A Safety Analysis via Minimal Path Sets Detection for Object-Oriented Models

C. Schallert
German Aerospace Centre, Institute of System Dynamics and Control, Wessling, Germany

ABSTRACT

Safety is of prime importance for aircraft and must be considered from the beginning of the design process. Conducting a safety analysis involves considerable effort, particularly if traditional methods, such as manual fault tree analysis, are used. This may explain why the analyses are performed usually only once or twice during an entire aircraft development. Furthermore, traditional methods are hardly linked to other tools for system conceptual design.

In the field of modelling and simulation, multi-domain object-oriented languages are increasingly adopted. They enable an intuitive way of modelling, since objects and their interconnections correspond with real components. Thus, large-scale system models can be created and simulated efficiently. This has led to the appearance of model libraries for various physical domains, such as mechanics, electrics and hydraulics. Recently, application specific libraries emerge, such as flight dynamics and control, actuation or energy systems.

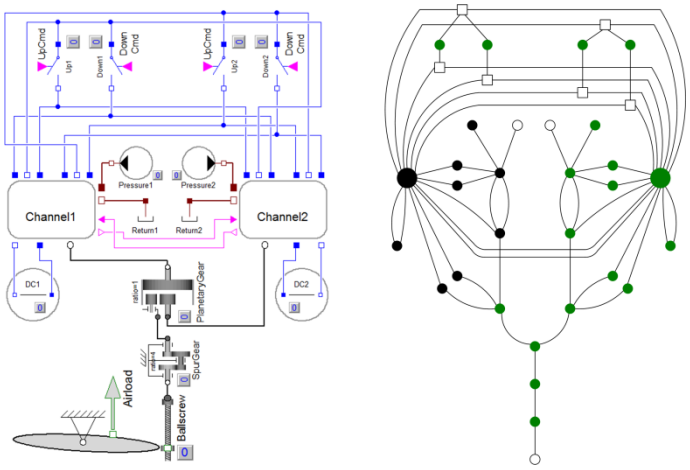

Figure1. Stabiliser trim system model and corresponding graph

A new method is developed that integrates safety analysis with multi-domain object-oriented modelling. This is the contribution aimed by the corresponding paper. First, it recaps the basics of the modelling approach. Then, a model of a safety relevant aircraft system, a stabiliser trim control and actuation system, is established using component models from generic libraries that are supplemented with failure behaviour. Next, a method is developed that automatically detects the minimal path sets of a technical system based on the corresponding object-oriented model, thus performing a safety analysis. Techniques from graph theory are adopted for computational efficiency and feasibility of the method.

The minimal path sets detection method is exemplified by means of the established realistic system model.

REFERENCES

Airbus Training. *A320 Flight Crew Operating Manual, section 1.27.00 - Flight Controls*. Available at website: http://www.smartcockpit.com/

Boeing B737 NG - *Systems Summary - Flight Controls*. Available at website: http://www.smartcockpit.com/

Bouissou, M. & Elmqvist, H. & Otter, M. & Benveniste, A. 2014. Efficient Monte Carlo simulation of stochastic hybrid systems. *Proceedings of the 10th international Modelica conference,* Lund, Sweden, March 10-12, 2014.

Elmqvist, H. 1978. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Lund Institute of Technology, Lund, Sweden.

Elmqvist, H. & Mattsson, S. E. & Otter, M. 2014. Modelica extensions for Multi-Mode DAE Systems. *Proceedings of the 10th international Modelica conference,* Lund, Sweden, March 10-12, 2014.

Heidtmann K. D. 1989. Smaller Sums of Disjoint Products by Subproduct Inversion. *IEEE Transactions on Reliability* (Vol. 38, No. 3): pp. 305 - 311.

Krumke, S. O. & Noltemeier, H. 2012. *Graphentheoretische Konzepte und Algorithmen*. Vieweg+Teubner Verlag, 3. Auflage. In German.

The Modelica Association 2000. *Modelica – A Unified Object-Oriented Language for Physical Systems Modeling*. Available at website: https://modelica.org/documents/ModelicaTutorial14.pdf/

Persson, U. & Schallert, C. 2001. *The 728 JET flight control system*. Deutscher Luft- und Raumfahrtkongress, Hamburg, Germany, September 2001.

Roberts, B. & Kroese, D. P. 2007. Estimating the number of s-t paths in a graph. *Journal of Graph Algorithms and Applications* (Vol. 11, No. 1): pp. 195 - 214.

Schallert, C. 2014 (to appear). *Integration of Safety and Reliability Analysis Methods with Object-Oriented Modelling*. PhD thesis, Technische Universität Berlin, Germany.

A Safety Analysis via Minimal Path Sets Detection for Object-Oriented Models

C. Schallert
German Aerospace Centre, Institute of System Dynamics and Control, Wessling, Germany

ABSTRACT: Safety is of prime importance for aircraft and must be considered from the beginning of the design process. Conducting a safety analysis involves considerable effort, particularly if traditional methods, such as manual fault tree analysis, are used. This may explain why analyses are performed usually only once or twice during an entire aircraft development. Furthermore, traditional methods are hardly linked to other tools for system conceptual design.

In the field of modelling and simulation, multi-domain object-oriented languages are increasingly adopted. They enable an intuitive way of modelling, since objects and their interconnections correspond with real components. Thus, large-scale system models can be created and simulated efficiently.

A new method is developed that integrates safety analysis with multi-domain object-oriented modelling. This is the contribution aimed by this paper. In essence, the proposed method automatically detects the minimal path sets of a technical system.

1 INTRODUCTION

This paper describes a method for automated safety analysis of a technical system that is represented as a multi-domain object-oriented model. The proposed method automatically detects the minimal path sets of the modelled system. Then, the probability of system operation or failure is computed from the minimal path sets using component failure rates.

Object-oriented modelling based on differential algebraic equations offers large expressive power for establishing multi-disciplinary engineering tools. In addition, the structure of an object-oriented model (OOM) resembles the functional paths that cause a technical system to operate. This property is exploited by the minimal path sets detection method.

