Industrial Process Controllers and Simulators

Topic 10

Simulators validation

Problem Statement

Definitions

Validation is:

“A process of evaluating a system or its components in the process of its development or at its end to determine whether they satisfy specified requirements"

or

“A process of collection of information showing that both software and associated products satisfy the system requirements at the end of each cycle of its life, as well as satisfy some user needs in a specific case"

Simulators creation

- Development of a new product
- Modification
- Configuration / reconfiguration
- Use of ready-for-use software applications

Validation when creating a new software product using the "Configure/Reconfigure" approach

It is performed in two main ways:

- Validation of the basic software;
- Validation of the configuration data.

The validation of the basic software is done by using the existing software (one of the two options).

- For each subsystem and for each module responsible for any of the functions of the new SW;
- For each operation of the basic SW which results in a change of configuration/reconfiguration of the SW.

Data validation

Validation of data is divided into two activities:

- Validation of data values;
- Validation of techniques and methods for data storage, modification and use.

In both cases, the validation has to determine

whether the data are used correctly

and

whether they allow to develop SW that will function correctly.

Validation of data values

One has to determine the correctness of every data at every moment of the software operation.

- Check if the input signals are received in the right places;
- Check whether the input signals comply with the criteria for quality and correctness;
- Check for correctness of the reactions generated by the input signals.

Validation as part of a system

The final validation results from the successful validation of the following elements:

- Evaluation as a part of the system;
- Validation as part of a technological process;
- Product validation.

This is very important in case of a “partial” or “semi-natural” simulator.

Validation of a software application as part of an integral system

- In addition to the validation of software, a number of activities to validate the integral system are performed;
- Evaluation of the relation 'Requirements to the requirements − requirements to the software';
- In some cases of application development (using pre-programed software pieces, configuration/ reconfiguration) validation on system level can be performed directly.

