# ExCuSe – A Method for the Model-Based Safety Assessment of Simulink and Stateflow Models

MATLAB Expo 2018 | 2018-06-26 | München Julian Rhein





- Introduction
- Property Proving
- Application to Safety Assessment
- Implementation
- Demo
- Summary & Outlook



#### Introduction

- Functional safety assessment is an integral part of software and systems development according to SAE ARP 4754A
- Task of functional safety assessment
  - Establish relations between component faults and system failure conditions
  - Validation & Verification of safety requirements

#### **Typical methods:**

- Failure Modes and Effects Analysis (FMEA)
  - Determine system level effects from (single) component faults
- Fault Tree Analysis (FTA)
  - Find possible causes for system failure conditions



**Our goal:** Partially automated generation of safety assessment artefacts from annotated models:













Julian Rhein – Model-based Safety Assessment 6 ✓ Support of analysts during repetitive, error-prone tasks

Provides methods to system designers and software engineers to evaluate their designs prior to formal safety assessment

Enables modularity of safety assessment and reusability of artefacts

✓ Additional validation of manual assessment results



# Introduction to Property Proving

- Traditionally used in software verification
- Formal approach to prove that a property is satisfied or violated by a system
- Properties are disproved by counterexamples
- Infinite state and continuous systems are treated by inductive proving and SMT satisfiability
- Powerful free/open-source solvers available





## Safety Assessment as Property Proving Problem



#### **Formal concept**

Fault injection

 $M(u) \Rightarrow M^*(u,f), f \in \mathcal{F}$ 

Cut-sets computation

 $CS \coloneqq \{ cs_i \in \mathcal{F} \mid M^*(u^*, f^*) \not\vDash AG \mathcal{P} \land \exists j \ s. t. f_{i,j} \}$ 

Minimal cut-sets

 $MCS := \{ cs \in CS \mid \nexists cs' \in CS \land cs' \subset cs \}$ 

- Verification:  $|\mathcal{F}| = 0$  ,  $|\mathcal{P}| > 0$
- FMEA:  $|\mathcal{F}| = 1$ ,  $|\mathcal{P}| > 0$

• FTA:  $|\mathcal{F}| > 0$ ,  $|\mathcal{P}| = 1$ 



#### Intuitive explanation

- Extension of the model by additional inputs to trigger fault events
- Computation of sets (combinations) of failures, which lead to a requirements violation from the counterexamples
- Minimal combinations of failures, i.e. failure configurations that are necessary for the occurrence of system failures
- Verify that the system fulfills all requirements in the failure free case
- Determine all possible effects (i.e. violation of requirements) of single failures
- Determine all possible causes of single a single effect (i.e. its MCSs)



 Extension of the model by additional component failure models and additional inputs to trigger the activation and deactivation of component faults





### Implementation

 Implementation in Simulink/Stateflow by fault injection interface

Idea:

- Extend the model by a nondeterministic layer
- Provide modelling facilities for failure logic modeling or failure injection
- Provide interface for automatic analysis
- Share failure flow information between components without requiring additional signals
- Allows common cause modeling
- Predefined and custom fault models
- Automatic cut-set analysis based on the Simulink Design Verifier property proving function

	Fault Inlec	tor
	ID:Failure	e
Block Parameters: Faul	t Injector	
Generic Failure Injecto	r	
nes in order to inject f predefined failure m Main Custom Fail	faults to that signals. Th odes or define custom f ure Modes Probabilis	e user can choose from a ailure modes. tic Attributes
D: Failure		
Generic Failure Mode	25	
Active Mode: Stuck		
	Min	Max
✓ Bias	0	: 1
✓ Stuck		
	Driftrate Min	Max
✓ Stuck ✓ Drift	Driftrate Min 0	Max
Stuck  Constant From	Driftrate Min 0 Constant Min	Max i 1 Max
<ul> <li>✓ Stuck</li> <li>✓ Drift</li> <li>✓ Constant Error</li> </ul>	Driftrate Min 0 Constant Min 0	Max i 1 Max i 1
Stuck  Constant Error  Constant Error	Driftrate Min 0 Constant Min 0 Pass-through Consta	Max 1 Max 1 1 Coverage
Stuck  Constant Error  Detected Failure	Driftrate Min 0 Constant Min 0 Pass-through Consta	Max 1 Max 1 1 Coverage 0.99
Stuck  Constant Error  Detected Failure	Driftrate Min 0 Constant Min 0 Pass-through Consta 0 Min	Max 1 Max 1 1 Coverage 0.99 Max