Fault modelling is introduced in addition to the common modelling of normal operation. This forms a basis for the proposed method that belongs to the class of state space simulations. In this context, the state space denotes the set of all possible combinations of intact and failed components of a system. Search algorithms from graph theory are adopted to narrow down the state space, thus preventing unfeasibility of the method due to an exponential increase of the number of combinations and thus simulations to be performed.

The developed modelling approach is not exclusively dedicated to safety analysis. It rather serves several purposes, such as system performance or parameter studies, as usually is the motivation for physical modelling and simulation.

This paper is organised as follows: Section 2 recaps the basics of OOM, describes the selected modelling approach and establishes an example model of a safety relevant aircraft system. Section 3 describes the minimal path sets detection and exemplifies the method by means of the established system model. Section 4 concludes the paper.

2 MULTI-DOMAIN OBJECT-ORIENTED MODELLING OF SAFETY RELEVANT AIRCRAFT SYSTEMS

2.1 *Multi-domain object-oriented modelling*

This section briefly recaps the basic principles of the modelling approach.

The idea of multi-domain object-oriented modelling was formulated by Elmqvist (1978), who proposed a language called Dymola (Dynamic Modelling Language). In object-oriented modelling, objects, their boundaries and interconnections correspond with real existing equipment. Physical equations have to be established or understood only for each object. By connecting objects with each other, further model equations are introduced. A translator then automatically transforms the model equations into a simulation ready runtime model. This approach has a couple of advantages over block dia-

grams (e.g. Simulink), as explained by the two corresponding model implementations of a small electronic circuit shown in Figure 1.

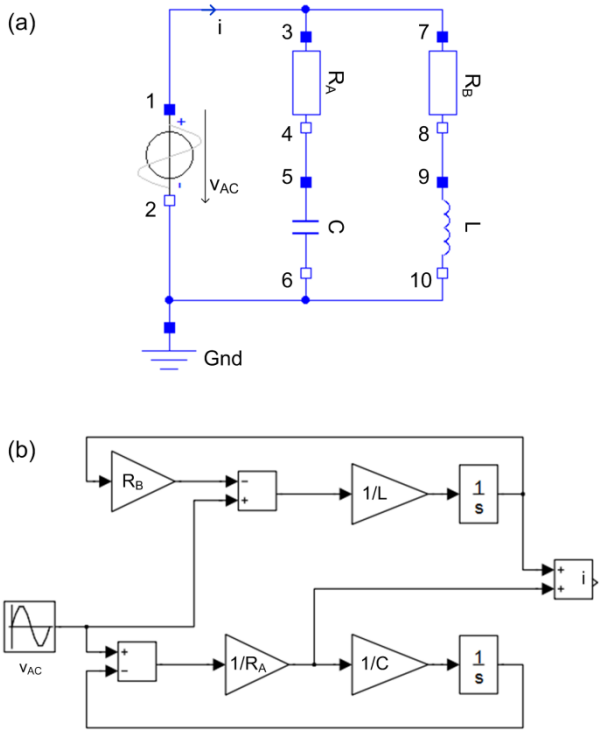

Figure 1. Small electronic circuit (a) and its block diagram representation (b). Taken from The Modelica Association (2000).

As Figure 1 shows, the OOM (a) retains the physical structure of the circuit, whereas the block diagram (b) does not. The OOM is changed intuitively by removing or adding components and connecting them with other components. Different physical domains can be combined in a single model.

The circuit model includes, among others, a resistor, capacitor and inductor. Such component models are available from standard libraries. The model equations are $v = R \cdot i$, $i = C \cdot der(v)$ and $v = L \cdot der(i)$, respectively, where *der()* denotes the time derivative. Each component has two electric pin interfaces (filled and non-filled blue squares in Figure 1 a) that include the voltage v as a potential and current i as a flow variable. Model equations are introduced for connected component (object) interfaces as follows: The potential is the same, whereas the sum of flow equals zero according to Kirchhoff's node rule. For example, in case of the circuit model shown by Figure 1 a): $v_1 = v_3 = v_7$ and $i_1 + i_3 + i_7 = 0$.

The system model described in section 2.3 also covers the mechanical (M) and hydraulic (H) domains, as well as real and Boolean signals (S). Table 1 summarises the according interface definitions and icons. These appear in Figures 1 (a), 3, 4 and 5.

Thus, an OOM consists of the equations of each object and of those imposed by the connections between them. By symbolic processing, this set of equations is translated into a simulation runtime model of the differential algebraic (DAE) form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{B} \cdot \boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C} \cdot \boldsymbol{x}(t) + \boldsymbol{D} \cdot \boldsymbol{u}(t) \end{aligned} \quad (1)$$

Table 1. Model interfaces, icons and domain (dom) identifiers

connector	potential	flow	icon	dom
electric pin	voltage v	current i	■ □	E
translational flange	position s	force f	■ □	M
rotational flange	angle φ	torque τ	● ○	M
hydraulic port	pressure p	flow rate q	■ □	H
real signal	input u, output y		▶ ▷	S
Boolean signal	dto.		▶ ▷	S

2.2 *Modelling additions for safety analysis*

2.2.1 *Modelling and simulation of failures*

The proposed safety analysis method is based on simulation of failures. Component models that only reflect normal operation have to be enhanced accordingly. At system level, control or reconfiguration logics have to be implemented that define how a system reacts to the occurrence of failures. The pursued modelling approach is structure invariant, i.e. the DAE structure (equation 1) remains always the same. Failures are represented by model parameter (values in matrices $\boldsymbol{A}$, $\boldsymbol{B}$) changes.

Elmqvist et al. (2014) propose and Bouissou et al. (2014) use a structure variant approach that is even more flexible for the modelling of failures. Future work will combine this approach with the minimal path sets detection method described in this paper.

In general, the concept of minimal path sets distinguishes only between operation and failure. Thus, the proposed detection method addresses two states (intact, failed) at component and at system level.