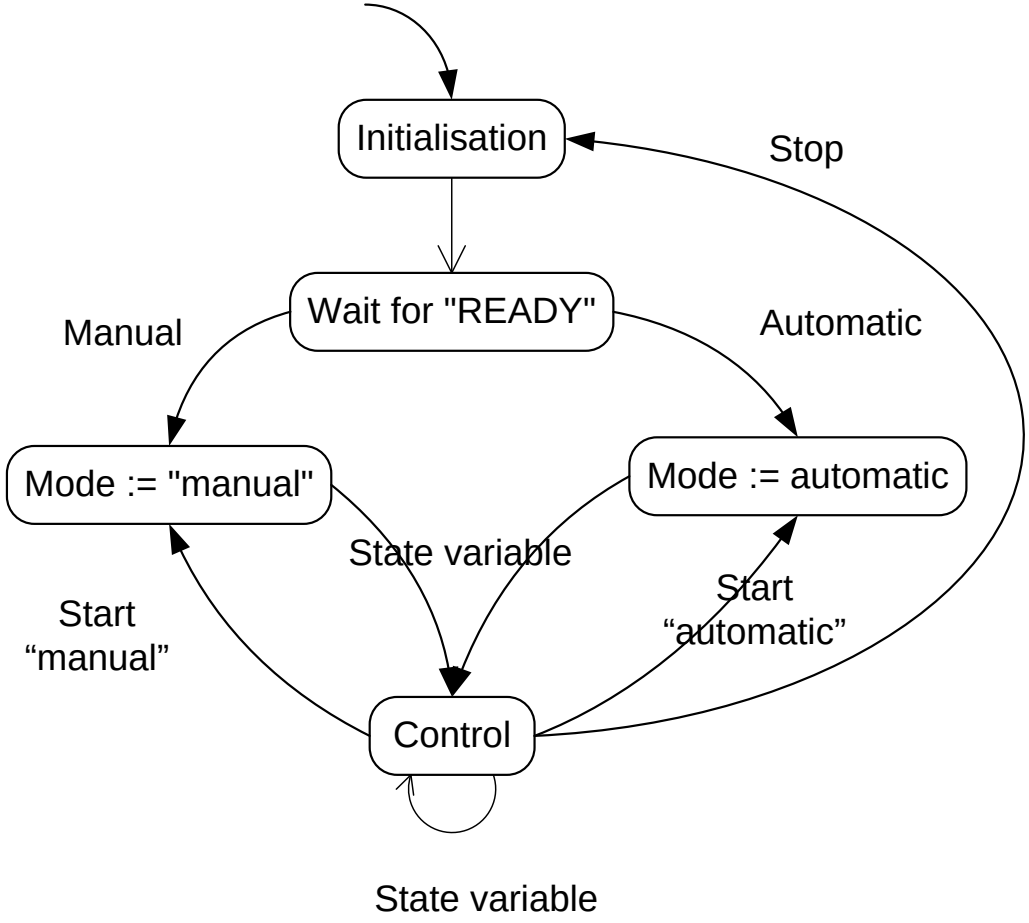
The program generator PrGen

Program generator information inputs

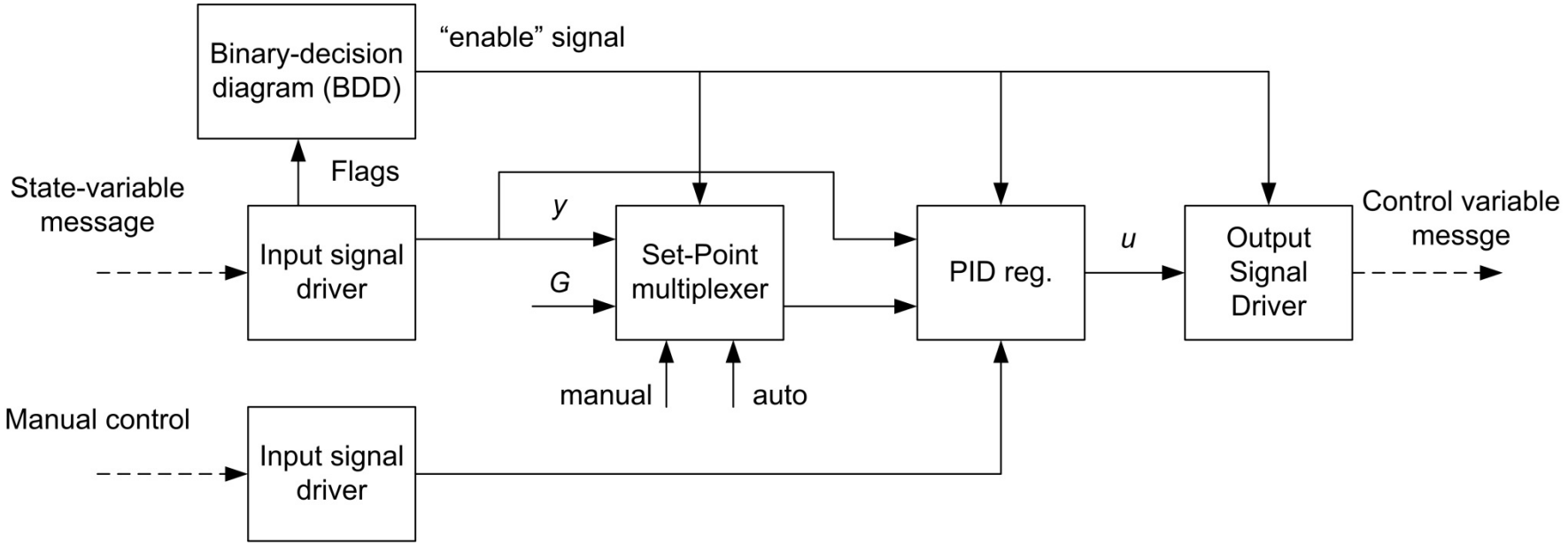
1. The Design Patterns (the template to which the code will be produced)
2. The Domain Meta-Data (the topology that should be modelled in the code, usually augmented with extra data provided by the developer)
3. The Domain Rules (the rules that dictate the structure and behaviour of the domain meta-data. This area is normally encapsulated in the generator program itself)

PrGen – program architecture

State-transition diagram



PrGen – program architecture



Signal-flow graph

PrGen – program architecture

PrGen is designed to answer the following requirements:

- hybrid object/process configuration specification;
- an object model facilitating the implementation of open and reconfigurable systems;
- a process model capturing both the reactive and the transformational aspects of system behavior in the context of various types of control systems;
- predictable scheduling of process execution and communication in the context of local and remote subsystems interactions;
- support for modern software engineering techniques such as program generation, formal verification of process and process interaction, etc.

Simulator projects

Harbour crane simulator

Harbour crane simulator

Harbour crane simulator

The following actions has to be modelled :

A1: Crane with no container, turning to container and lowering hook.