### **Generic Failure Models**

















- Enable definition of arbitrary, user-defined fault models
- Uncertain parameters can be modelled using special source blocks

		Custom lanu		noueis				
demo	o_mdl ×	Loss of Effectiveness	$\times$	Custom FM 2	$\times$	Custom FM 3	$\times$	
۲	🎦 demo_i	mdl 🕨 눰 Fault Injector	►	Loss of Effective	eness			•
Ð								
アン								
⇒		1						
ΑΞ			•	×				
$\sim$		1: [0.2, 1]	•		<u> </u>			
		Uncertain						
01		parameter						
Read	у			100%		Fi	xedSte	pDiscrete

	ID:Failure
🚡 Block Pa	rameters: Fault Injector
Generic Fa	ilure Injector
ines in or of predefin	ler to inject faults to that signals. The user can choose from a se red failure modes or define custom failure modes.
	Effectiveness
Namo	Loss of Effortivoposs
Onon	
Open	n new tab
Delete	I new window
▼ Custon	2 FM 2
Name	Custom FM 2
Open	n new tab
Open i	n new window
Delete	
Custon	n FM 3
Add custo	m failures modes and click



- Distributions of the component lifetime and repair time
- Similar to basic event models in fault tree analysis
- Common models are built-in:
  - Exponential distribution
  - Weibull distribution
  - Periodic test
- Custom models can be specified as custom expressions or histograms





	ID:Failure	
🚡 Block	k Parameters: Fault Injector	
Generic	: Failure Injector	
ines in of prede	order to inject faults to that signals. The user can choose fr efined failure modes or define custom failure modes.	om a
Main	Custom Failure Modes Probabilistic Attributes	
Model:	Exponential	
🗹 Repa	a Weibull	
Model	Periodic test	
Failure	Custom (Expression)	
1e-5	Custom (Distribution Object)	
Repair	ir rate (1/h):	
1e-1		
Te-1 Failure	e on demand probability:	
1e-1 Failure 1e-6	e on demand probability:	
1e-1 Failure 1e-6 Test c	e on demand probability:	
1e-1 Failure 1e-6 Test c	e on demand probability: coverage:	







- Example problem: A320 SEC decision logic for ground spoiler deployment
- Hazards:
  - Inadvertent full/partial spoiler deflection inflight
  - Missing full/partial spoiler deflection on ground
- Question: Which combinations of sensor failures can potentially cause the hazards





### **Demo: Ground Spoiler Deployment Logic**







#### Summary

- Model checking provides a powerful method for model-based safety assessment
- Cut-set analysis can be expressed as property proving problem
  - o Performance enhancement by incremental search
  - Anytime approximation of probability boundaries
- Successful integration in Simulink/Stateflow

#### Outlook

- Scalability considerations
- Creation of structured fault trees from the minimal cut-sets
- Using structural analysis to obtain initial guess of the minimal cut-sets
- Extension to undirected models



### Thank you for your attention!

### **Questions?**







#### Julian Rhein Research Associate

Lehrstuhl für Flugsystemdynamik / Institute of Flight System Dynamics Technische Universität München Boltzmannstraße 15 D-85748 Garching Germany

Phone:+49 (89) 289-16051Fax:+49 (89) 289-16058Email:julian.rhein@tum.de