2.2.2 *Indication of system status*

Safety assessment requires the analyst to define criteria that distinguish normal operation from failure of a system. The analysis method proposed by this paper requires implementing these criteria in a system model, such that an output signal indicating the system status is computed. A system model is simulated by the analysis method for various combinations of intact and failed components, and the resulting system status (operation or failure) is correlated with the respective component states.

2.2.3 *Insertion of component failure rates*

Failure rates λ are stored in each component model that is enhanced with failures. Constant failure rates (exponentially distributed lifetimes) are assumed per default. Since the stress level of a component is known in the simulation, its failure rate can be adapted accordingly. Failure rates are used to compute probability of system operation $R_{sys}(t)$ or failure $F_{sys}(t)$ from the detected minimal path sets.

2.3 *Stabiliser trim system model*

This section describes the multi-domain object-oriented model of a stabiliser trim control and actuation system. This model of a safety relevant aircraft system is based on generally available information (Airbus Training, Boeing B737 NG, Persson et al. 2001). It is established for exemplification of the automated minimal path sets detection method.

2.3.1 *System description*

Longitudinal control of an aircraft is achieved by elevators and a horizontal stabiliser surface. The latter is positioned by a stabiliser trim control and actuation system. A main part of this system is a ballscrew actuator that is equipped with two hydraulic motors to rotate the ballscrew through a reduction gear. If the stabiliser is not moved, it is held in place by the actuator. To this end, an electro-mechanical, power-off engaged brake (POB) is mounted to each motor shaft. The brakes are engaged if the associated motor is shut off or if the system has to be stopped after a failure is detected.

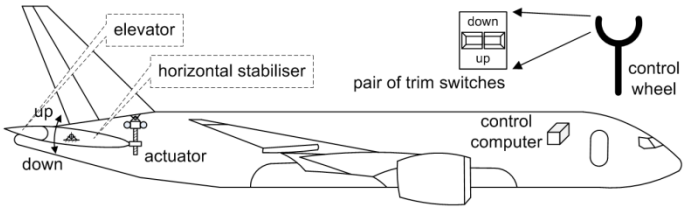

Figure 2. Stabiliser trim control and actuation system

The pilots operate the system via pairs of trim switches that are mounted to each control wheel (or centre pedestal). The trim switches are spring-loaded to the neutral position. To command trimming, the two switches of a pair have to be pushed up or down simultaneously. This causes the active system channel to release the associated brake and energise a hydraulic valve to direct fluid to the motor. The actuator moves the stabiliser until the trim switches are released, the end of the stroke is reached, or a fault is detected. The system is monitored for faults, such as loss of hydraulic pressure, valve or motor failure, by comparing the sensed motor speed and stabiliser motion rate with the actual trim command.

The minimal path sets detection method reveals those conditions that cause a system to operate. The example system is considered operational if it moves the stabiliser surface according to the actual command and holds it in place if no command is present.

2.3.2 *Component modelling*

The modelling of the stabiliser trim system includes one-dimensional mechanics with elasticity, damping, inertia and friction; hydraulics with bulk modulus, fluid density, turbulent and laminar flow; and resistive electrics. One of the developed component models is explained below.

A spur (reduction) gear is essentially described by the ratio of input and output torques and angles:

$$0 = I_{SG} \cdot \tau_a + \tau_b, \qquad \varphi_a = I_{SG} \cdot \varphi_b \tag{2}$$

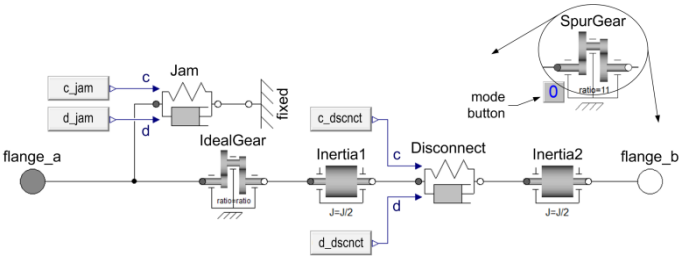

Figure 3. Spur gear model including disconnection and jam

The "IdealGear" element introduces these equations to the spur gear model. The "Disconnect" and "Jam" elements are rotational spring-dampers described by:

$$\tau = c \cdot \varphi_{rel} + d \cdot \dot{\varphi}_{rel}, \qquad \tau = \tau_b = -\tau_a, \qquad \varphi_{rel} = \varphi_b - \varphi_a \tag{3}$$

They are used to represent failures of the spur gear. To this end, the spring and damping coefficients (c and d) are varied as indicated by Table 2. In equations 2 and 3, indices a and b denote the rotational flanges of the spur gear and of the spring-damper elements (shown in Figure 3), respectively.

Table 2. Modelling of spur gear failures by variation of spring and damping coefficients. $[c] = Nm/rad$, $[d] = Nms/rad$

	$mode$	c_{dscnct}	d_{dscnct}	d_{jam}
normal operation	0	10^4	10^4	d_{G-drag}
disconnection	1	0	0	d_{G-drag}
jam	2	10^4	10^4	10^4

In normal operation, the "Disconnect" element is quasi-rigid. It is literally broken to represent a disconnection of the spur gear. A jam is modelled by a quasi-rigid connection to the "fixed" element. The coefficient values implicate small relative rotations in the order of $\varphi_{rel} = 10^{-3} rad$ for a transferred torque of $10Nm$, which is negligible for the present application. The coefficients have to be adapted for applications of different scale. The "Jam" element also reflects the friction of the intact spur gear, e.g. by $d_{G-drag} = 2 \cdot 10^{-4}\, Nms/rad$.

Normal operation ($mode$ = 0) or disconnection ($mode$ = 1) of the spur gear are considered for the detection of minimal path sets.

2.3.3 *System model*

Figure 4 shows an image of the developed stabiliser trim control and actuation system model. It contains two instances of the drive channel model depicted by Figure 5. Each channel consists of a control computer, a hydraulic valve, motor and brake (POB) that are electrically activated. The channels are connect-

ed to different electric (DC1, 2) and hydraulic (Pressure1, 2) sources. Each computer is connected to the trim switches (Up1, 2 and Down1, 2), and to the other channel computer for coordination of active or stand-by status. The model interface and connection types are as defined in Table 1.