A2: Crane with container, lifting the container and turning to the ship.

A3: Lowering the load and positioning the container into the ship.

A4: Empty crane, from initial position (turned to the port). Turning to the ship, positioning above container, lifting load and unloading it into port.

Harbour crane simulator

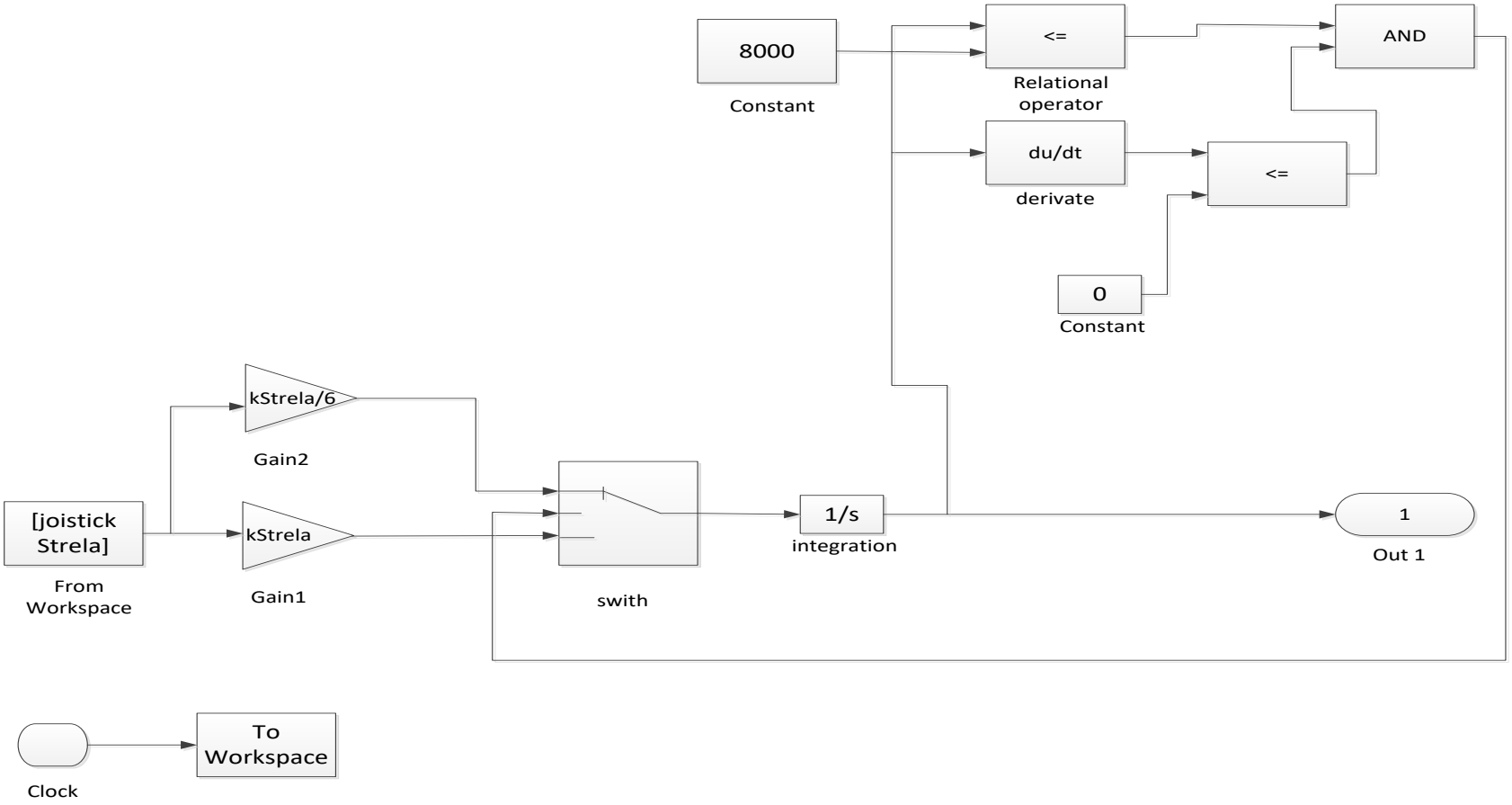
The crane has four different mechanical movements:

- turn around vertical axe,
- grapple lifting/lowering,
- grapple control
- grapple horizontal shifting.

The mechanical construction is independent for each of these movements.

