IMBSA Tutorial

Finite Degradation Structures

Liu Yang* and Antoine Rauzy*

Speaker: Liu Yang

*Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
CONTENTS

1. Mathematical Framework
2. Modeling Components
3. Modeling Systems
4. Assessing Models
5. Case Study
6. Conclusion
7. References
Finite Degradation Structures (FDS) are the most general mathematical framework of combinatorial reliability/safety models such as fault trees, reliability block diagrams, etc.

**Combinatorial models**: describe the state of the system as a combination of the states of its components or subsystems.

\[
\text{System.state} = \text{Function(C1.state, C2.state, \ldots, Cn.state)}
\]

States of system and components can be seen as variables taking their values into finite domains. The type of the function depends on the domain of variables.
Finite Degradation Structures

1. Mathematical Framework

More precisely, FDS extend the fault tree analysis (FTA) from Boolean systems into multi-state systems. FDS generalize formally almost all the notions used in FTA, including:

- **Basic events**
- **Top/intermediate events**
- **Logic gates**
- **Probability of top event**
- **Minimal path sets**
- **Minimal cutsets**
- **State variables**
- **Flow variables**
- **Multi-valued operators**

### Diagram

- **FTA**:
  - Basic events
  - Top/intermediate events
  - Logic gates
  - Probability of top event
  - Minimal path sets
  - Minimal cutsets
  - Extended

- **FDS**:
  - State variables
  - Flow variables
  - Multi-valued operators
  - Probability of each state
  - Maximal scenarios
  - Minimal scenarios

### Boolean
- **Boolean systems**
- **Formal generalization**

### Multi-state systems
- **Multi-state systems**
- **Boolean systems**
Finite Degradation Structures

1. Mathematical Framework

**FDS** are mathematically *meet-semi-lattices* equipped with *probability measure*.

A meet-semi-lattice is a triple \( \Theta, \sqsubseteq, \bot \) s.t.

- \( \Theta \) is a **finite set**
- \( \sqsubseteq \) is a **partial order** defined over \( \Theta \)
- \( \bot \) is the **least element** in \( \Theta \) with respect to \( \sqsubseteq \)

**Example.** Boolean component, can be either working (W) or failed (F)

Component A modeled by

- **Basic event**: Boolean variable A, valued in the set \( \{0,1\} \)
- **State variable**: valued in the FDS named as **WF**
  - **WF** is the triple: \( \{W,F\} \) with \( W \sqsubseteq F \), \( W \)
  - "W is less degraded than F"

FTA

Component A

 Degradation ordering

Hasse diagram

 1. State space
 2. Degradation order among states
 3. Least degraded state

W

F
Finite Degradation Structures

2. Modeling Components

Example. Three-state component

Component A
- Working (W)
- Degraded (D)
- Failed (F)

Component B
- Working (W)
- Failed to open (Fo)
- Failed to close (Fc)

FTA

Component A:
- A=W
- A=D
- A=F

Component B:
- B=W
- B=Fo
- B=Fc

There is no difference between component A and B, except the name of states.

FDS

A valued in the FDS named WDF

B valued in the FDS named WFoFc

There is a structural difference between A and B.

This difference impacts on the performance of the system made by A and B.
Finite Degradation Structures

2. Modeling Components

Example. Three-state component

Component A
- Working (W)
- Degraded (D)
- Failed (F)

Component B
- Working (W)
- Failed to open (Fo)
- Failed to close (Fc)

FTA

Component A
- A=W
- A=D
- A=F

Component B
- B=W
- B=Fo
- B=Fc

FDS

Component A
- A
- D
- W

Component B
- B

There is no difference between component A and B, except the name of states.

Similar situations
Similar structures
Similar performances?