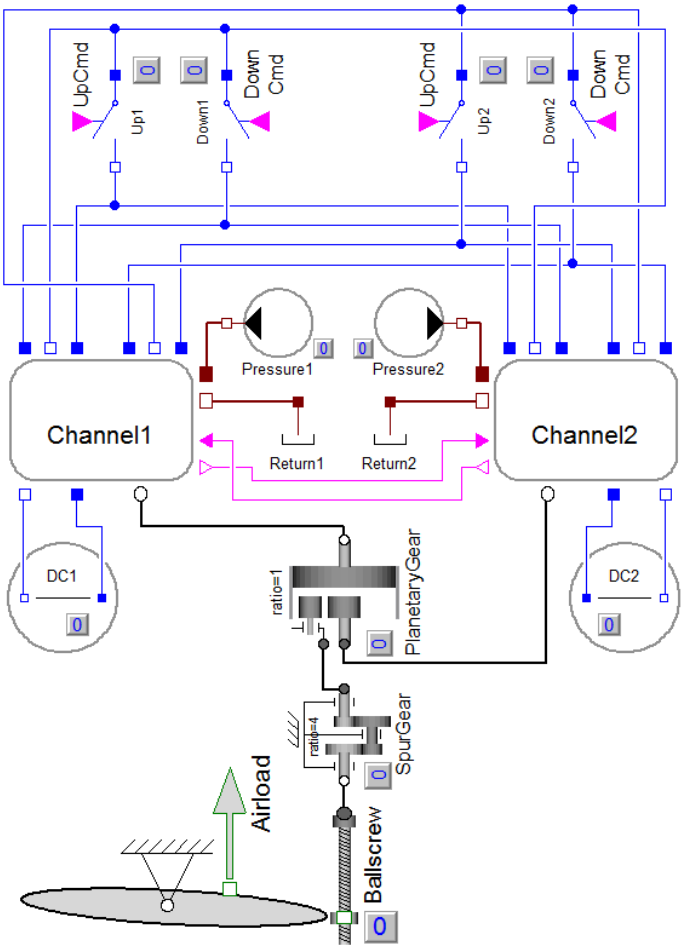

Figure 4. Stabiliser trim control and actuation system model

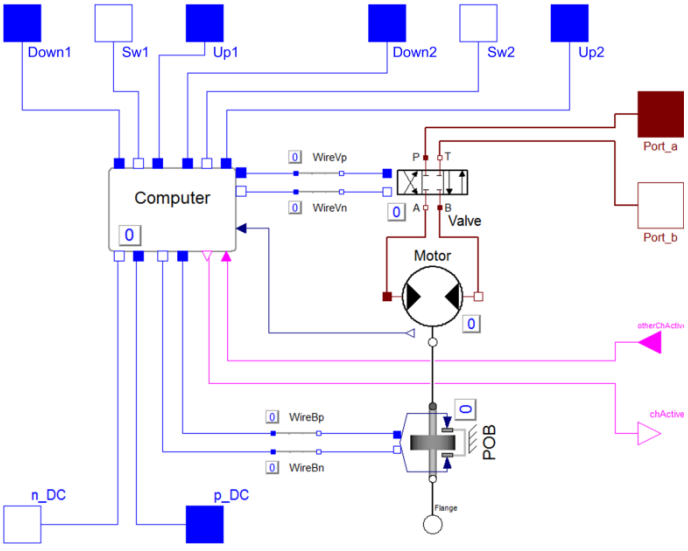

Figure 5. Model of a drive channel of the stabiliser trim system

As explained in section 2.2.2, a system status output (not shown in the images) is computed for use by the minimal path sets detection (section 3). This output indicates if the system operates normally or if it fails (no motion or uncommanded motion). To this end, the actual motion rate of the stabiliser surface is compared with the command (trim switch inputs).

3 DETECTION OF MINIMAL PATH SETS

This section describes the proposed method for automated detection of the minimal path sets of a technical system. It is based on the idea that the related object-oriented model structure resembles the functional paths that cause the system to operate.

After detection of the minimal path sets, probabilities of system operation $R_{sys}(t)$ or failure $F_{sys}(t)$ are computed from failure rates (section 2.2.3) using the method described by Heidtmann (1989).

3.1 *Definitions*

A *path set* is a set of intact components that causes a system to operate. A *path set* is *minimal* if it contains only as many intact components as are necessary for the system to operate.

A *path* is a sequence of consecutive nodes in a graph, where each two successive nodes are connected via an edge.

A *union* (denoted by ∪) of a collection of *paths* consists of all nodes in the collection of *paths*.

3.2 *Properties of minimal path sets in an object-oriented model*

The structure of an OOM is regarded as a graph. Nodes and edges of the graph represent components (objects) and connections between them. Nodes that include failure behaviour are indicated by filled circles in Figures 6, 7 (b), 8, 10 - 15. Non-filled circles indicate nodes that are always intact. Non-filled squares indicate auxiliary nodes.

In such a graph, minimal path sets are not just simple s-t paths. An s-t path connects two nodes “s” and “t” of a graph through an unbranched, open walk:

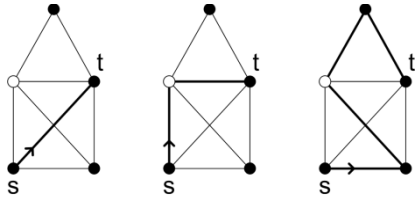

Figure 6. Several s-t paths in a graph

Minimal path sets are sometimes unbranched, just as s-t paths are. In general, however, a minimal path set includes junctions or circles, as Figure 7 indicates for a part of the stabiliser trim system model graph. The electric connections between the computer (15), valve (9) and brake (5) introduce such junctions. (The numbering of nodes is consistent with Figures 11 - 14 and section 3.4.) Since there is more than one junction, circles exist in the graph.

A minimal path set consists of one or more nodes. Power, material or signals are exchanged between neighboured nodes across the connections (edges) between them, as depicted in Figure 7. This causes a technical system to operate. Only neighboured nodes, i.e. coherent sets of nodes, can perform this exchange and render a system operational.

