

ERRORSIM: A SIMULATOR FOR ERROR PROPAGATION ANALYSIS OF CONTROL SYSTEMS DEVELOPED IN SIMULINK

Andrey Morozov*, Mustafa Saraoğlu, Klaus Janschek
MATLAB EXPO, Germany
Munich, 27.06.2017

Introduction

- Model-based system analysis
- Dependability and error propagation
- Analytical and simulative approaches

ErrorSim

- Workflow
- Fault types and injection methods
- Reported statistical information

Case study

- Reference Simulink model
- Experiments
- Result interpretation

Introduction

- Model-based system analysis
- Dependability and error propagation
- Analytical and simulative approaches

1

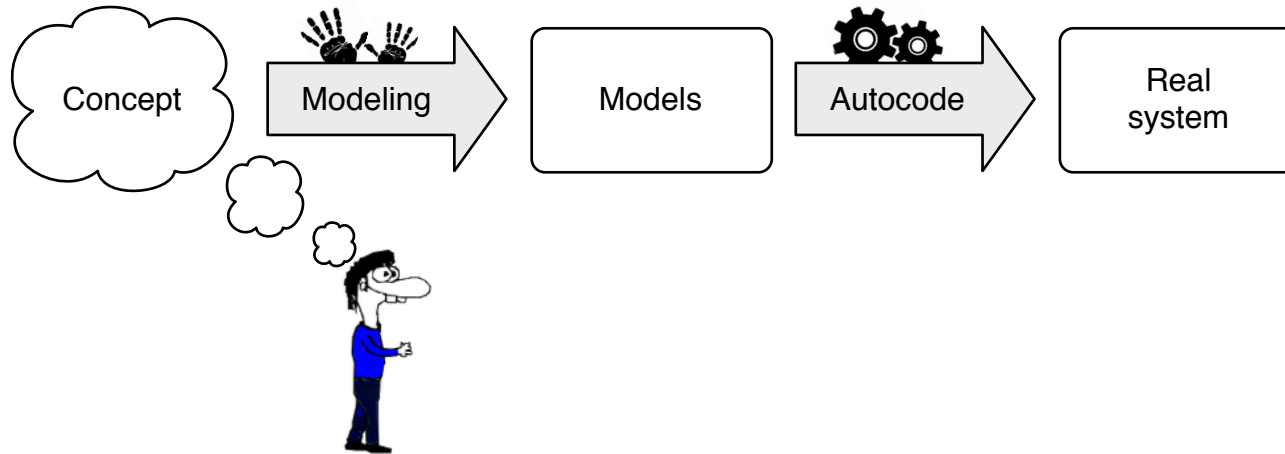
ErrorSim

- Workflow
- Fault types and injection methods
- Reported statistical information

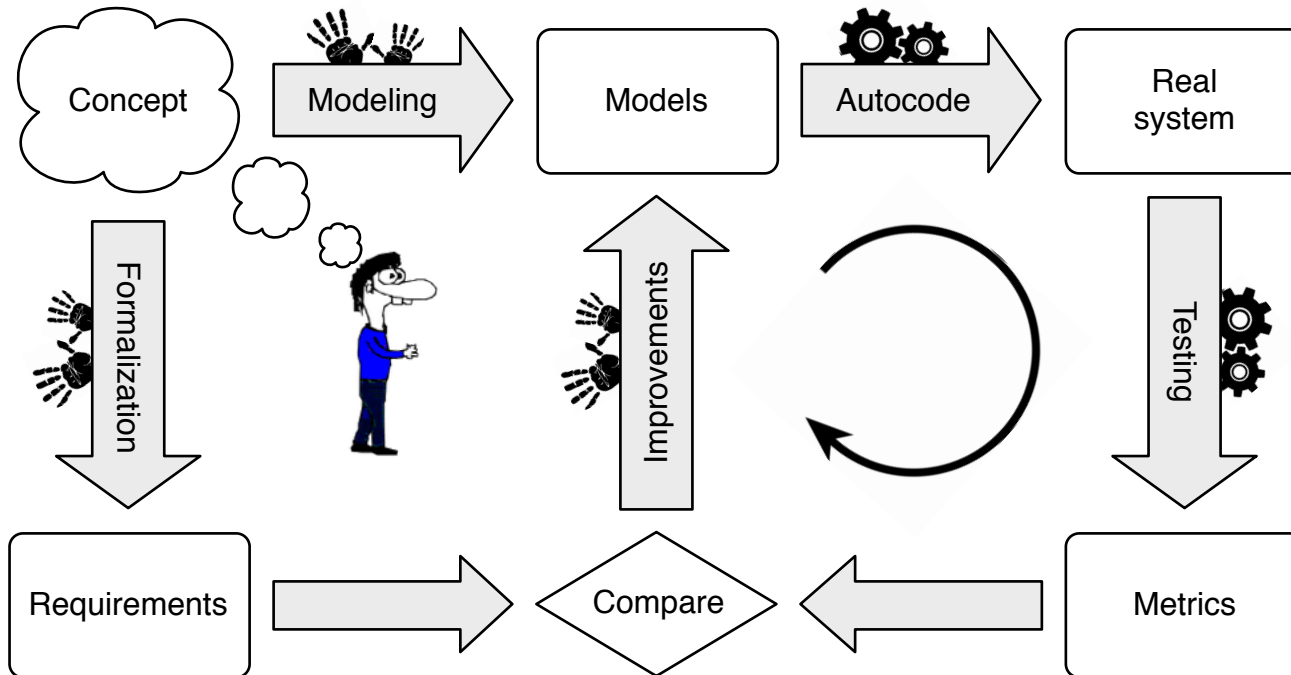
Case study

- Reference Simulink model
- Experiments
- Result interpretation

Simplified **model-based system development** workflow:

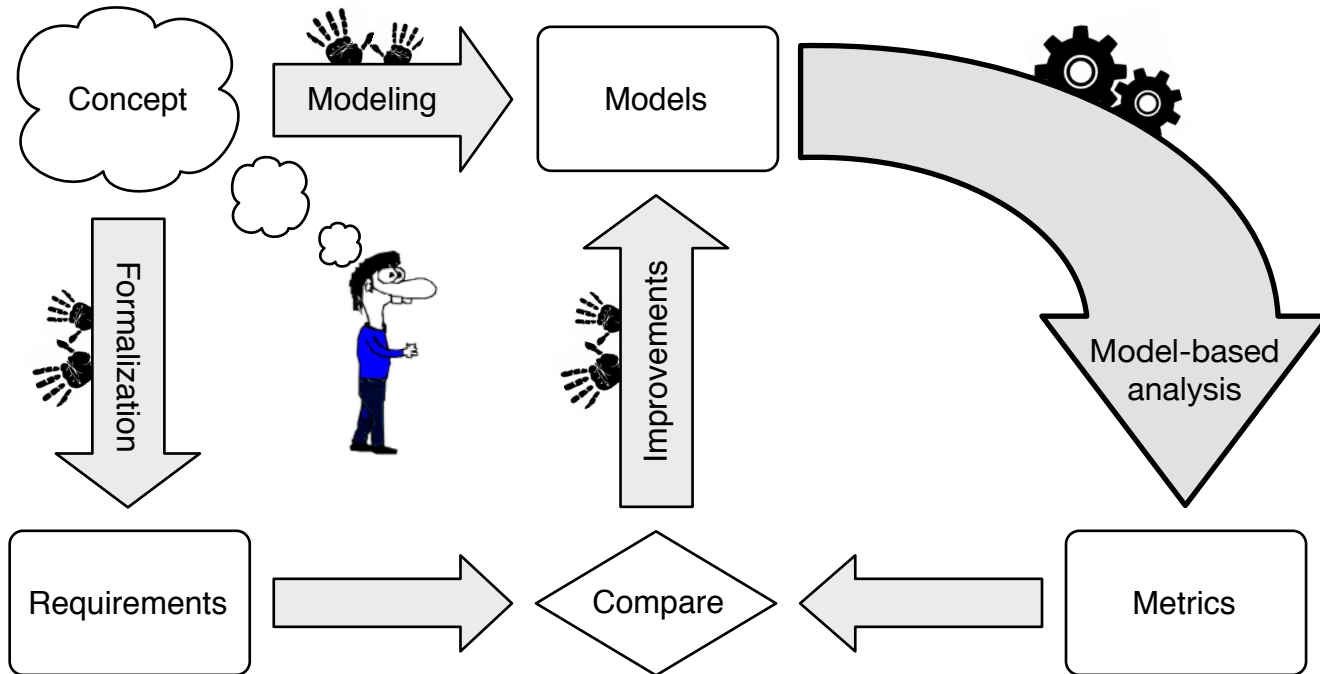


Conventional **system analysis**:



