Diagnosing faults in systems using a fault tree based model

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- This is a conference paper

Metadata Record: [https://dspace.lboro.ac.uk/2134/3626](https://dspace.lboro.ac.uk/2134/3626)

Publisher: © E.E. Hurdle, L.M. Bartlett and John D. Andrews

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Diagnosing Faults in Systems Using a Fault Tree Based Method

E. E. Hurdle, L. M. Bartlett, J. D. Andrews
Presentation Contents

1. Background and Introduction
2. Development of the fault tree based method
   - Example Water Tank Level Control System
3. Discussion of Results
4. Conclusions
1. Background

• Fault Diagnosis:
  – The process of identifying the cause of a malfunction by observing its effect on a system

• The increased complexity of modern day systems has made the diagnosis of faults a more difficult task to perform
  – Quick rectification is required in order to:
    • reduce the time taken for a system to resume normal service
    • minimise the effects of failure on the system
1. Background

• Fault diagnosis can be performed in two different ways:
  – Testing for faults at specific points in time
  – Monitoring the system continuously to detect for faults as and when they occur

• Testing for faults at specific points in time:
  – Sequential testing
  – Heuristic search algorithms
  – FMEA

• Monitoring the system continuously to detect for faults as and when they occur:
  – Statecharts and fault trees
  – Genetic algorithms
1. Introduction

• Method for diagnosing single or multiple faults in systems using Fault Tree Analysis (FTA)
  – Used to explain deviations in sensor outputs
  – System failure described in terms of component states
  – Needs a model of expected system behaviour
    • Sensor observations
    • System parameters
1. Generalised Method

- Obtain readings from sensors and calculate parameters
- Develop non-coherent fault trees for each sensor reading
  - AND, OR and NOT logic
- Compare monitored parameters for different sensor types where possible. If they do not agree they are unreliable
  - Direct comparison of readings or use of readings to calculate parameters
- Obtain a model of system behaviour in order to identify how the system should be behaving
1. Generalised Method

- **Perform analysis to obtain potential causes of failure**
- **Remove working states from the potential causes**
  - Only failed component states need to be considered
- **Check potential causes against sensors reading true to the operating mode**
  - Remove causes of failure that contradict these sensor readings
- **If there is more than one possible potential cause of failure, use importance measures to obtain the most likely outcome**
2. Water Tank Level Control System
2. Water Tank Basic Events

- Potential Component Failures:

<table>
<thead>
<tr>
<th>Code</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_iB$ $(1 \leq i \leq 6)$</td>
<td>Pipe $P_i$ is Blocked</td>
</tr>
<tr>
<td>$P_iF$ $(1 \leq i \leq 6)$</td>
<td>Pipe $P_i$ is Fractured</td>
</tr>
<tr>
<td>$V_iFC$ $(1 \leq i \leq 3)$</td>
<td>Valve $V_i$ Fails Closed</td>
</tr>
<tr>
<td>$V_iFO$ $(1 \leq i \leq 3)$</td>
<td>Valve $V_i$ Fails Open</td>
</tr>
<tr>
<td>$S_iFF$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails Full</td>
</tr>
<tr>
<td>$S_iFVH$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails Very High</td>
</tr>
<tr>
<td>$S_iFH$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails High</td>
</tr>
<tr>
<td>$S_iFN$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails Normal</td>
</tr>
<tr>
<td>$S_iFL$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails Low</td>
</tr>
<tr>
<td>$S_iFE$ $(1 \leq i \leq 2)$</td>
<td>Sensor $S_i$ Fails Empty</td>
</tr>
<tr>
<td>$C_iFH$ $(1 \leq i \leq 2)$</td>
<td>Controller $C_i$ Fails High</td>
</tr>
<tr>
<td>$C_iFL$ $(1 \leq i \leq 2)$</td>
<td>Controller $C_i$ Fails Low</td>
</tr>
<tr>
<td>TR</td>
<td>Water Tank Ruptured</td>
</tr>
<tr>
<td>TL</td>
<td>Tank Leaks</td>
</tr>
<tr>
<td>NWMS</td>
<td>No Water from Main Supply</td>
</tr>
</tbody>
</table>

- Potential Phases:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Flow Phase Through Valve V2</td>
</tr>
<tr>
<td>DORMANT</td>
<td>No Flow Phase Through Valve V2</td>
</tr>
</tbody>
</table>
2. Operating Assumptions

- Flow into the tank through V1 has the capability to be greater than flow out of the tank through V2
- P5 and P6 have cross-sectional areas that are twice the size of the other pipes in the system
- If a rupture occurs in the tank then flow out through the rupture is greater than flow into the tank at V1
- Flow into the tank is greater than flow out of a leak in the system
- Initial conditions have the water level as normal
2. System Information

- Readings from VF1, VF2, VF3 and SP1
  - In addition can be used to generate system states:
    - The level of water in the tank
    - The rate of change

- Sensor readings from the level sensors S1 and S2 are also used in the system analysis
  - Read the level of water in the tank
  - Read the rate of change
2. Model Flow Diagram

- Set Operating Mode
  - Generate Fault
    - Simulation Model
      - Sensor Readings
        - Expected Values
          - VF1 = NF
          - VF2 = NF
          - VF3 = NF
          - SP1 = NW
        - Compare with Expected Values
          - VF1 = F
          - VF2 = NF
          - VF3 = F
          - SP1 = NW

- DORMANT
  - V1FO,C2FH
  - Determine Potential Faults
    - Importance Measures
      - Obtain Most Likely Cause