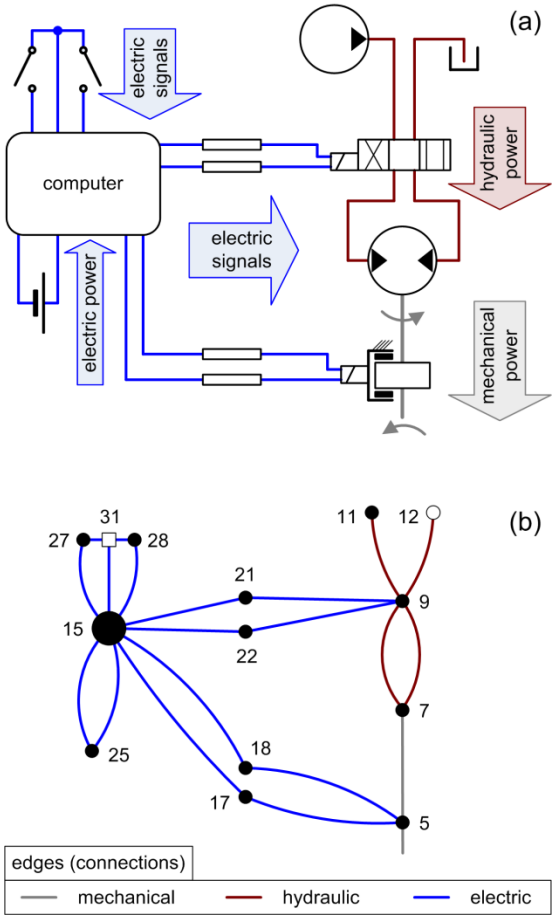

Figure 7. Exchange of power, material and signals across the edges in a coherent set of nodes – Part of OOM of stabiliser trim system (a) and corresponding graph (b)

3.3 *Detection method*

The proposed detection method belongs to the class of state space simulations. At first, the system model graph is evaluated to find candidates of minimal path sets. This is achieved by an adapted depth-first search algorithm and establishing of unions of the found paths. Then, the system model is simulated only for the candidates to detect the actual minimal path sets. Figure 9 shows a flow chart of the detection method. Without the prior evaluation of the model graph, the number of simulations would be 2^{nr}, nr being the number of system components.

3.3.1 *An adapted depth-first search*

Due to the described properties of minimal path sets, it is meaningful to use a depth-first search (DFS) for detection. This algorithm is known from many references, such as Krumke et al. (2012). It finds sets of consecutive nodes (paths) in a graph. In doing so, it commences at a start node and progresses into depth with one of the neighbours of each visited node. For the purpose of detecting minimal path sets, the algorithm is adapted (→ DFSMP) with a definition of the target nodes. It finds the following types of path:

a) Ties. A tie is a path that ends at a node already contained in the path.
b) Circles. A circle is a special case of a tie.
c) Open walks. An open walk ends at a node that only has one neighbour.

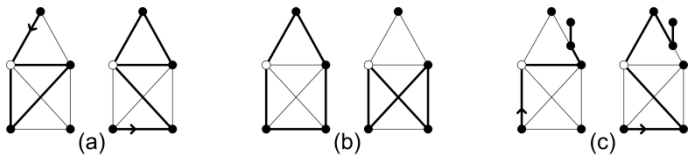

Figure 8. Ties (a), circles (b) and open walks (c) in a graph

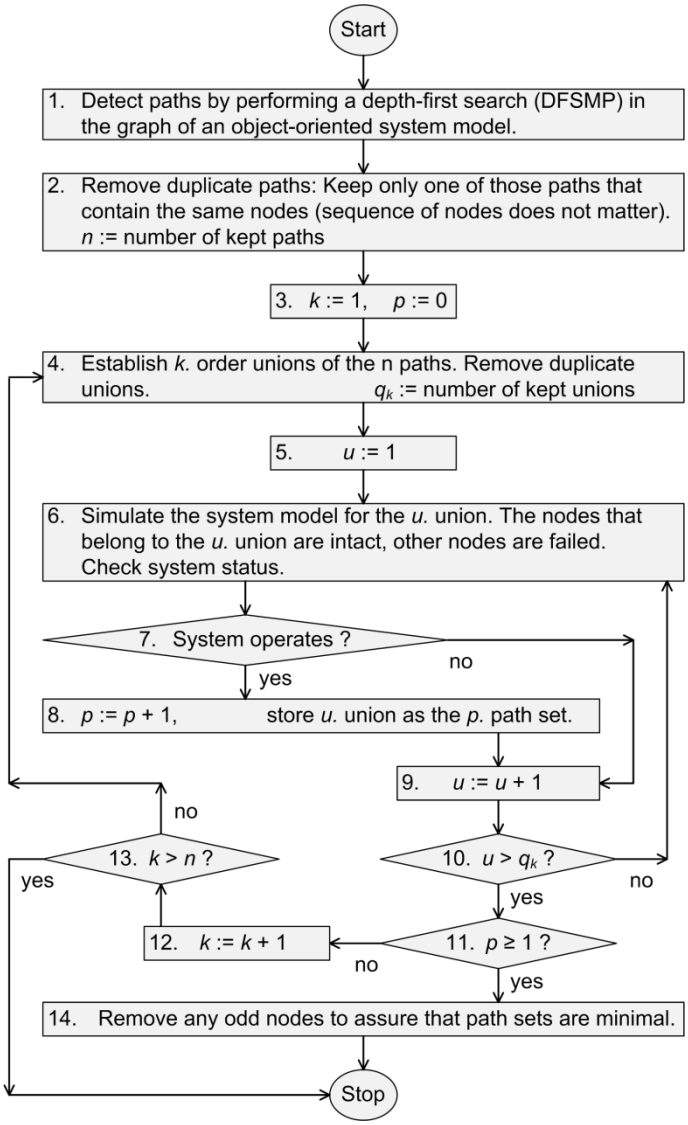

Figure 9. Flow chart of a minimal path sets detection method

3.3.2 *Unions of paths*

The DFSMP is capable of finding minimal path sets that include a junction or circle. Still, a minimal path set might not be detected if the start node is unfavourably selected, or if more than one junction or circle occur. To accommodate for such formations

(present in Figure 7), unions of paths are established. This corresponds with step 4 of Figure 9.