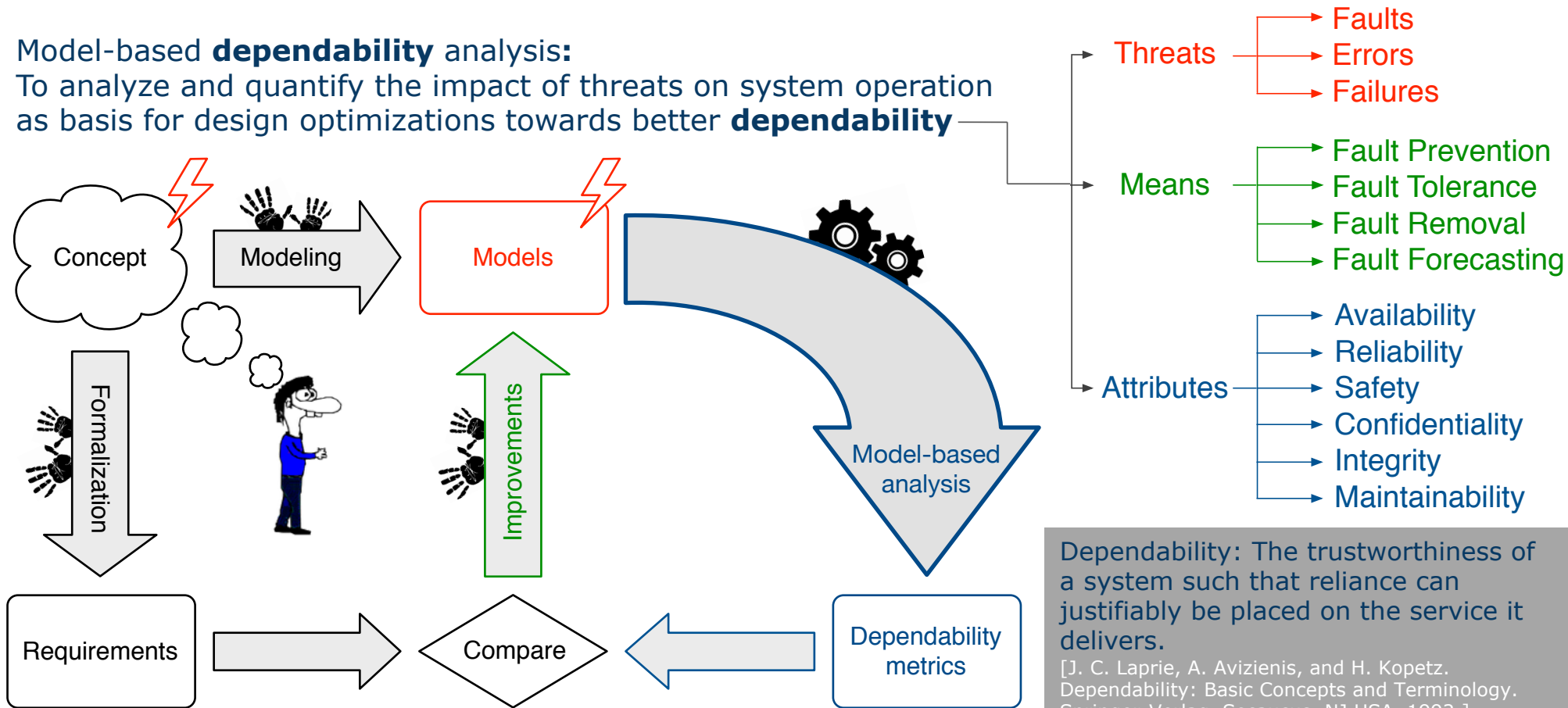
Model-based system analysis:

To recognize system characteristics and inherent shortcomings in the early design stages in order to provide optimized design solutions.



Model-based **dependability** analysis:

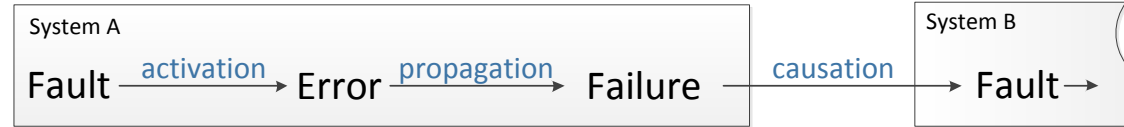
To analyze and quantify the impact of threats on system operation as basis for design optimizations towards better **dependability**



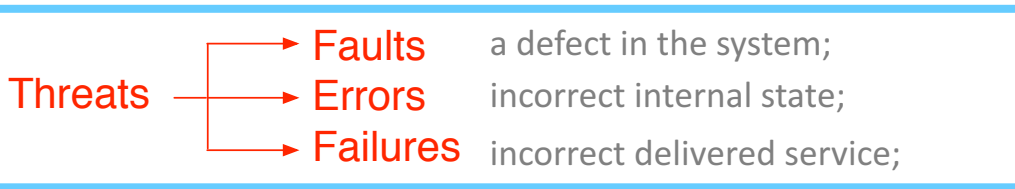
Dependability: The trustworthiness of a system such that reliance can justifiably be placed on the service it delivers.

[J. C. Laprie, A. Avizienis, and H. Kopetz. Dependability: Basic Concepts and Terminology. Springer-Verlag, Secaucus, NJ, USA, 1992.]

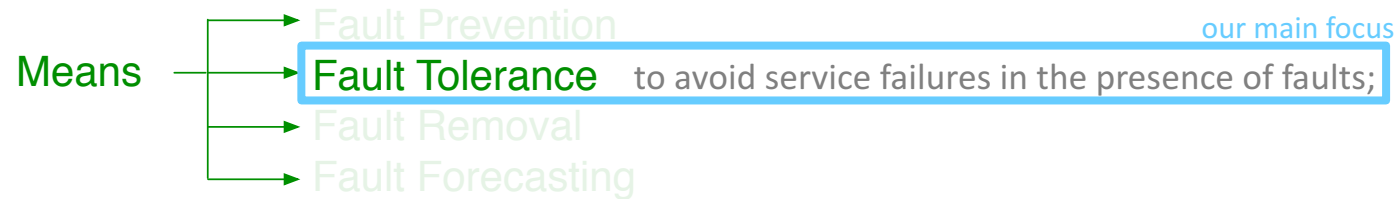
Terminology Laprie[1992]



our main focus



Dependability

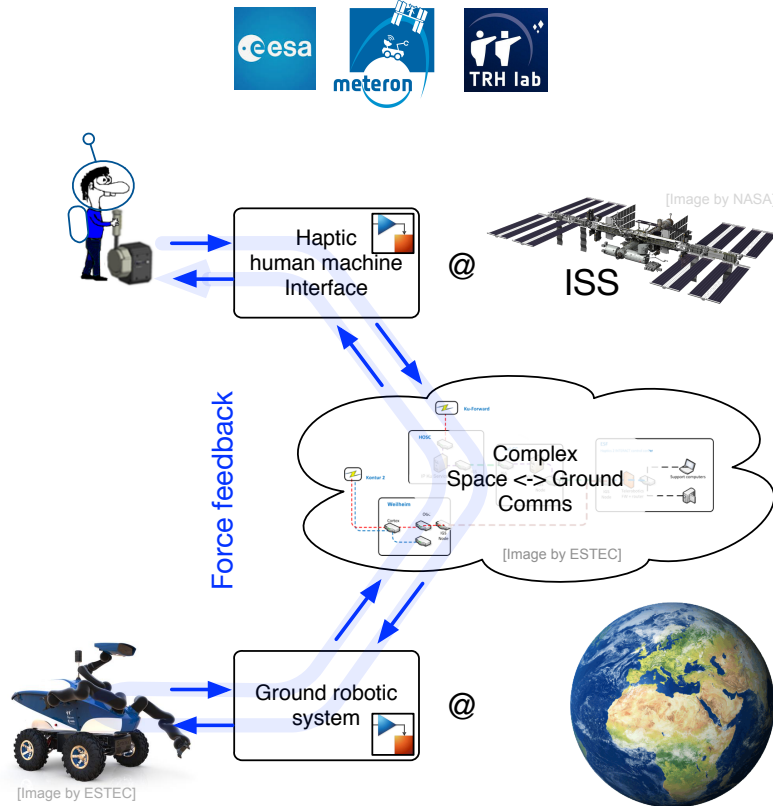


our main focus



our main focus

Recent project:



