UNIVERSITY OF Hull

4th International Symposium on Model Based Safety Assessment

Reliability Analysis of Dynamic Systems by Translating Temporal fault Trees into Bayesian Networks

Sohag Kabir, Martin Walker, and Yiannis Papadopoulos

Department of Computer Science, University of Hull, UK

email: s.kabir@2012.hull.ac.uk

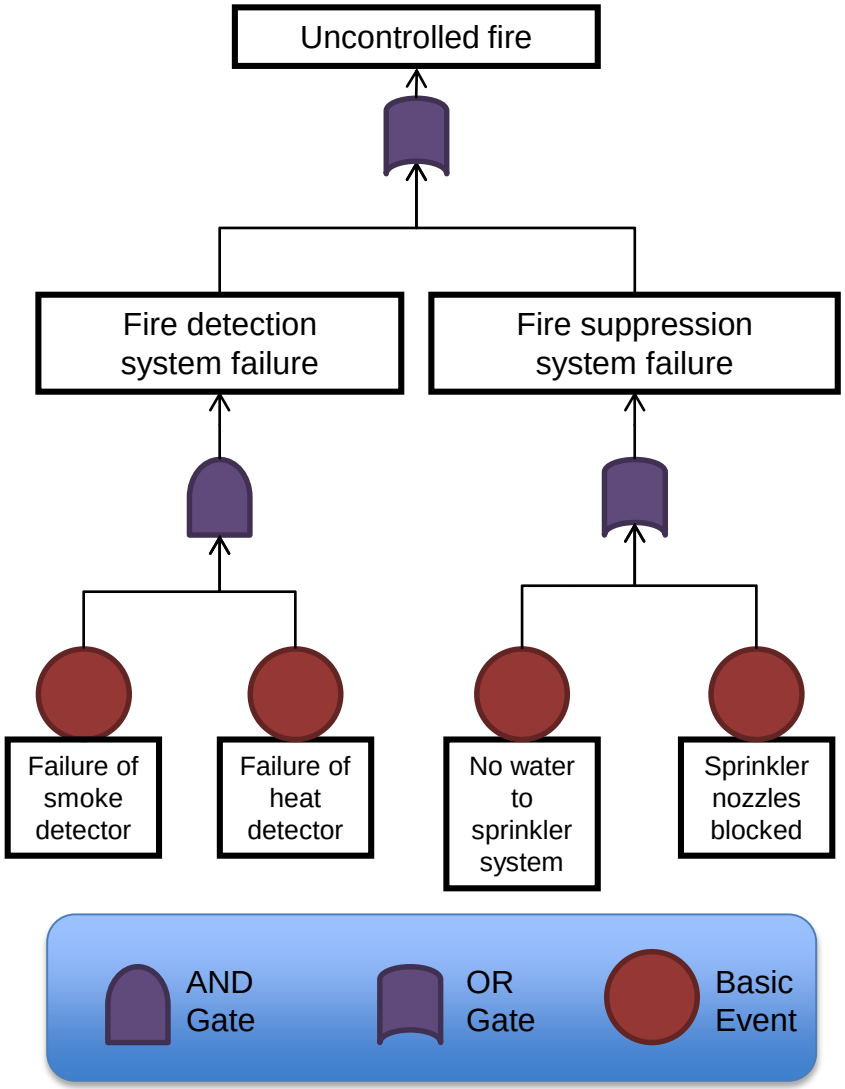
October 29, 2014; Munich, Germany

Safety And Reliability Analysis

1. Safety Critical Systems are increasingly common in many areas
   - e.g. transport, energy, medicine, industrial processes
2. Architecture of systems becoming more complex and their behaviour also becoming dynamic
3. Systems on which we depend the most, those with the worst consequences should they fail
4. Important to make systems as reliable and as safe as possible

