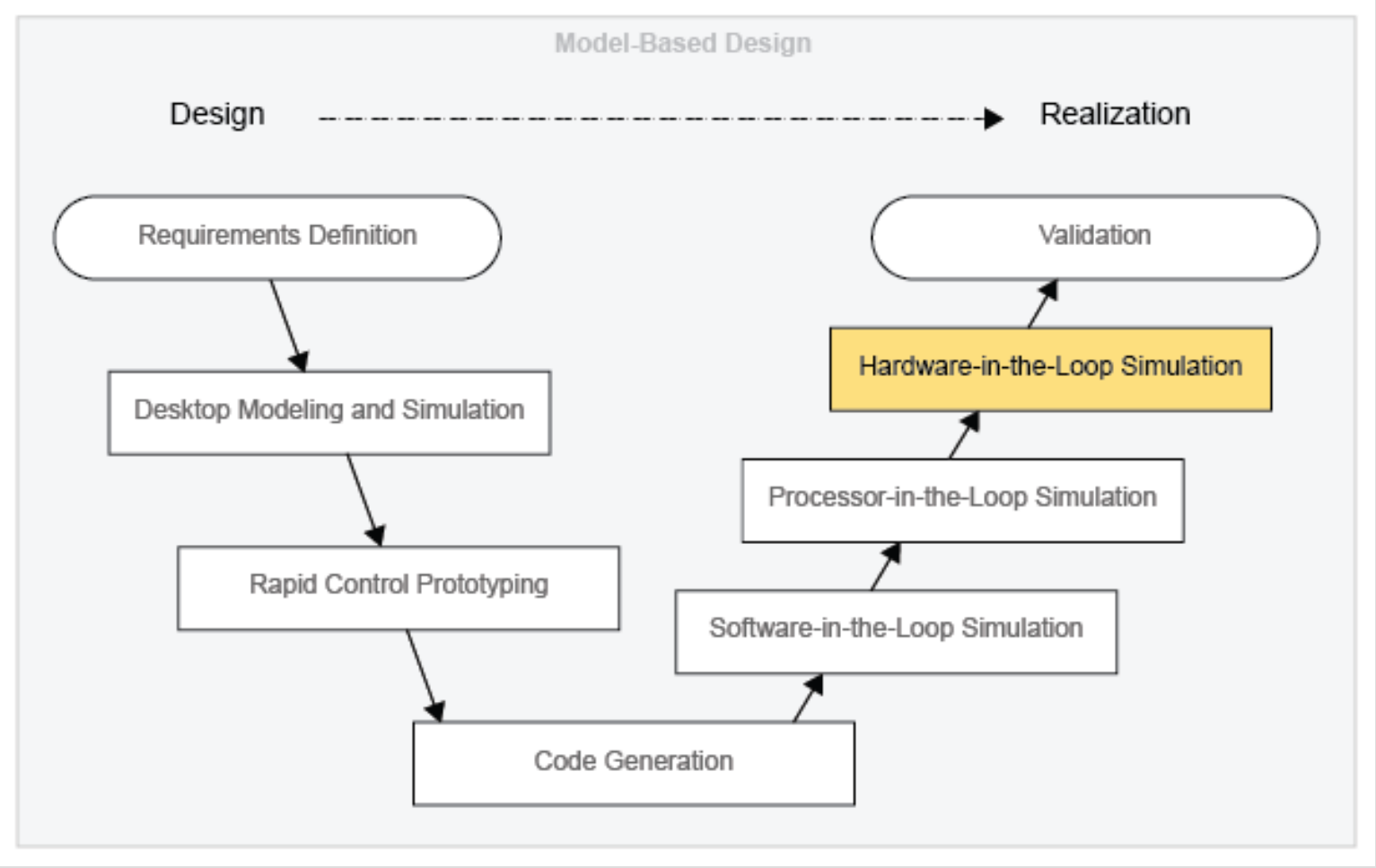
Simulators – a General Theory

Applications of HIL:

- Medical Devices
- Industrial machines
- Power Generation Systems
- White Goods
- Aerospace
- Automotive
- Process Control
- Nuclear Energy
- …..