Robot control system:

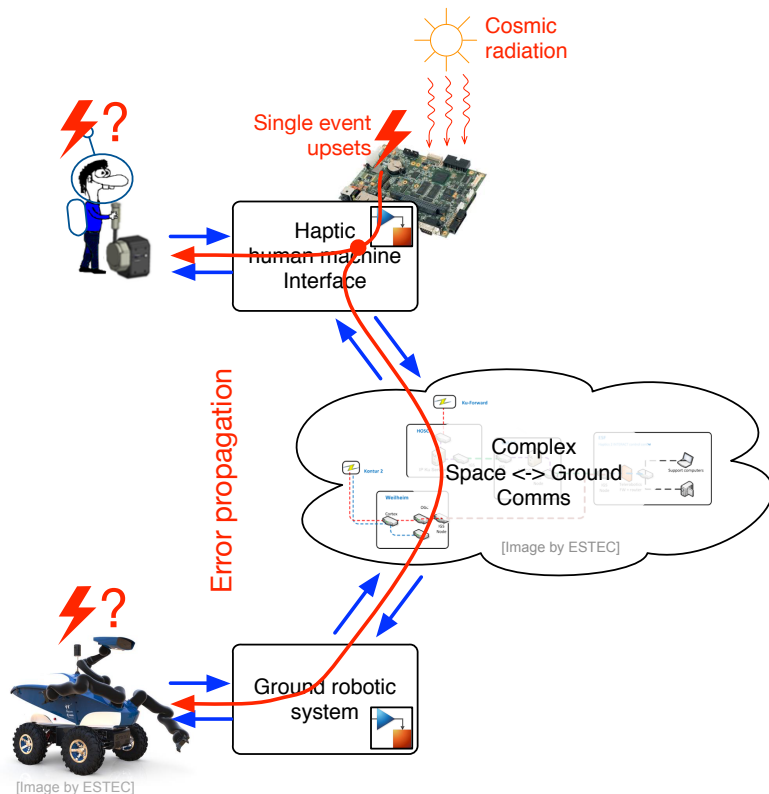
- Distributed, space to ground
- Haptic telemanipulation
- Software, hardware, network
- Model-based design with Simulink

Astronaut Andreas Mogensen
controls a ground rover from space
07.09.2015.

[Video from ESA YouTube channel]

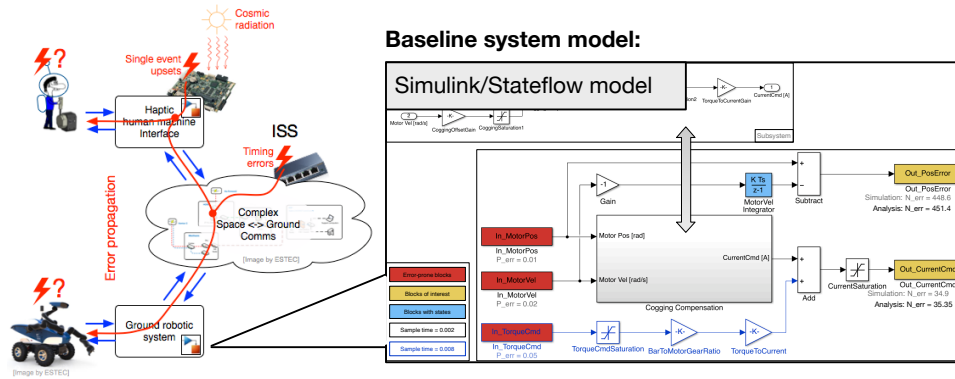
Fault model:

Single event upset (SEU) – an ionizing particle hits a micro electronic device (CPU, RAM) causing a bit-flip, and leading to silent data corruption (**data error**).



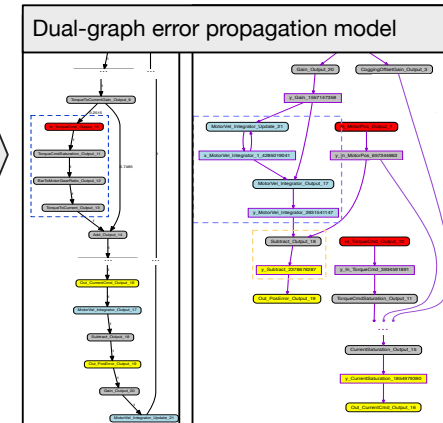
SEUs detected in Columbus Mass Memory Units of the International Space Station.

[Ivano Verzola, Anne-Emmanuelle Lagny, and Janos Biswas, A Predictive Approach to Failure Estimation and Identification for Space Systems Operations. SpaceOps 2014 Conference. Pasadena, CA.]



Automated Transformation
Parsing
Instrumentation
Partial simulation

Analytical model:



Error propagation analysis:

- Central part: stochastic dual-graph error propagation model (DEPM).
- Automatic generation of the DEPM from different baseline system models including Simulink.
- Optimized, automatic computation of reliability metrics of the system based on the DEPM.

Markov chains
Petri nets
Temporal logic

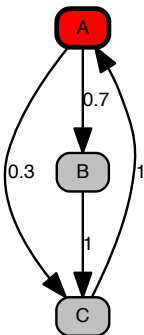
Automated Computation

Reliability and (Safety) metrics

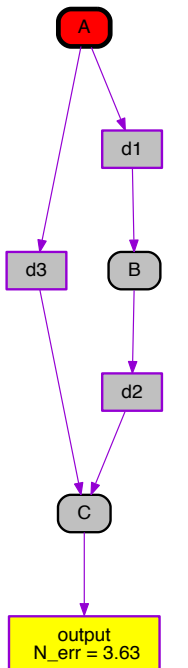
e.g. mean number of errors in critical system outputs during given execution time

Dual-graph error propagation model (DEPM)

Control flow graph



Data flow graph



[A. Morozov and K. Janschek. Probabilistic error propagation model for mechatronic systems. Mechatronics, 2014.]

Component-level reliability properties

Conditions of element A
 if (True):
 then (d1=ok)&&(d2=ok) with pr 0.1
 then (d1=error)&&(d2=error) with pr 0.9

Conditions of element B
 if (d1==ok):
 then (d2=ok) with pr 1.0
 if (d1==error):
 then (d2=ok) with pr 0.9
 then (d2=error) with pr 0.1

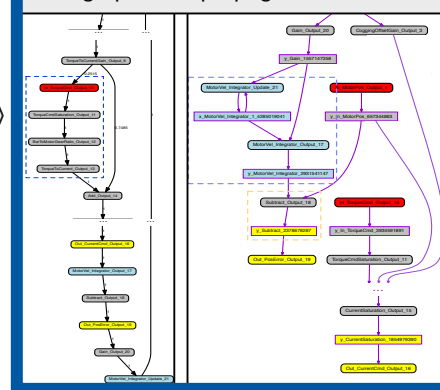
Conditions of element C
 if (d2==ok)&&(d3 == ok):
 then (output=ok) with pr 1.0
 if (d2==error)|| (d3 == error):
 then (output= error) with pr 0.8
 then (output= ok) with pr 0.1

Mean number of errors in data storage output: 3.63

Automated Transformation
 Parsing
 Instrumentation
 Simulation

Analytical model:

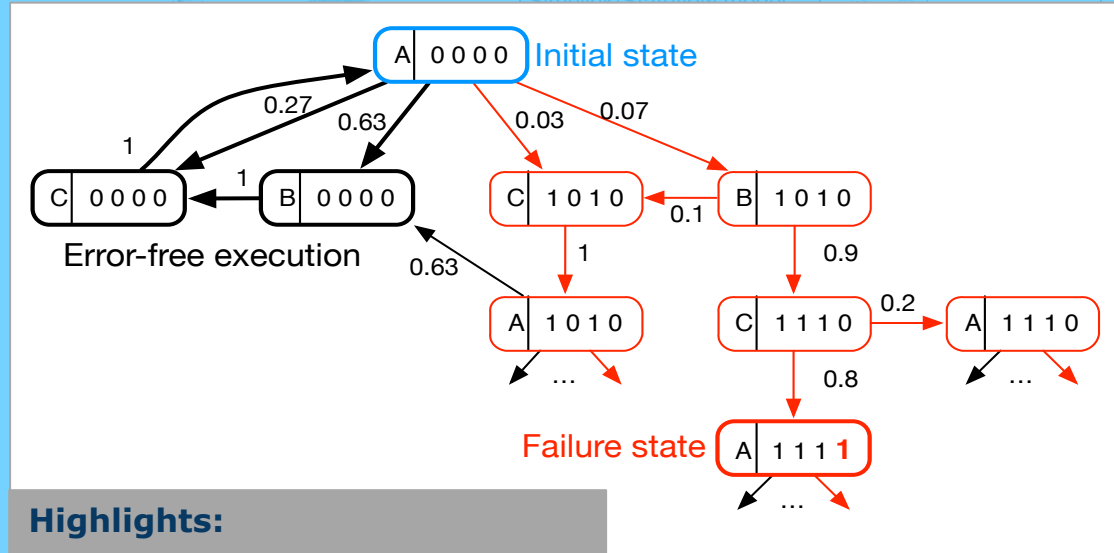
Dual-graph error propagation model



Highlights:

- Probabilities of:
 - control flow transitions
 - faults activation
 - errors propagation
- Cycles in control and data flow graphs
- Parallel processes
- Complex hierarchical models

Discrete time Markov models (DTMC)

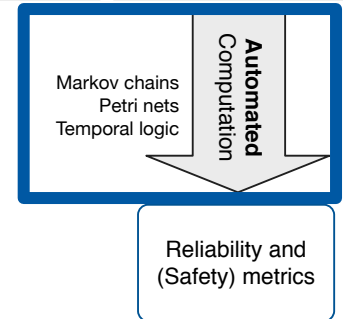
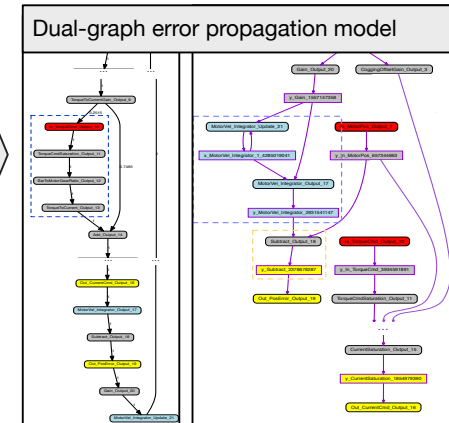


- Highlights:**
- Automatic generation and computation of DTMC models
 - Petri net models for timing and parallel processes
 - Optimizations: Fast solvers, nesting, data flow slicing

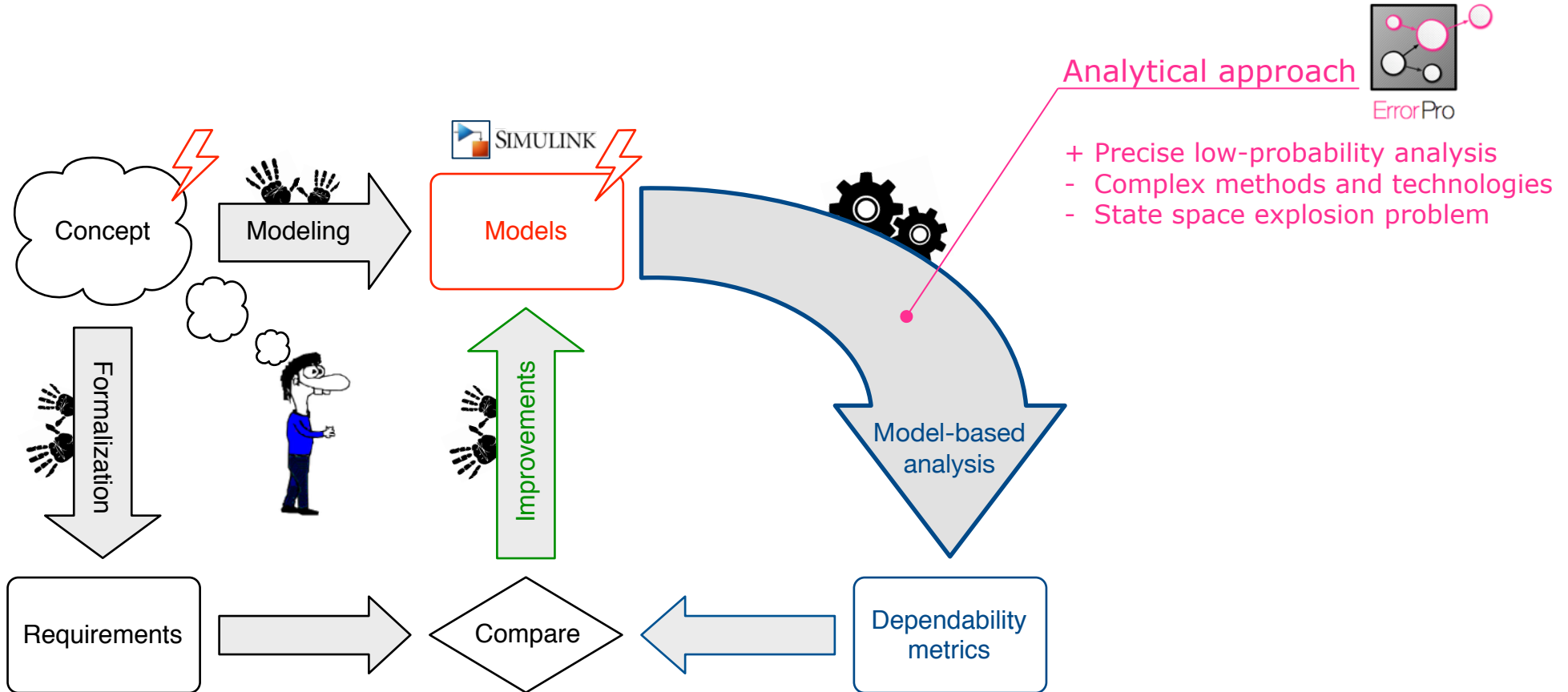
[A. Morozov and K. Janschek. Dual graph error propagation model for mechatronic system analysis, 2011]
 [A. Morozov and K. Janschek. Probabilistic error propagation model for mechatronic systems. Mechatronics, 2014.]