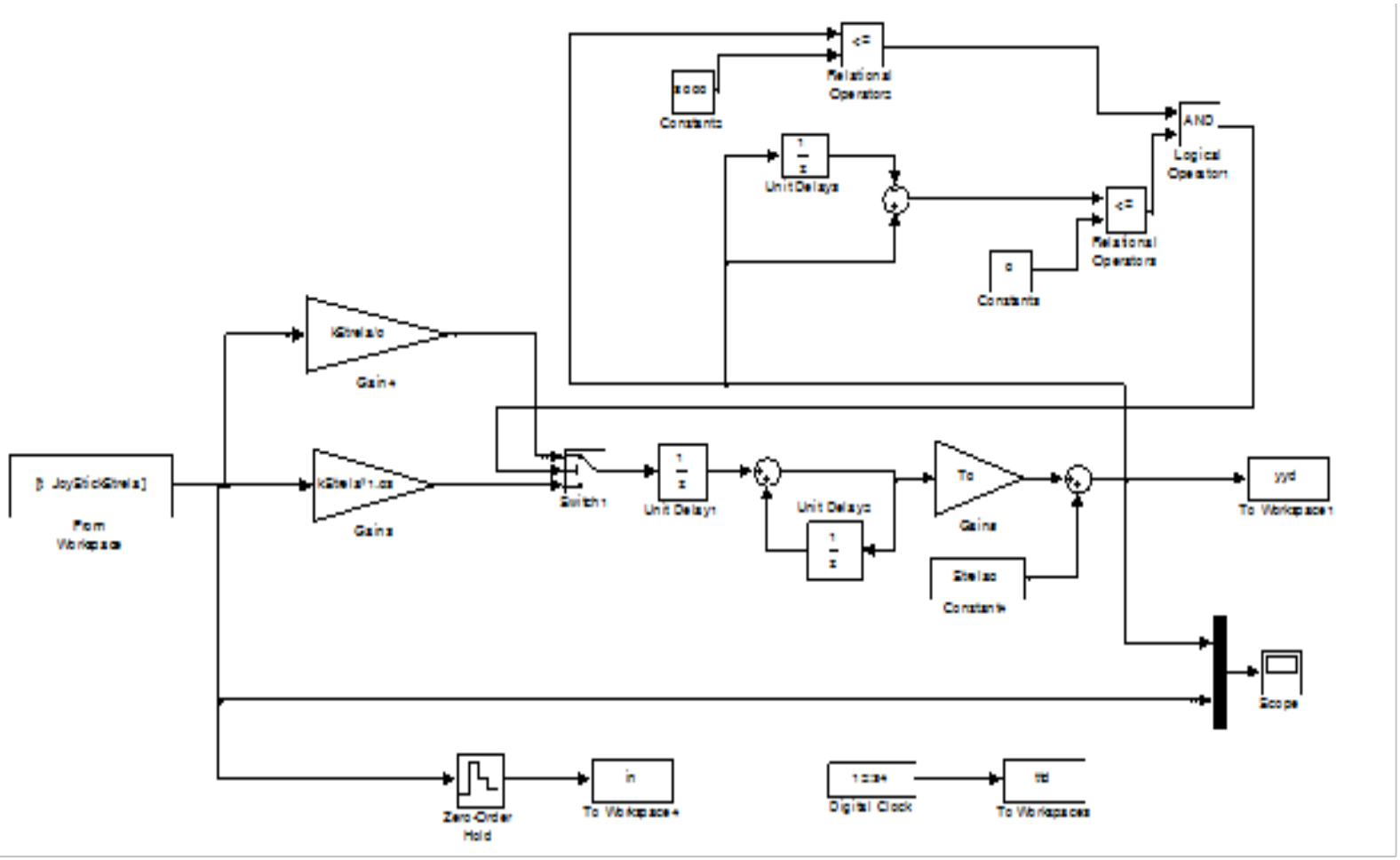
Harbour crane simulator - example

Harbor Crane grapple non-linear model



Harbour crane simulator – example cont.