The maximum number of unions is denoted by the binomial

$$\binom{n}{k} = C(n,k) = \frac{n!}{k!\cdot(n-k)!} \quad (4)$$

where n is the number of all paths and k is the number of paths to be merged in a union. The actual number of unions q_k is much lower, dependent on the density d of the graph. d is generally defined, e.g. by Roberts et al. (2007), as

$$d(N,E) = \frac{2\cdot E}{N\cdot(N-1)} \quad (5)$$

For the three graphs depicted in Figure 10, each having $N = 5$ nodes, the number of edges E, density and number of paths n are listed in Table 3. The paths are established starting from node "a" according to the rules defined in section 3.3.1. Since the sequence of nodes in a path or union does not matter, duplicates are removed. n paths or, respectively, q_k unions are kept as summarised in Table 4.

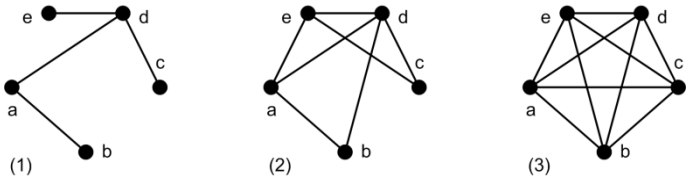

Figure 10. Three graphs of different density

Table 3. Number of edges, density, and number of paths

graph	E	d	n
(1)	4	0.4	3
(2)	7	0.7	5
(3)	10	1	11

Table 4. Binomial (n in rows, k in columns) and actual number of unions q_k for the three graphs

$C(n,k)$	2	3	4	5	q_k	2	3	4	5
3	3	1			3	3	1		
5	10	10	5	1	5	3	2	1	1
11	55	165	330	462	11	5	5	5	1

For the complete graph ($d = 1$), q_k is much lower than the corresponding binomial. The reason is that the paths are longer, the higher the density of the graph is. The longer the paths are, the more they cover the graph. Unions of long paths cover the graph even more, so the number of different unions q_k is relatively low. For a low-density graph, the paths are relatively short, so relatively more different unions (compared to the binomial) occur.

3.3.3 *Removal of odd nodes*

In some cases, the detected path sets are not minimal. This means that more (odd) nodes are contained in a path set than are necessary to cause the system to operate. A path set is minimal only if a failure of any of its nodes (components) causes system failure.

Thus, a check is introduced at the end of the detection method (step 14 in Figure 9): Tracing and removal of odd nodes.

Essentially, the check proceeds by simulating the system model for each path set such that one of its nodes fails, one by one, while the others are intact. If the system still operates, then the respective path set is stored without the failed node. This is repeated until no odd nodes are detected in any path set anymore. Since it is basic trial-and-error, without any evaluation of the model graph, this check is used only to assure that the path sets are minimal by removing a small number of odd nodes, if necessary.

3.3.4 *Separation of physical domains*

The number of paths n in a graph increases strongly with the number of its nodes N and density d, the phenomenon known as combinatorial explosion. This is clarified by an estimation of the number z of s-t paths proposed by Roberts et al. (2007) for graphs of a density $d \in [0.1; 0.9]$. (The DFSMP detects three different kinds of paths. However, estimations exist only of the number of s-t paths.)

$$z(N,d) \approx K(N)\cdot d^{N-1+\delta(N,d)} \quad \text{where}$$

$$K(N) = \sum_{k=0}^{N-2} \frac{(N-2)!}{k!}, \qquad \delta(N,d) = \frac{3.32}{N} - \frac{5.16}{d\cdot N} \quad (6)$$

It is proposed to mitigate the growth phenomenon by splitting up the model graph at the boundaries of the physical domains. These are identified by the different types of model interfaces (listed in Table 1). Thus, a model graph split-up algorithm is created that recognises these interface attributes.

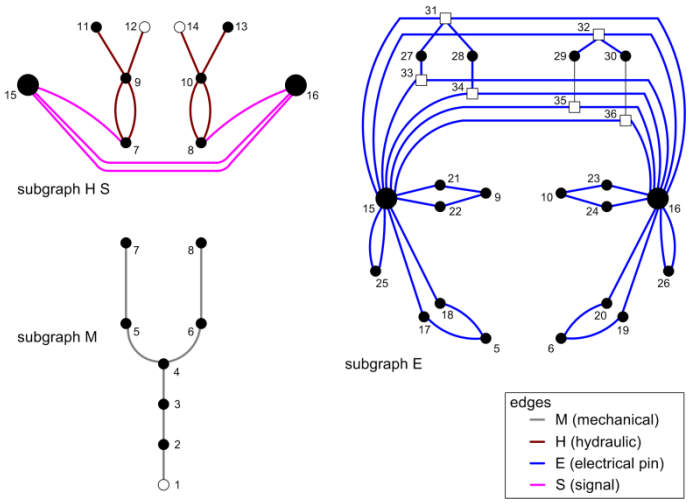

Figure 11. Subgraphs of stabiliser trim system model

Smaller subgraphs are established that are evaluated one by one according to the flow chart of Figure 9. (Due to splitting up the model graph, some additions to the algorithm are needed, as Schallert (2014) describes.) Figure 11 shows the subgraphs established for the stabiliser trim system model. Table 5 lists some of the properties of the model graph

before split-up (all-up) and of the subgraphs. *n* are the actual numbers of paths found in a (sub-)graph.

Table 5. Number of nodes, edges, density, estimated number of s-t paths and actual number of paths in (sub-)graphs

sub-graph	*N*	*E*	*d*	*z*	*n*
all-up	36	54	0.086	1766	1932
M	8	7	0.25	2.4	3
H S	10	9	0.2	2.09	4
E	26	38	0.117	244	91

3.4 *Course of minimal path sets detection algorithm for the stabiliser trim system model*

The minimal path sets detection algorithm is exemplified by means of the stabiliser trim system model.