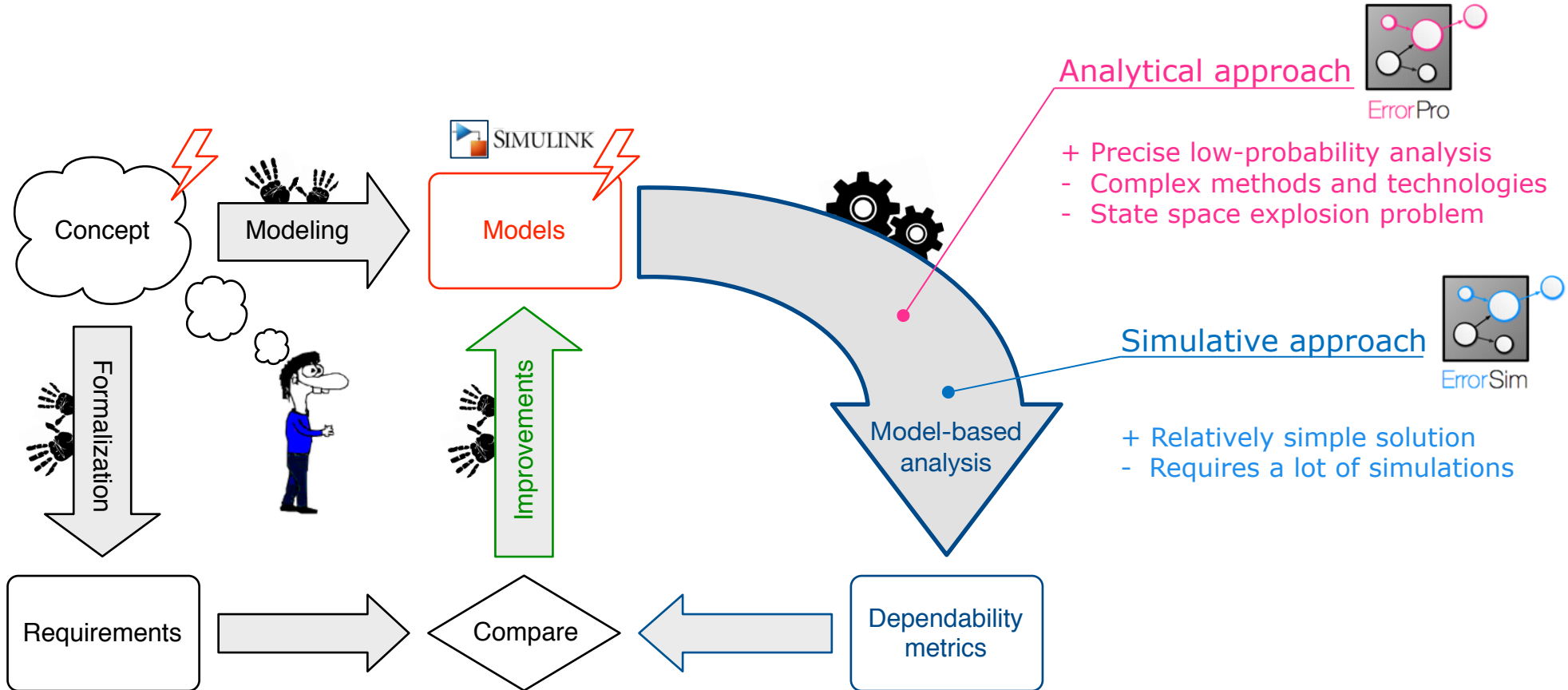
Automated Transformation
 Parsing instrumentation simulation

Analytical model:



e.g. mean number of errors in critical system outputs during given execution time





Introduction

- Model-based system analysis
- Dependability and error propagation
- Analytical and simulative approaches

ErrorSim

- Workflow
- Fault types and injection methods
- Reported statistical information

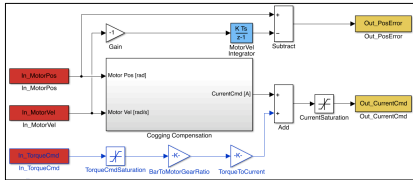
Case study

- Reference Simulink model
- Experiments
- Result interpretation

2

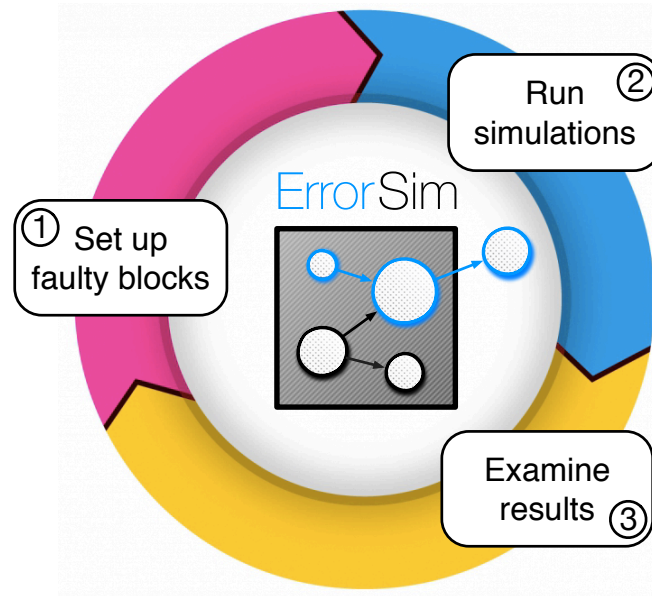
Input:

Simulink model with highlighted blocks



Red faulty blocks, errors are injected in the outputs of these blocks.

Yellow blocks of interest, statistical information is gathered for these blocks.



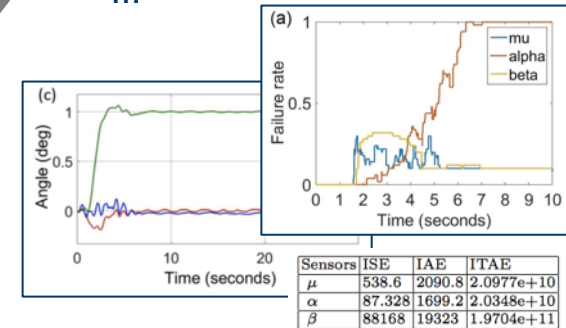
MATLAB, Simulink API, GIUDE

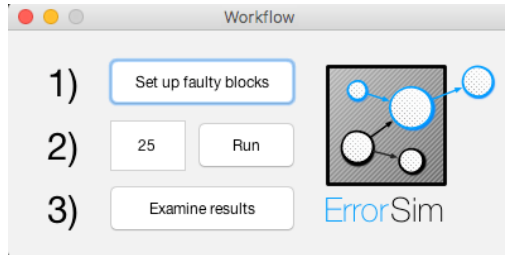
Output:

Reliability metrics for the blocks of interest:

- Error plots
- Number of errors
- Failure rates

...





Fault types

- Sensor faults
- Hardware faults
- Network faults

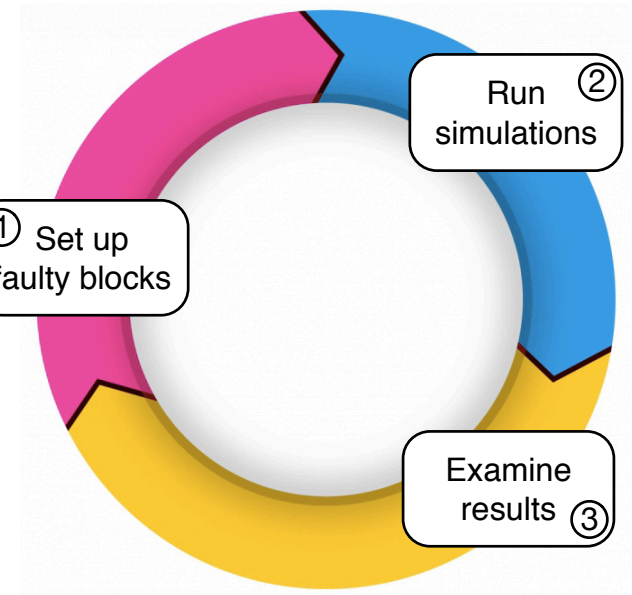
Injection methods

Event (when):

- Failure probability
- Mean Time To Failure (MTTF)
- Failure rate distribution

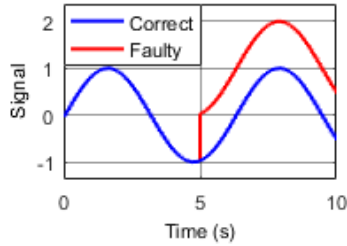
Effect (how long):

- Once
- Constant time
- Infinite time
- Mean Time To Repair (MTTR)