Harbor Crane grapple discrete model

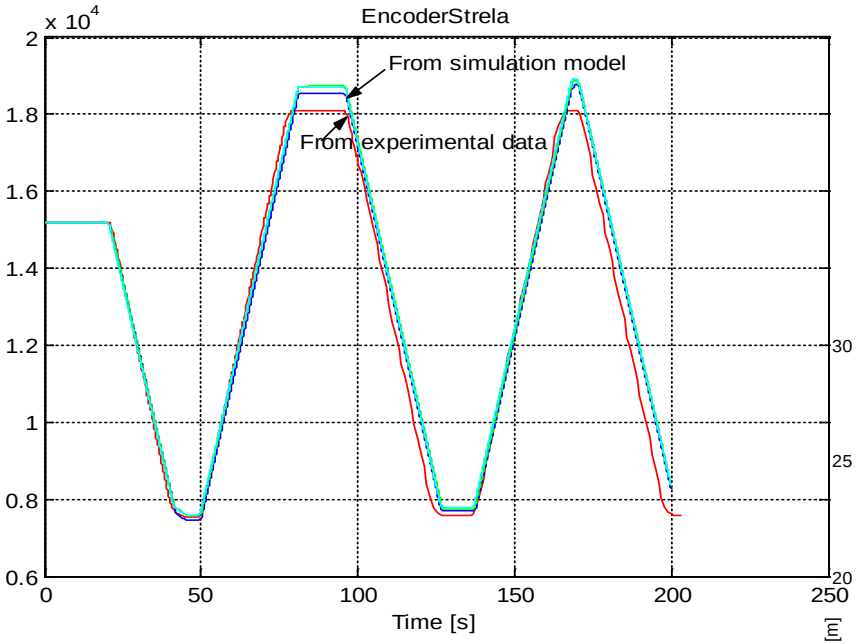


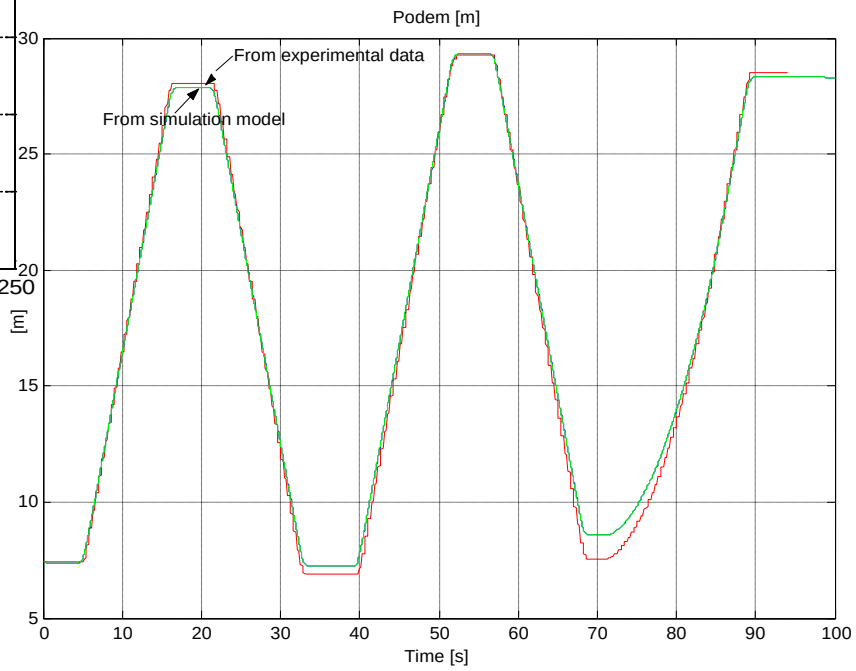
Harbour crane simulator

Comparison between crane real data and simulator outputs

N.	Measure-ments Crane (samples)	Measure-ments Simulator (samples)	Min. Movement Acc. %	Max. Movement Acc. %	Total Acc. %
A 1	680,000	890,000	95	98	97
A 2	1, 280,000	1,460.000	94	97	95
A 3	1,800,000	1,960,000	95	98	97
A 4	3,640,000	4,000,000	93	97	96

Harbour crane simulator validation





Conclusion

- The presented results confirm that program generation is a promising way to implement object simulators with hardware and software structures representing the modelled object very closely.
- Once the program generator internal model is validated, it becomes a useful tool for implementation of both simulators and control systems.
- The validation of the tool and generated by it systems are separated.

- Validation of the implemented simulators follows both approaches – model validation and data validation. By implementing sequentially these validation techniques, the designed simulator can achieve required accuracy and functionality.
- Using one and the same environment and tool for both controller and simulator building has the advantage to use only one tool for everything. This makes possible to distribute all changes in every building element immediately both in the controller and the simulator.

The End