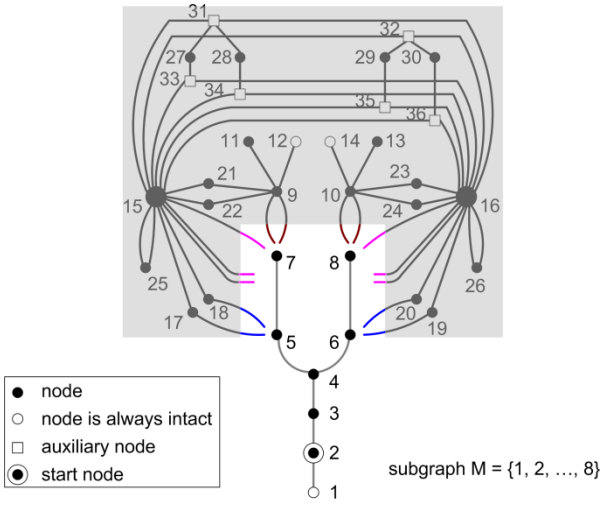

Figure 12. Evaluation of mechanical domain subgraph

Subgraph M is evaluated first, starting from node 2 (ballscrew in Figure 4). Figure 12 shows this stage of the algorithm. The DFSMP finds three paths (parentheses indicate nodes that are always intact):

$path_1$ = {2, (1)}; $path_2$ = {2, 3, 4, 5, 7};
$path_3$ = {2, 3, 4, 6, 8}

The three paths are tested by simulation of the system model (steps 6. - 10. in Figure 9). At the same time, all nodes covered by the shaded area are intact. The result of this first iteration is that the system does not operate for any of the three paths.

Thus, steps 6. - 10. are repeated for k = 2, which reveals that the system operates for the union {2, 3, 4, 5, 6, 7, 8}. The removal of odd nodes (step 14. in Figure 9) leads to storing the two, yet incomplete minimal path sets

MP_1 = {2, 3, 4, 5, 6, 7}; MP_2 = {2, 3, 4, 5, 6, 8}

These are indicated by the green nodes in Figure 13. After this, the evaluation of subgraph M ends.

The algorithm continues by evaluating subgraph H S for each of the two incomplete minimal path sets. For MP_1, the DFSMP starts from node 7 (as shown by Figure 13) and finds four paths:

$path_1$ = {7, 9, 11}; $path_2$ = {7, 15, 16, 8, 10, (14)}
$path_3$ = {7, 9, (12)}; $path_4$ = {7, 15, 16, 8, 10, 13}

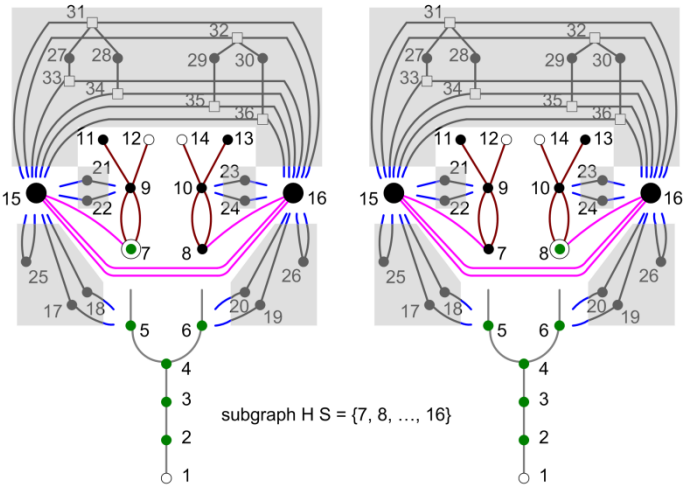

Figure 13. Hydraulic and signal domains subgraph evaluation

Next, the paths are tested by simulating the system model (steps 6. - 10. in Figure 9 for k = 1). In doing so, those nodes are intact that belong to $MP_1 \cup path_j$, $\forall j \in \{1, 2, 3, 4\}$ respectively. Also, all nodes covered by the shaded area are intact. The system does not operate. The next iteration (k = 2) reveals that the system operates for the union $MP_1 \cup path_1 \cup path_2$. The removal of odd nodes leads to storing MP_1 = {2, 3, 4, 5, 6, 7, 9, 11, 15}.

The algorithm proceeds accordingly for the second minimal path set (DFSMP starts from node 8). After the evaluation of subgraph H S is complete, it is stored as MP_2 = {2, 3, 4, 5, 6, 8, 10, 13, 16}. The two minimal path sets developed so far are indicated by the green nodes in Figure 14.

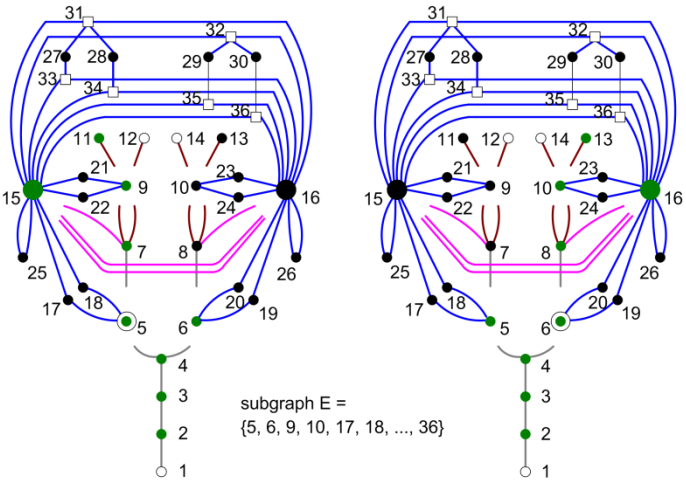

Figure 14. Evaluation of electrical domain subgraph

Next, the algorithm evaluates subgraph E to complete the two minimal path sets. For MP_1, the DFSMP starts from node 5 and find 91 paths. Three of them are:

$path_1$ = {5,17,15,25}; $path_2$ = {5, 18, 15, 21, 9, 22};
$path_3$ = {5, 17, 15, (33), 27, (31), 28, (34), 16, (36), 30, (32), 29, (35)}

The loop (steps 6. - 10. in Figure 9) of testing paths and unions by simulating the system model is repeated up to $k = 3$. Then, the system operates for the union $MP_1 \cup path_1 \cup path_2 \cup path_3$. Odd nodes are removed, and the minimal path set is stored as $MP_1 = \{2, 3, 4, 5, 6, 7, 9, 11, 15, 17, 18, 21, 22, 25, 27, 28, 29, 30\}$.