Fault Tree Analysis

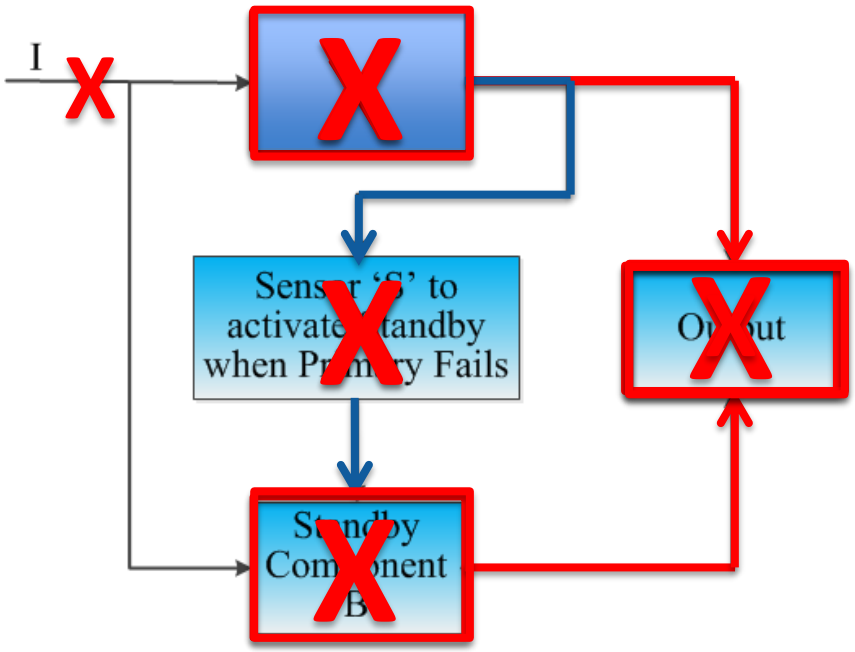
- Widely used method for evaluating system reliability
- Shows logical connections between faults and their causes
- Uses Boolean Logic
- Possible to understand how combinations of component faults can lead to system failure



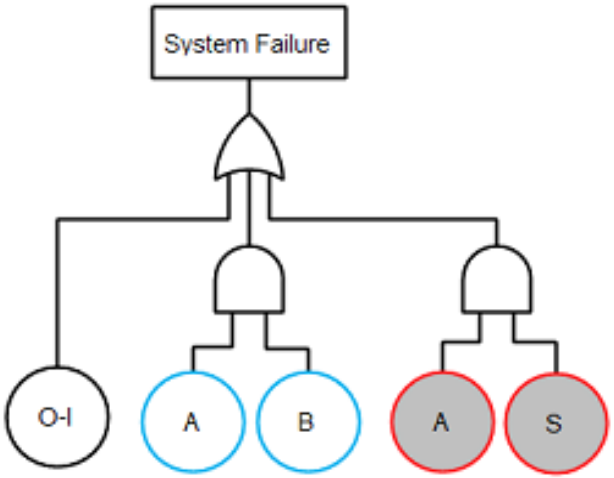
Fault Tree Analysis Cont…

- For many systems, assessing the effects of combinations of failure events is not enough to capture the system failure behaviour
- In addition, understanding the order in which they fail is also required for a more accurate failure model
- FTA has limitations in
  - Expressing time or sequence-dependent behaviour

Limitation of FTA: Example



- FTA results:



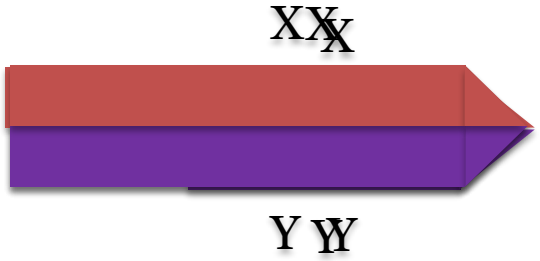
1. There is no input at I.
2. Both A and B fail.
3. Both A and S fail.

Pandora Temporal Fault Trees(TFTs)

- An extension of FTs with temporal gates and compatible with model-based design and analysis techniques
- Capable of capturing sequence-dependent dynamic behaviour
  - Generate Minimal Cut Sequences (MCSQs)
- Provide temporal laws to allow qualitative analysis
  - Help minimising complex temporal expressions