Simulators – a General Theory

Model-Based Design workflow



Source: MathWorks®

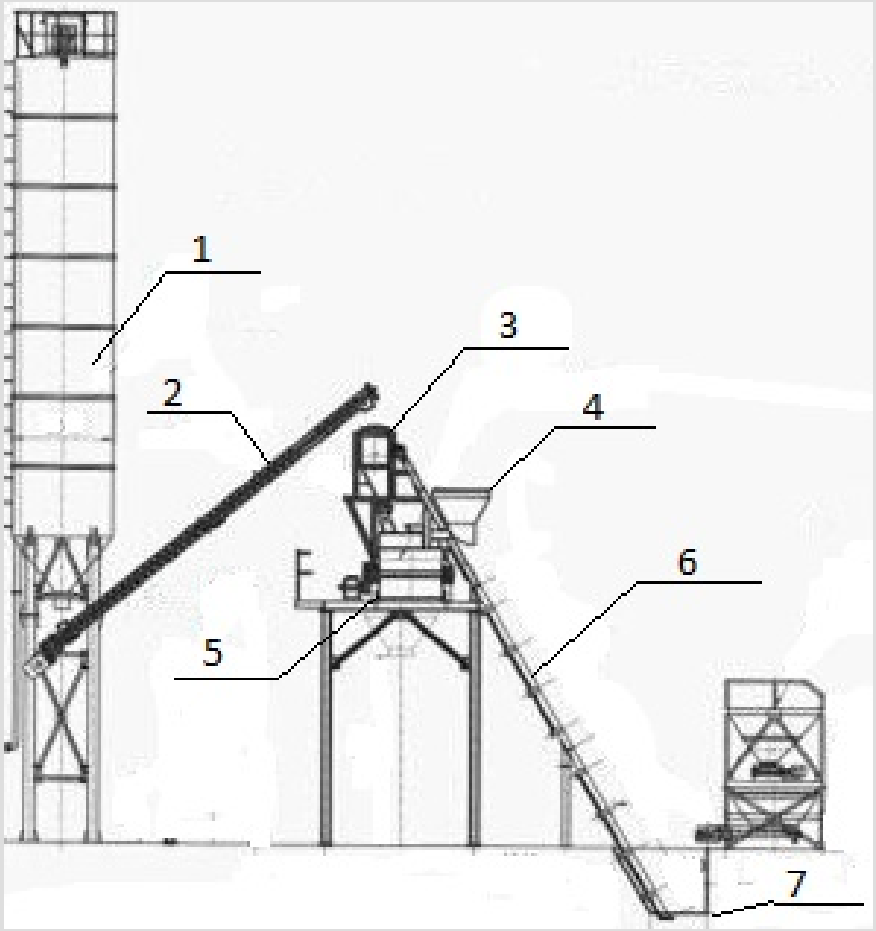
Simulators – a General Theory

Step in HIL simulation:

- Development of a mathematical model of the object and controller synthesis
- Design and implementation of HIL – both software and hardware
- Implementation of the controller on the real object

Simulators – an example

Concrete plant



- 1. Cement silo
- 2. Cement auger
- 3. Cement and water scales
- 4. Skip cart
- 5. Mixer
- 6. Skip cart railway
- 7. Inert materials scale

Simulators – an example

Concrete plant

Functional diagrams – representations by State machines

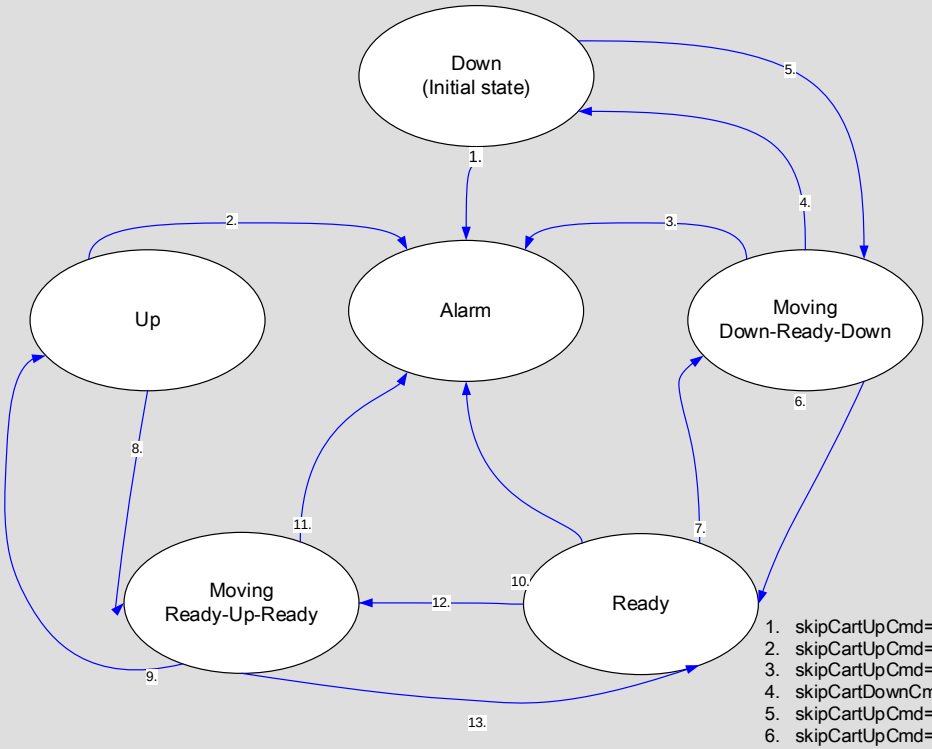
1. doseGravelCmd == ON: true
2. doseSandCmd == ON
3. cartIsDownFbk==OFF AND (doseSand OR doseGravel)
4. doseSandCmd==ON OR doseGravelCmd==ON
5. cartIsDownFbk == OFF
6. cartIsDown==OFF
7. Scale is overfilled
8. doseSandCmd==ON AND doseGravel==ON
9. doseGravelCmd==OFF
10. cartIsDown==OFF
11. Scale is overfilled
12. doseSandCmd==OFF
13. doseSandCmd==ON AND doseGravelCmd==ON
14. cartIsDown==OFF
15. Scale is overfilled
16. doseGravelCmd==ON AND doseSandCmd==OFF
17. doseGravelCmd==OFF AND doseSandCmd==ON
18. doseGravelCmd==OFF AND doseSandCmd==OFF
19. cartIsDownFbk==OFF AND (doseSand==ON OR doseGravel==ON)
20. doseSandCmd==ON AND doseGravel==ON
21. doseSandCmd==ON
22. doseGravelCmd==ON
23. cartIsDownFbk==OFF
24. doseSandCmd==ON
25. doseSandCmd==ON AND doseGravelCmd==ON
26. doseSandCmd==ON OR doseGravelCmd==ON
27. cartIsDownFbk==OFF AND (doseSandCmd==ON OR doseGravelCmd==ON)
28. cartIsDownFbk==ON AND cartIsEmpty==TRUE
29. doseGravelCmd==ON
30. cartIsDownFbk==ON AND cartIsEmpty==FALSE
31. cartIsDownFbk==OFF