For $k = 2$, $C(91, 2) = 4095$ unions were established of which a number of $q_k = 634$ remained after removal of duplicates. For $k = 3$, $C(91, 3) = 121485$ unions were established of which a number of $q_k = 1279$ remained. As explained, the model is simulated only for the respective q_k unions.

Next, the second minimal path set is completed by starting a DFSMP from node 6. The loop of testing paths and unions in the simulation is run through up to $k = 3$ with similar computing effort. The result is $MP_2 = \{2, 3, 4, 5, 6, 8, 10, 13, 16, 19, 20, 23, 24, 26, 27, 28, 29, 30\}$. Both complete minimal path sets are indicated by the green nodes in Figure 15.

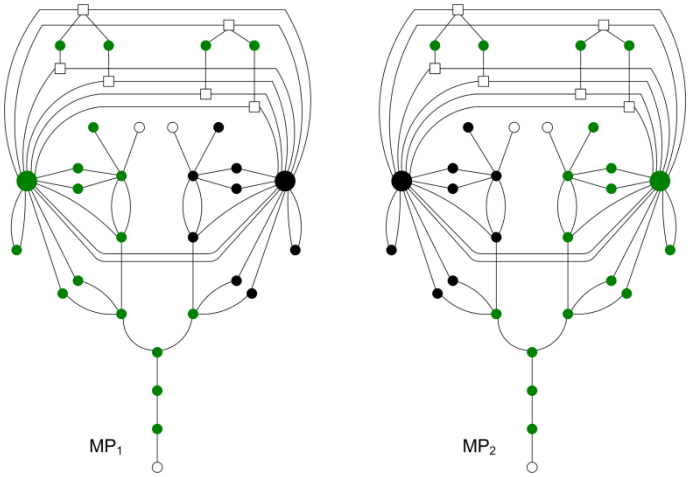

Figure 15. Minimal path set of stabiliser trim system

The total number of system model simulations performed is 4129. Most of them are caused by step 6., and a few are due to step 14. of the minimal path sets detection algorithm depicted by Figure 9. If the model graph was not split up (row 1 in Table 5), this would cause a total of > 615000 simulations, which is unfeasible. Without any evaluation of the model graph, the number would even be $2^{27} = 1.34 \cdot 10^8$ due to the system comprising 27 components (nodes) that include failure behaviour.

4 CONCLUSIONS

This paper described and exemplified a method for minimal path sets detection of a technical system that is represented by a multi-domain object-oriented model. The method is the core of an automatic safety analysis that can be performed throughout the design process of the system, thus keeping the analysis consistent with design iterations. It is meaningful to apply the method if multi-domain object-oriented modelling is used already in systems engineering. The proposed method enhances the scope of application of a system model while permitting all other simulation studies that originally motivated development of the model to be conducted.

The method belongs to the class of state space simulations. A search algorithm from graph theory is adapted to evaluate the object structure of the model, which prevents exponential growth of the state space and thus unfeasibility of the method.

Further detail of the proposed method, estimation of computing effort, as well as modelling and analysis examples are described by Schallert (2014).

It must be beared in mind that all model-based analysis methods capture only those phenomena that are covered in the modelling approach. Then again, the automation ensures that all relevant failure conditions (at least in so far as modelled) are considered, whereas a manually conducted analysis might be errorneous or incomplete.

ACKNOWLEDGEMENT

This research has received funding from the European Union's 7th Framework Programme (FP7/2007-2013) for the CleanSky Joint Technology Initiative under grant agreement CSJU-GAN-SGO-2008-001.

REFERENCES

Airbus Training. *A320 Flight Crew Operating Manual, section 1.27.00 - Flight Controls*. Available at website: http://www.smartcockpit.com/

Boeing B737 NG - *Systems Summary - Flight Controls*. Available at website: http://www.smartcockpit.com/

Bouissou, M. & Elmqvist, H. & Otter, M. & Benveniste, A. 2014. Efficient Monte Carlo simulation of stochastic hybrid systems. *Proceedings of the 10th international Modelica conference,* Lund, Sweden, March 10-12, 2014.

Elmqvist, H. 1978. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Lund Institute of Technology, Lund, Sweden.

Elmqvist, H. & Mattsson, S. E. & Otter, M. 2014. Modelica extensions for Multi-Mode DAE Systems. *Proceedings of the 10th international Modelica conference,* Lund, Sweden, March 10-12, 2014.

Heidtmann K. D. 1989. Smaller Sums of Disjoint Products by Subproduct Inversion. *IEEE Transactions on Reliability* (Vol. 38, No. 3): pp. 305 - 311.

Krumke, S. O. & Noltemeier, H. 2012. *Graphentheoretische Konzepte und Algorithmen*. Vieweg+Teubner Verlag, 3. Auflage. In German.

The Modelica Association 2000. *Modelica – A Unified Object-Oriented Language for Physical Systems Modeling*. Available at website: https://modelica.org/documents/ModelicaTutorial14.pdf/

Persson, U. & Schallert, C. 2001. *The 728 JET flight control system*. Deutscher Luft- und Raumfahrtkongress, Hamburg, Germany, September 2001.

Roberts, B. & Kroese, D. P. 2007. Estimating the number of s-t paths in a graph. *Journal of Graph Algorithms and Applications* (Vol. 11, No. 1): pp. 195 - 214.

Schallert, C. 2014 (to appear). *Integration of Safety and Reliability Analysis Methods with Object-Oriented Modelling*. PhD thesis, Technische Universität Berlin, Germany.