Finite Degradation Structures
Finite Degradation Structures

3. Modeling Systems

Example. Series composition of three-state components

Component A1
- Working (W)
- Degraded (D)
- Failed (F)

Component A2
- Working (W)
- Degraded (D)
- Failed (F)

Series operator

<table>
<thead>
<tr>
<th>A1</th>
<th>W</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Valuation table

(F(Sys=W) := WW)
(F(Sys=D) := WD + DW + DD)
(F(Sys=F) := WF + DF + FF + FD + FW)
3. Modeling Systems

Example. Series composition of three-state components

Finite Degradation Structures

Component A1
Working (W) Degraded (D) Failed (F)

Component A2
Working (W) Degraded (D) Failed (F)

Series operator

<table>
<thead>
<tr>
<th>A1</th>
<th>W</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Valuation table

Monoidal product on FDSs

Valuations defined by the operation $\lor$
Most highlighted contribution of FDS --- Critical scenarios for multi-state systems

- **Minimal scenarios**: least degraded state(s) that the system enters into an undesired state ~ minimal cutsets
- **Maximal scenarios**: most degraded state(s) that the system still remains in an optimal state ~ minimal path sets

Valuations defined by the operation $\vee$

Extended from

- Minimal scenarios
- Maximal scenarios
- Least upper bounds
- Greatest lower bounds
Finite Degradation Structures

4. Assessing Models

Extended decision diagram for assessing multi-state models built on FDSs

- **Terminal nodes**: valuation results from different paths
- **Internal nodes**: labeled with variables

If $A = W$, then...

Otherwise, ...

If $A = W$, then...

Else-edge
Finite Degradation Structures

5. Case study

Example. HIPPS (High Integrity Pressure Protection Systems)

Step 1. Components: S1, S2, S3, LS1, LS2, V1, V2 whose valuation domains are customized FDSs

Step 2. Operators

<table>
<thead>
<tr>
<th>Series</th>
<th>W</th>
<th>Fs</th>
<th>Fdd</th>
<th>Fdu</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>Fs</td>
<td>Fdd</td>
<td>Fdu</td>
</tr>
<tr>
<td>Fs</td>
<td>Fs</td>
<td>Fs</td>
<td>Fdd</td>
<td>Fdd</td>
</tr>
<tr>
<td>Fdd</td>
<td>Fdd</td>
<td>Fs</td>
<td>Fdd</td>
<td>Fdu</td>
</tr>
<tr>
<td>Fdu</td>
<td>Fdu</td>
<td>Fs</td>
<td>Fdd</td>
<td>Fdu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parallel</th>
<th>W</th>
<th>Fs</th>
<th>Fdd</th>
<th>Fdu</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>Fs</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>Fs</td>
<td>Fs</td>
<td>Fs</td>
<td>Fs</td>
<td>Fs</td>
</tr>
<tr>
<td>Fdd</td>
<td>Fdd</td>
<td>Fdd</td>
<td>Fdd</td>
<td>Fdd</td>
</tr>
<tr>
<td>Fdu</td>
<td>Fdu</td>
<td>W</td>
<td>Fdd</td>
<td>Fdu</td>
</tr>
</tbody>
</table>

SensorGroup := Parallel(S2, S3)
ValveGroup := Parallel(V1, V2)
System := Parallel(SafetyChannel1, SafetyChannel2)

Step 3. Formulate the model of the system

SafetyChannel1 := Series(Series(S1, LS1), V1)
SafetyChannel2 := Series(Series(SensorGroup, LS2), ValveGroup)
Finite Degradation Structures

5. Case study

Example. HIPPS (High Integrity Pressure Protection Systems)

1. Probabilistic results

2. Critical scenarios
(made up of state combinations of the 7 components)

Maximal scenarios that the system is still in the working state **W**

\[
\binom{16}{4} = \frac{16!}{4!(16-4)!} = 1,820
\]

Minimal scenarios that the system is failed into **Fdu** state

\[
\binom{9}{4} = \frac{9!}{4!(9-4)!} = 126
\]

# of Critical/mapped/total scenarios
6. Conclusion

- FDS **unify** Boolean and multi-state combinatorial models into one framework, from both theoretical and practical point of view.

- In particular, FDS make it possible to generalize (and to revisit) the central notion of **minimal cutsets**.

- FDS also provide **interfaces** with systems architectural decompositions (to synchronize with the system design), ...
7. References


Thanks for your listening.

Questions?