## 2. Expected System Behaviour

<table>
<thead>
<tr>
<th>Mode</th>
<th>Height</th>
<th>Scenario</th>
<th>VF1</th>
<th>VF2</th>
<th>VF3</th>
<th>SP1</th>
<th>Rate Of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Empty</td>
<td>4</td>
<td>Flow</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &gt; 0 )</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Low</td>
<td>4</td>
<td>Flow</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &gt; 0 )</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Normal</td>
<td>12</td>
<td>No Flow</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>High</td>
<td>12</td>
<td>No Flow</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Very High</td>
<td>10</td>
<td>No Flow</td>
<td>Flow</td>
<td>Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Full</td>
<td>10</td>
<td>No Flow</td>
<td>Flow</td>
<td>Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>Empty</td>
<td>8</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &gt; 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>Low</td>
<td>8</td>
<td>Flow</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h &gt; 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>Normal</td>
<td>16</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h = 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>High</td>
<td>16</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Flow</td>
<td>No Water</td>
<td>( h = 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>Very High</td>
<td>14</td>
<td>No Flow</td>
<td>No Flow</td>
<td>Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
<tr>
<td>DORMANT</td>
<td>Full</td>
<td>14</td>
<td>No Flow</td>
<td>No Flow</td>
<td>Flow</td>
<td>No Water</td>
<td>( h &lt; 0 )</td>
</tr>
</tbody>
</table>
2. Fault Tree Construction

- Non-coherent fault trees were used in the analysis
  - Constructed using AND, OR and NOT logic
  - System Observation points and level sensor readings are described in terms of the component failures and working states
2. Fault Tree Construction

- Non-coherent fault trees were used in the analysis
2. Fault Tree Construction

- Non-coherent fault trees were used in the analysis
2. Fault Tree Construction

• Non-coherent fault trees were used in the analysis
2. Fault Tree Construction

- Non-coherent fault trees were used in the analysis
2. Fault Tree Construction

• Non-coherent fault trees were used in the analysis
  – Constructed using AND, OR and NOT logic
  – System Observation points and level sensor readings are described in terms of the component failures and working states

• Potential causes of failure for No Flow Through Valve V2:
  1) V2FC   9) S1FN
  2) P3B   10) S1FH
  3) P3F   11) S1FVH
  4) P4B   12) S1FF
  5) TR   13) P1B
  6) C1FH   14) P1F
  7) NWMS   15) P2B
  8) V1FC
2. Top Event Structure

- Look for the causes of the deviation (use a coherent top event structure and non-coherent fault trees), then check these individually against those from the sensor readings which are true to the operating mode

- Check potential causes of failure in this case against flow through V2 and water in the overspill tray
3. Scenario Top Event Structure

- DORMANT Operating mode with Height = Normal
- Induced Failure: V1FO.C2FH

<table>
<thead>
<tr>
<th>Scenario</th>
<th>VF1</th>
<th>VF2</th>
<th>VF3</th>
<th>SP1</th>
<th>HEIGHT</th>
<th>RATE</th>
<th>S1/S2 HEIGHT</th>
<th>S1/S2 RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPECTED</td>
<td>NF</td>
<td>NF</td>
<td>NF</td>
<td>NW</td>
<td>$h = \text{Normal}$</td>
<td>$\dot{h} = 0$</td>
<td>$h = \text{Normal}$</td>
<td>$\dot{h} = 0$</td>
</tr>
<tr>
<td>ACTUAL</td>
<td>F</td>
<td>NF</td>
<td>F</td>
<td>NW</td>
<td>$h = \text{Normal}$</td>
<td>$\dot{h} &lt; 0$</td>
<td>$h = \text{Normal}$</td>
<td>$\dot{h} &lt; 0$</td>
</tr>
</tbody>
</table>

- Potential Causes:

<table>
<thead>
<tr>
<th>Number</th>
<th>Potential Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>V3FO.-V3FC.-P5B.-P5F.-P6B.-V1FC.-P1B.-P1F.-P2B.-S1FE.-S1FL.-S1FN.-S1FH.-S1FVH.-S1FF.-S2FE.-S2FL.-S2FN.-S2FH.-S2FVH.-S2FF.-TR</td>
</tr>
<tr>
<td>2)</td>
<td>C2FH.-V3FC.-C2FL.-P5B.-P5F.-P6B.-V1FC.-P1B.-P1F.-P2B.-S1FE.-S1FL.-S1FN.-S1FH.-S1FVH.-S1FF.-S2FE.-S2FL.-S2FN.-S2FH.-S2FVH.-S2FF.-TR</td>
</tr>
</tbody>
</table>

- Method has not obtained the exact cause of failure in this case
- The two induced failures can be the cause of both deviated sensor readings
- Timing will need to be taken into consideration
3. Dynamics

- **DORMANT Operating mode with height = Normal**
- **Induced Failure: V1FO.C2FH**

  - Once the level reaches ‘Low’ flow in at V1 is expected, so a failure will not be indicated from $t_2$ onwards for the induced failure V1FO.C2FH
  - If only C2FH is induced then the level would drop at a faster rate, until the ‘Low’ level is reached (at time $t_1$ in this case), at which point V1 is opened
  - The time taken for the level of water in the tank to drop may need to be considered in the analysis

![Diagram showing the level-time relationship with V1FO.C2FH and C2FH labels indicating specific points and times.]
4. Conclusions

- The method obtained in general is good at diagnosing faults in systems.
- The checking mechanism ensures that the potential causes of failure do not conflict with the sensors reading true to the operating mode.
- Reliable and unreliable sensor readings can be identified by comparing the level and rate of change of the height of water in the tank obtained from each different set of sensor readings.
- Further research is required to try and overcome problems such as the tool not identifying all failures induced:
  - The time taken for the level of water in the tank to drop may need to be considered in the analysis.
Questions?

Emma Hurdle
E.E.Hurdle@lboro.ac.uk
Dept. of AAE
Loughborough University
01509 227243