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
Outlook

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
1 Introduction: Model-based system analysis

Simplified **model-based system development** workflow:
1 Introduction: Model-based system analysis

Conventional **system analysis:**

- **Concept**
- **Modeling** → **Models**
- **Autocode**
- **Real system**

**Formalization**

- **Requirements** → **Compare**
- **Testing** → **Improvements**
- **Metrics**
Model-based system analysis:
To recognize system characteristics and inherent shortcomings in the early design stages in order to provide optimized design solutions.
Dependability: The trustworthiness of a system such that reliance can justifiably be placed on the service it delivers.

1 Introduction: Dependability and error propagation

**Dependability**

**Attributes**
- **Safety**: absence of catastrophic consequences;
- **Reliability**: continuity of correct service;
- **Confidentiality**
- **Integrity**
- **Maintainability**

**Means**
- **Fault Tolerance**: to avoid service failures in the presence of faults;
- **Fault Prevention**
- **Fault Removal**
- **Fault Forecasting**

**Threats**
- **Faults**: a defect in the system;
- **Errors**: incorrect internal state;
- **Failures**: incorrect delivered service;

**System A**
- Fault → Error → Failure

**System B**
- Fault

**Terminology** Laprie[1992]

- **Dependability**
- **Fault Prevention**
- **Fault Removal**
- **Fault Forecasting**
- **Fault Tolerance**: to avoid service failures in the presence of faults;
- **Availability**: continuity of correct service;
- **Reliability**: absence of catastrophic consequences;
- **Confidentiality**
- **Integrity**
- **Maintainability**

**Our main focus**
1 Introduction: Dependability and error propagation

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]
1 Introduction: Dependability and error propagation

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.

1 Introduction: Dependability and error propagation

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.
1 Introduction: Dependability and error propagation

Dual-graph error propagation model (DEPM)

Control flow graph

Data flow graph

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

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]
1 Introduction: Analytical and simulative approaches

Concept -> Modeling -> Models -> Model-based analysis

Requirements -> Formalization -> Improvements

Analytical approach

ErrorSim: A Simulator for Error Propagation Analysis of Control Systems Developed in Simulink

[ErrorSim: A Simulator for Error Propagation Analysis of Control Systems Developed in Simulink]

1 Introduction: Analytical and simulative approaches

Analytical approach

+ Precise low-probability analysis
- Complex methods and technologies
- State space explosion problem
1 Introduction: Analytical and simulative approaches

Analytical approach
- Precise low-probability analysis
- Complex methods and technologies
- State space explosion problem

Simulative approach
- Relatively simple solution
- Requires a lot of simulations

Concept → Modeling → Models

Requirements → Compare → Model-based analysis

Formalization → Improvements

Dependability metrics
Outline

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
**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.

**Output:**

Reliability metrics for the blocks of interest:

- Error plots
- Number of errors
- Failure rates

**ErrorSim:**

- A Simulator for Error Propagation Analysis of Control Systems Developed in Simulink

**MATLAB, Simulink API, GIUDE**
ErrorSim: Fault types and injection methods

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)
ErrorSim: Fault types and injection methods

**Fault types** following the IEC 61508 standard

- **Sensor**:
  - Offset
  - Stuck-at fault
  - Noise

- **Hardware**:
  - Bit flips

- **Network**:
  - Package drop

**Hardware**

- **01001010 = 74**
- **01101010 = 106**

**ErrorSim**

- A Simulator for Error Propagation Analysis of Control Systems Developed in Simulink

1. Set up faulty blocks
2. Run simulations
3. Examine results
**Event:** Failure probability

- **MTTF**
  - Distribution
  - Failure probability over time

- **Effect:** Once
  - Constant/infinite time
  - Signal over time
  - MTTR
  - Repair probability over time

**Injection methods** based on FIDES and Mil-HDBK-217

1. Set up faulty blocks
2. Run simulations
3. Examine results
ErrorSim: Workflow

1) Set up faulty blocks
2) Run simulations
3) Examine results

Technical details

- Correct run
- N runs with error injections
- Model instrumentation using block function callbacks of the Simulink API
ErrorSim: Reported statistical information

1. Set up faulty blocks
2. Run simulations
3. Examine results

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
Outline

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
Case study: Reference Simulink model

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 $\mu$, $\alpha$, and $\beta$ angles of an aircraft;
- evaluation of errors in critical outputs.

[Mathworks: Fault-tolerant control of a passenger jet - Matlab Simulink example, 2016]

Case study: Experiments

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

<table>
<thead>
<tr>
<th>Experiment 1</th>
<th>Sensors</th>
<th>Fault type</th>
<th>Fault injection method</th>
<th>Event</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>Oscillation, %10</td>
<td>Failure probability, 0.03</td>
<td>Constant time, 1s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>α</td>
<td>Offset, +0.05</td>
<td>MTTF, 10s</td>
<td>Constant time, 0.2s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>Stuck-at fault</td>
<td>Failure probability, 0.05</td>
<td>MTTR, 2s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 2</th>
<th>Sensors</th>
<th>Fault type</th>
<th>Fault injection method</th>
<th>Event</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>Oscillation, %20</td>
<td>Failure probability, 0.1</td>
<td>Constant time, 1s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>α</td>
<td>Offset, +0.05</td>
<td>MTTF, 4s</td>
<td>Constant time, 1s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>Stuck-at fault</td>
<td>Failure probability, 0.05</td>
<td>MTTR, 2s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 3</th>
<th>Sensors</th>
<th>Fault type</th>
<th>Fault injection method</th>
<th>Event</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>Oscillation, %20</td>
<td>Failure probability, 0.1</td>
<td>Constant time, 1s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>α</td>
<td>Offset, +0.05</td>
<td>MTTF, 4s</td>
<td>Constant time, 1s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>Stuck-at fault</td>
<td>Failure probability, 0.05</td>
<td>MTTR, 5s</td>
<td></td>
</tr>
</tbody>
</table>
Case study: Experiments

ErrorSim: A Simulator for Error Propagation Analysis of Control Systems Developed in Simulink
**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
Thomas Mutzke
Mustafa Saraoglu

Andrey Morozov
Kai Ding
Mikael Streuer

Siemens Healthineers
Erasmus

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 …