Scale model

Simulators – an example

Concrete plant

Functional diagrams – representations by State machines



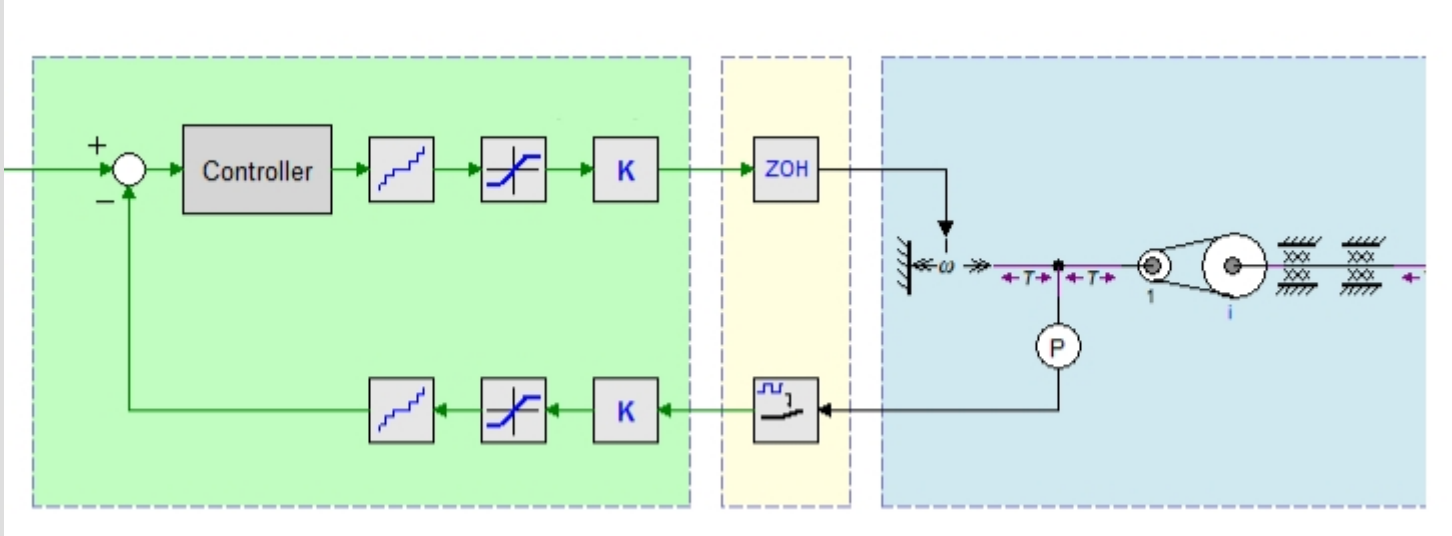
1. skipCartUpCmd==ON AND skipCartDownCmd==ON
2. skipCartUpCmd==ON AND skipCartDownCmd==ON
3. skipCartUpCmd==ON AND skipCartDownCmd==ON
4. skipCartDownCmd==ON AND cart's time is elapsed
5. skipCartUpCmd==ON
6. skipCartUpCmd==ON AND cart's time is elapsed
7. skipCartDownCmd==ON
8. skipCartDownCmd==ON
9. skipCartUpCmd==ON AND cart's time is elapsed
10. skipCartUpCmd==ON AND skipCartDownCmd==ON
11. skipCartUpCmd==ON AND skipCartDownCmd==ON
12. skipCartUpCmd==ON
13. skipCartDownCmd==ON AND cart's time is elapsed

Cart model

Simulators – an example

Concrete plant

Functional diagrams – representations by State machines



Cart model connection

Simulators – a General Theory

The End