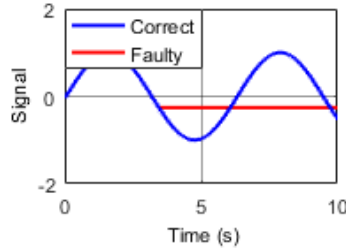


Fault types following the IEC 61508 standard

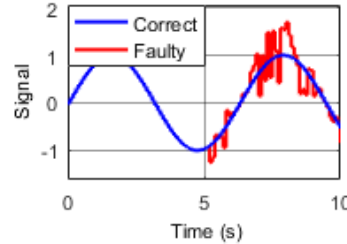
Sensor: Offset



Stuck-at fault



Noise



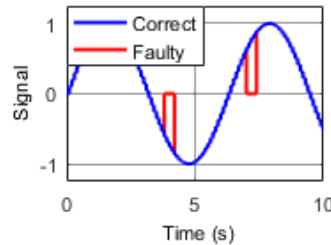
Hardware: Bit flips

01001010 = 74

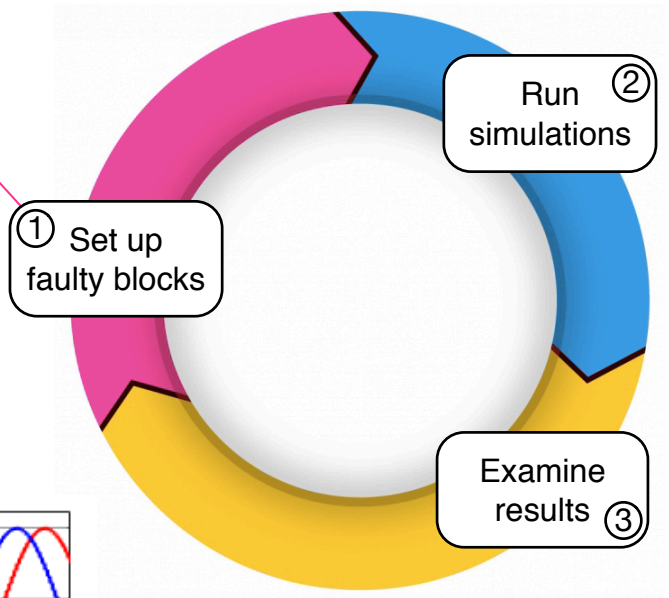
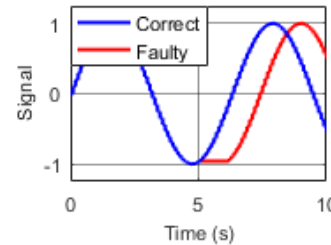


01101010 = 106

Network: Package drop

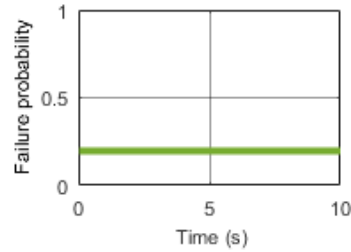


Delay

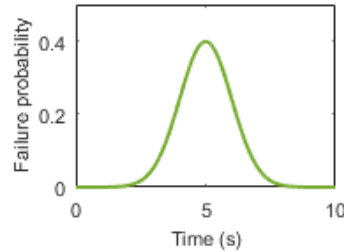


Injection methods based on FIDES and Mil-HDBK-217

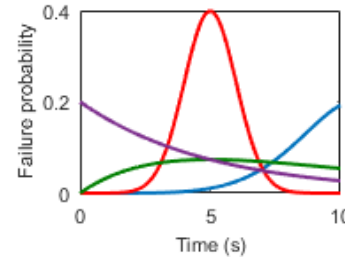
Event: Failure probability



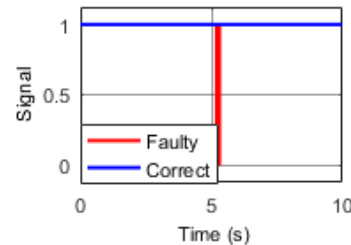
MTTF



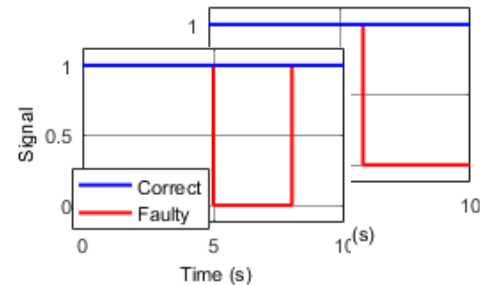
Distribution



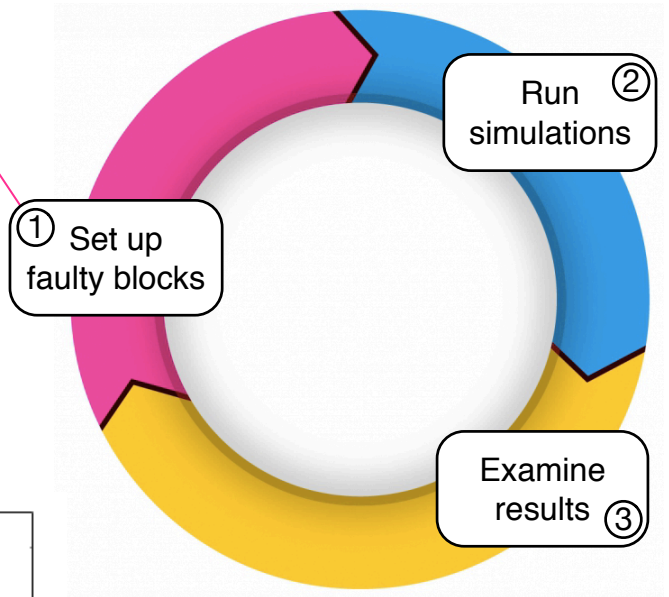
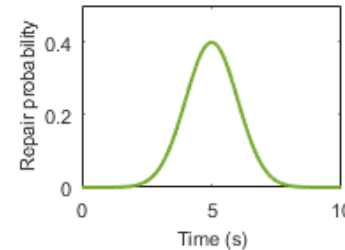
Effect: Once

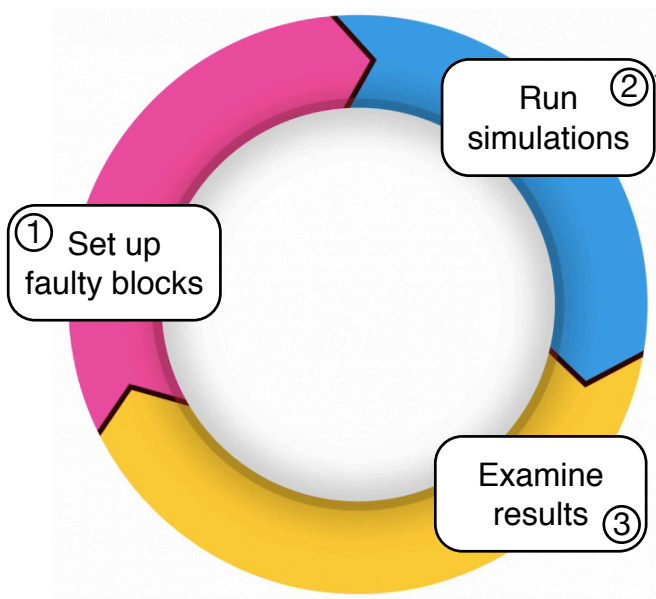


Constant/infinite time



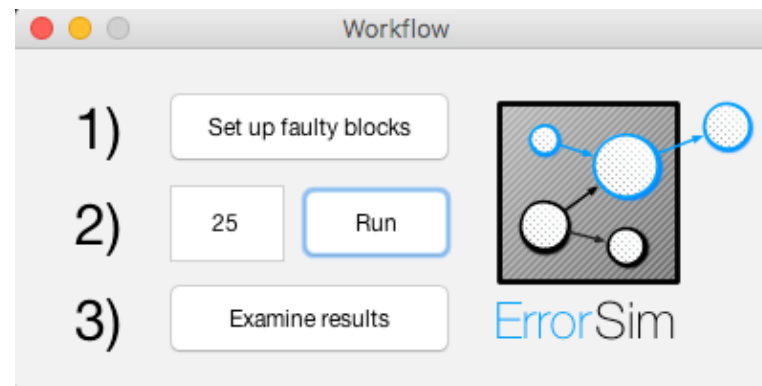
MTTR

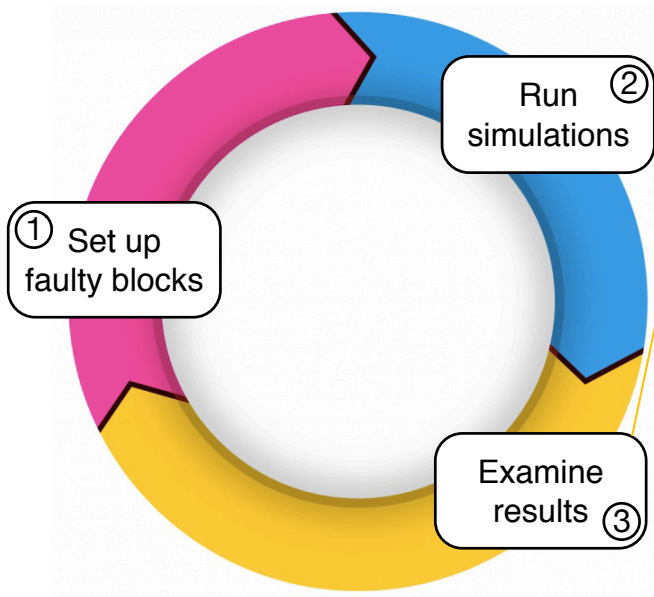




Technical details

- Correct run
- N runs with error injections
- Model instrumentation using block function callbacks of the Simulink API





Statistical information

Signal values:

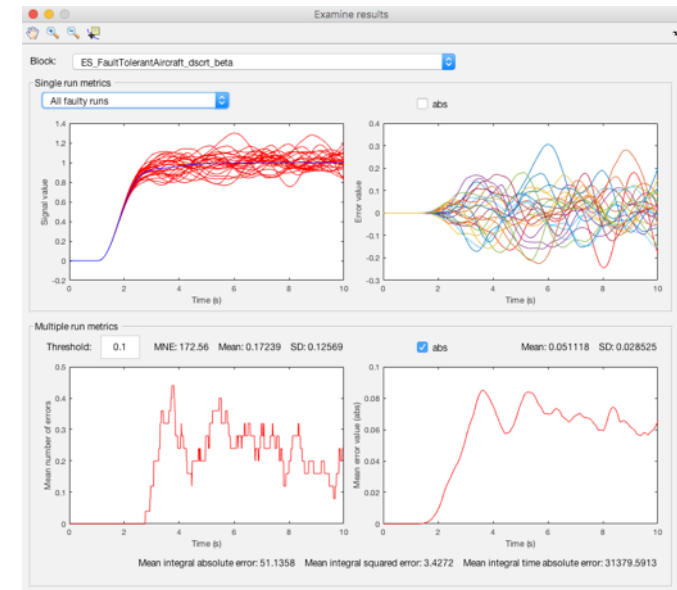
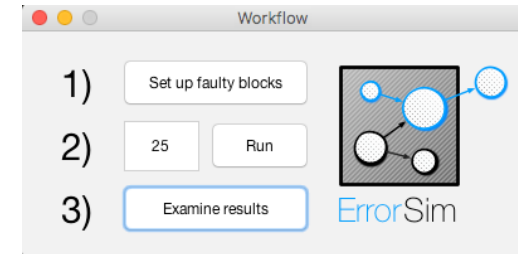
- Correct
- Faulty
- Error (residual)

Reliability metrics:

- Mean number of errors and its time distribution
- Mean error value and its time distribution

Performance indices:

- Integral squared error
- Integral absolute error
- Integral time absolute error



Introduction

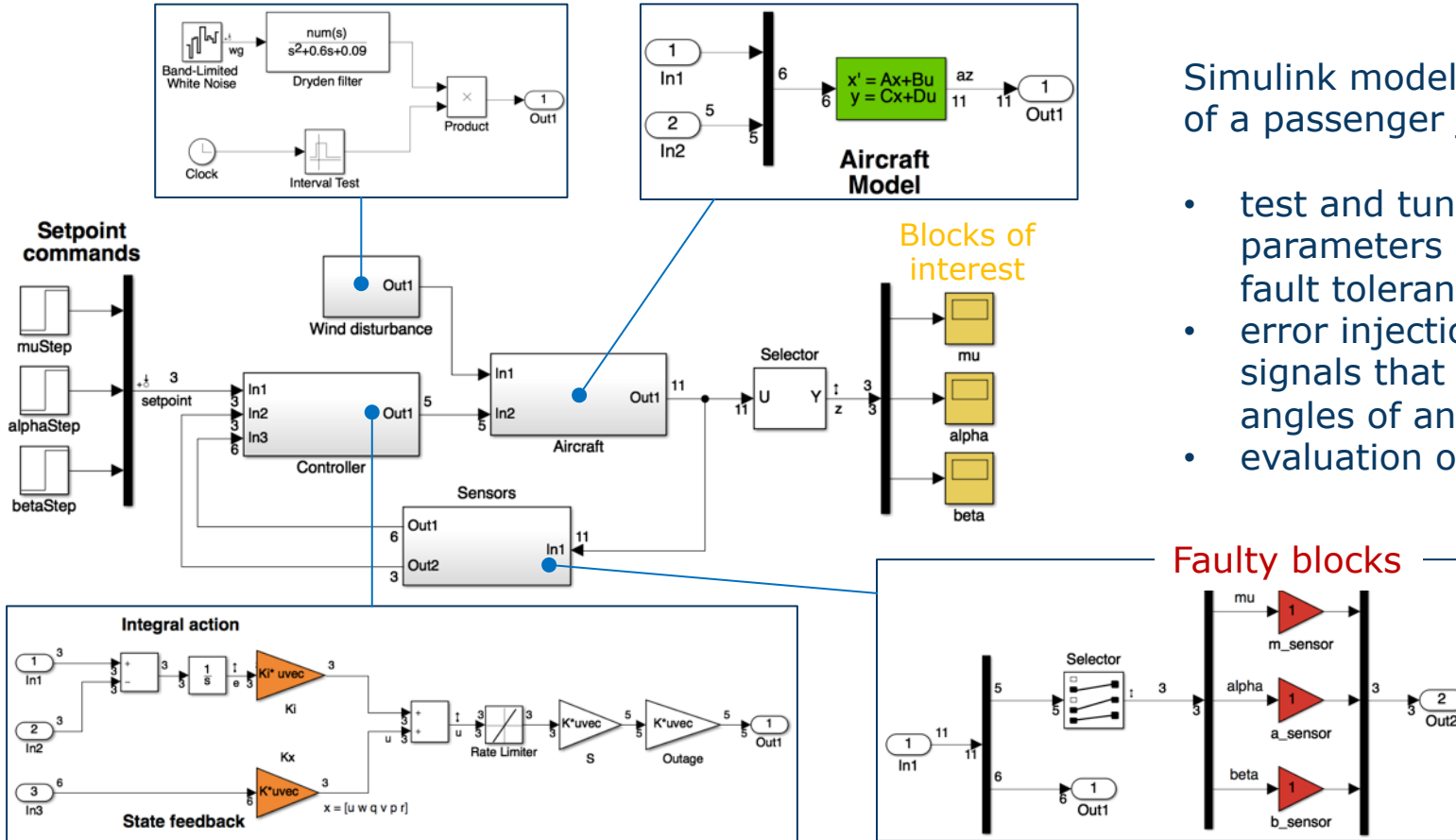
- Model-based system analysis
- Dependability and error propagation
- Analytical and simulative approaches

ErrorSim

- Workflow
- Fault types and injection methods
- Reported statistical information

Case study

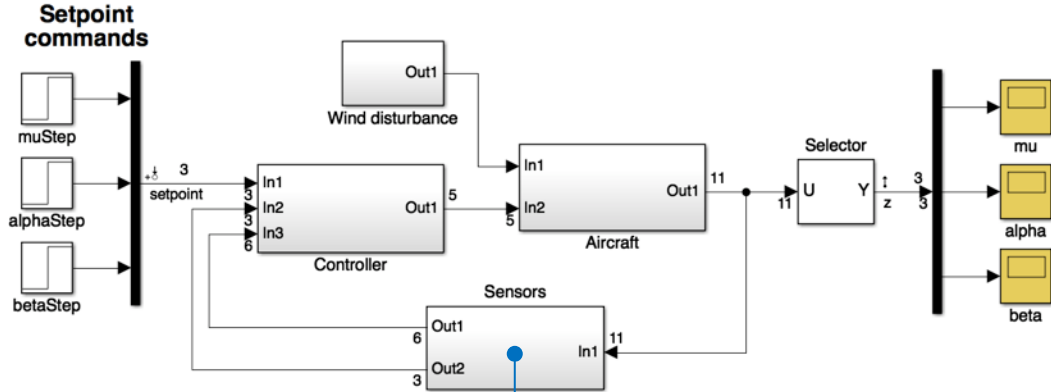
- Reference Simulink model
- Experiments
- Result interpretation



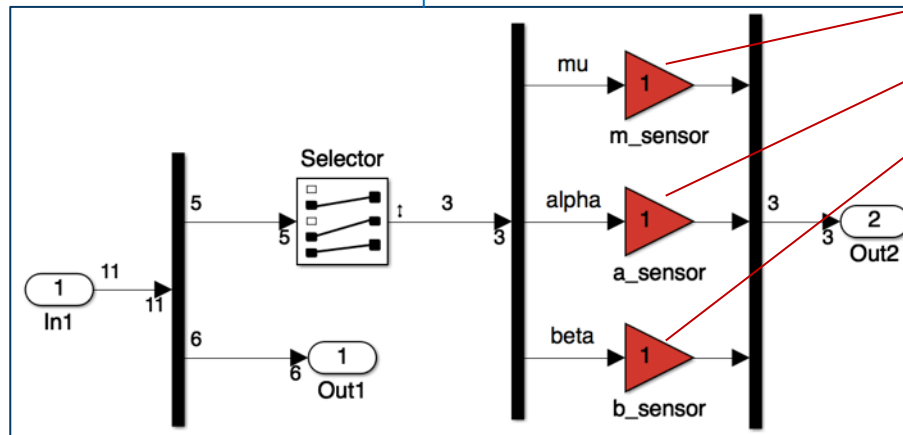
Simulink model of a fault-tolerant control of a passenger jet:

- test and tune, state feedback parameters in order to achieve better fault tolerance;
- error injection into three sensor signals that represent μ , α , and β angles of an aircraft;
- evaluation of errors in critical outputs.

[Mathworks: Fault-tolerant control of a passenger jet - Matlab Simulink example, 2016]
 [M. Saraoğlu, A. Morozov, M. Turan Soylemez and K. Janschek. ErrorSim: a Tool for Error Propagation Analysis of Simulink Models, SafeComp 2017]



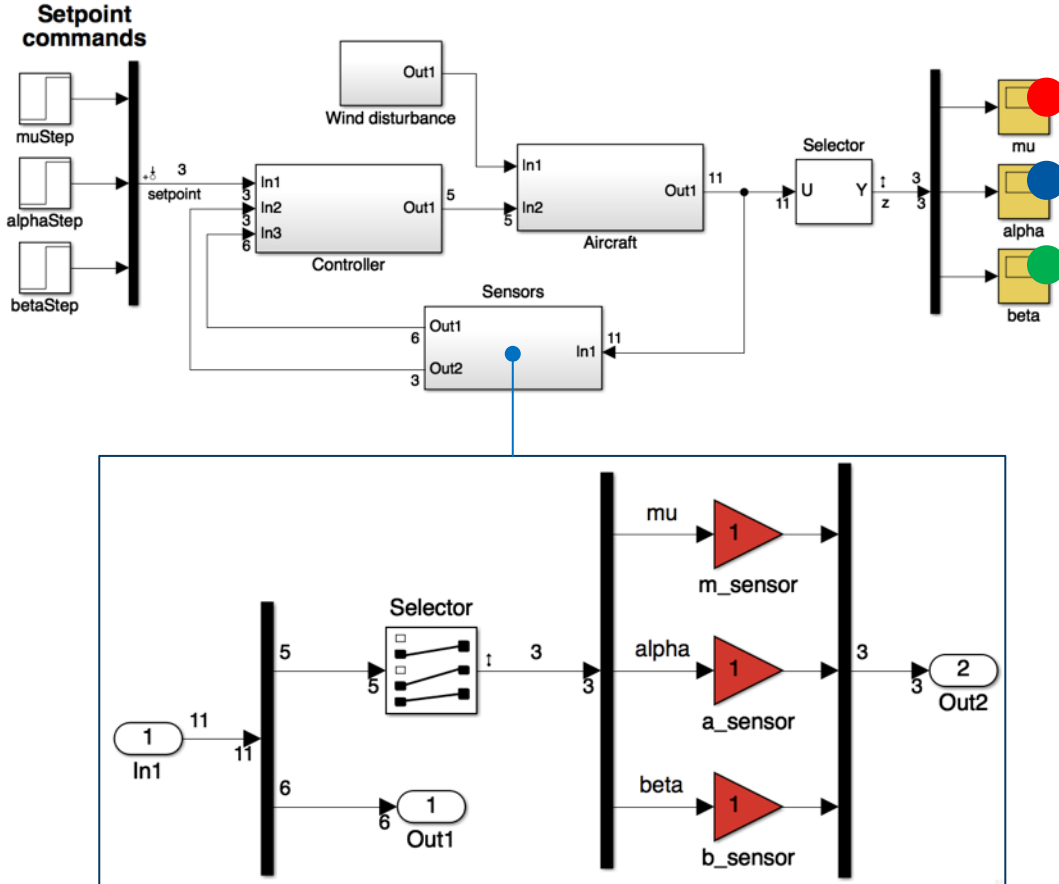
Three experiments with various fault types and injection methods from light to more severe:



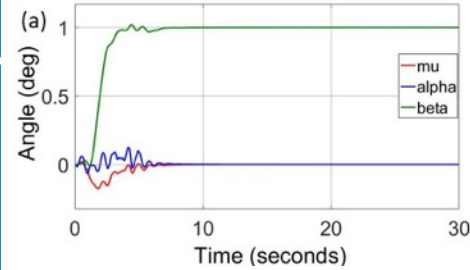
Experiment 1			
Sensors	Fault type	Fault injection method	
		Event	Effect
μ	Oscillation, %10	Failure probability, 0.03	Constant time, 1s
α	Offset, +0.05	MTTF, 10s	Constant time, 0.2s
β	Stuck-at fault	Failure probability, 0.05	MTTR, 2s

Experiment 2			
Sensors	Fault type	Fault injection method	
		Event	Effect
μ	Oscillation, %20	Failure probability, 0.1	Constant time, 1s
α	Offset, +0.05	MTTF, 4s	Constant time, 1s
β	Stuck-at fault	Failure probability, 0.05	MTTR, 2s

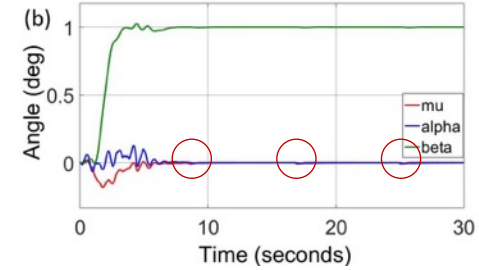
Experiment 3			
Sensors	Fault type	Fault injection method	
		Event	Effect
μ	Oscillation, %20	Failure probability, 0.1	Constant time, 1s
α	Offset, +0.05	MTTF, 4s	Constant time, 1s
β	Stuck-at fault	Failure probability, 0.05	MTTR, 5s



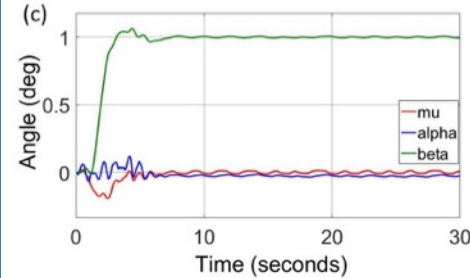
Correct run:



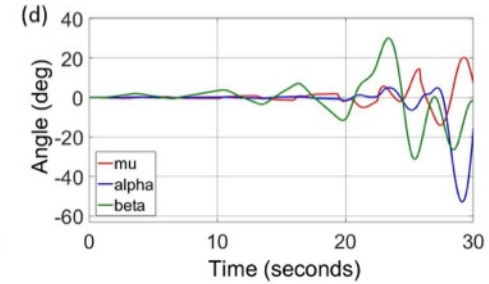
Experiment 1:
Eventual deteriorations



Experiment 2:
Minor continuous
deteriorations



Experiment 3:
Unstable behavior



Achieved results:

- New lightweight tool for error analysis of Simulink models
- Various fault types based on IEC 61508
- Various fault injection methods based on FIDES and Mil-HDBK-217
- Reported reliability metrics suitable for RAMS analysis of a complete system

Future challenges:

- Performance evaluation and optimization
- Automatic report generation for FTA/FMEA
- Integration into industrial development processes.

Thank you!

We are looking for partners for further development and integration of the ErrorSim into industrial processes.

andrey.morozov@tu-dresden.de

Research team:



Klaus
Janschek



Andrey
Morozov



Thomas
Mutzke
Siemens
Healthineers



Kai
Ding



Mustafa
Saraoglu
Erasmus



Mikael
Streuer
Hochschule
Nordhausen

Diploma students and interns:

2017: Boqi Ren, Fangmeng Zhu

2016: Yue Long, Tao Chen,

2015: Yu Zhou, Fusheng Zhao,

2014: Regina Tuk, Wei Zhang, Arsenii
Rashidov, Jin Li, Yao Li ...