Events and Time in Pandora TFTs

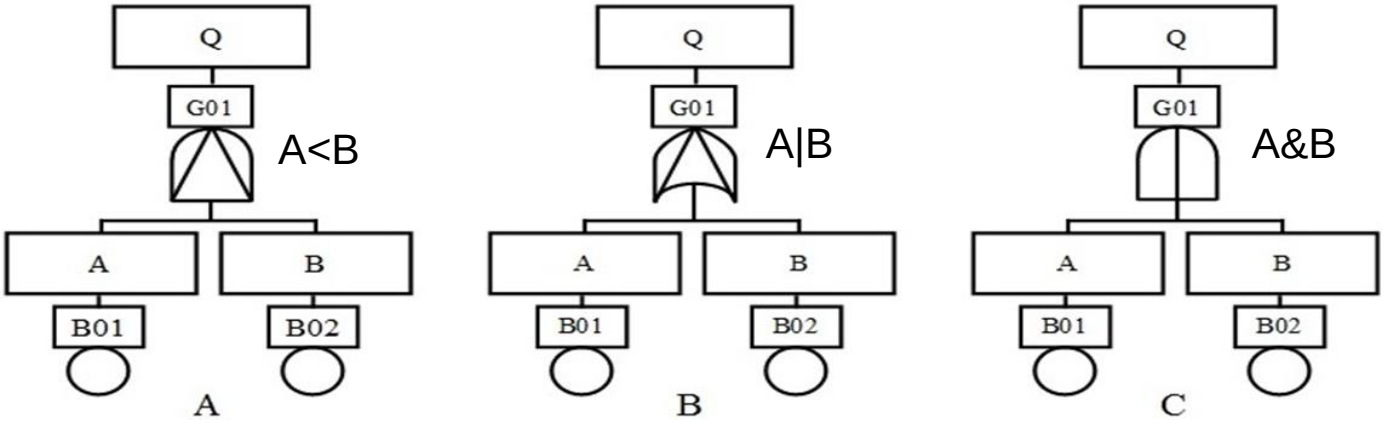
- Events are persistent and occur instantly
- Uses relative time, not exact time, and time can be discrete or continuous, interval or point based
  - Must be linear, no branching
- Three Possible relations between two events X and Y
  - **Before** : X occurs before Y
  - **After** : X occurs after Y
  - **At the same time** : Both X and Y occur simultaneously



Time in Pandora TFTs

- Just need to know when an event occurs relative to other events
  - E.g. which occurs first, which occurs second, and so on
- Uses sequence value as an abstraction of relative time
- If an event does not occur then it is given a sequence value 0
- If an event occurs then it is given a sequence value greater than 0
  - All represents logical true
  - sequence value 1 means the event occurred first, 2 means it occurred second and so on

Pandora Temporal Gates



- PAND: true only if all input events occur and in sequence from left to right
- POR: one input event has priority and must occur first but does not require all other input events to occur as well
- SAND: true if input events occur approximately simultaneously

Temporal Truth Table

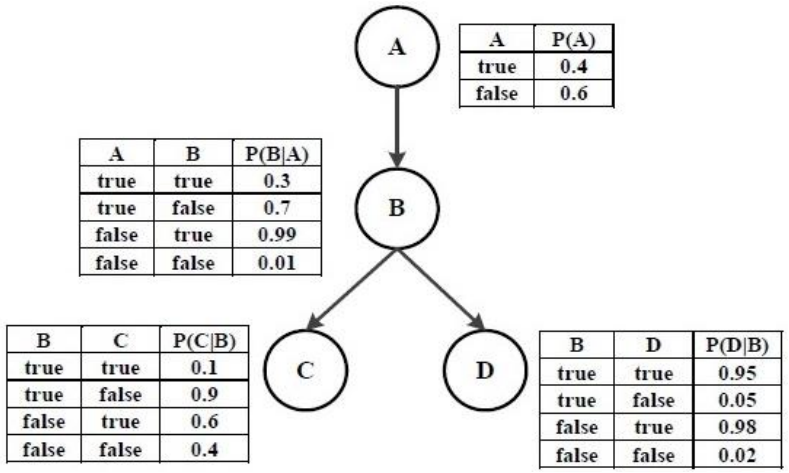
X	Y	X OR Y	X AND Y	X POR Y	X PAND Y	X SAND Y
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	2	2	0	0	0	0
1	0	1	0	1	0	0
1	1	1	1	0	0	1
1	2	1	2	1	2	0
2	0	2	0	2	0	0
2	1	1	2	0	0	0
2	2	2	2	0	0	2

Quantification Pandora TFTs

- Usually performed based on exponential distribution of components' failure rates
- Events are considered statistically independent
- Simulation based approach is also used for quantifying Pandora TFTs
  - Monte Carlo Simulation is used
  - Time Consuming

Bayesian Networks

- Provide probabilistic method of reasoning under uncertainty
- Directed acyclic graph with nodes and arcs
  - Nodes represent random variables
  - Arcs represents dependencies among the nodes
- Consists of a set of prior and conditional probability tables (CPTs)
- Can work with different distribution of the behaviour of the nodes



A	P(A)
true	0.4
false	0.6

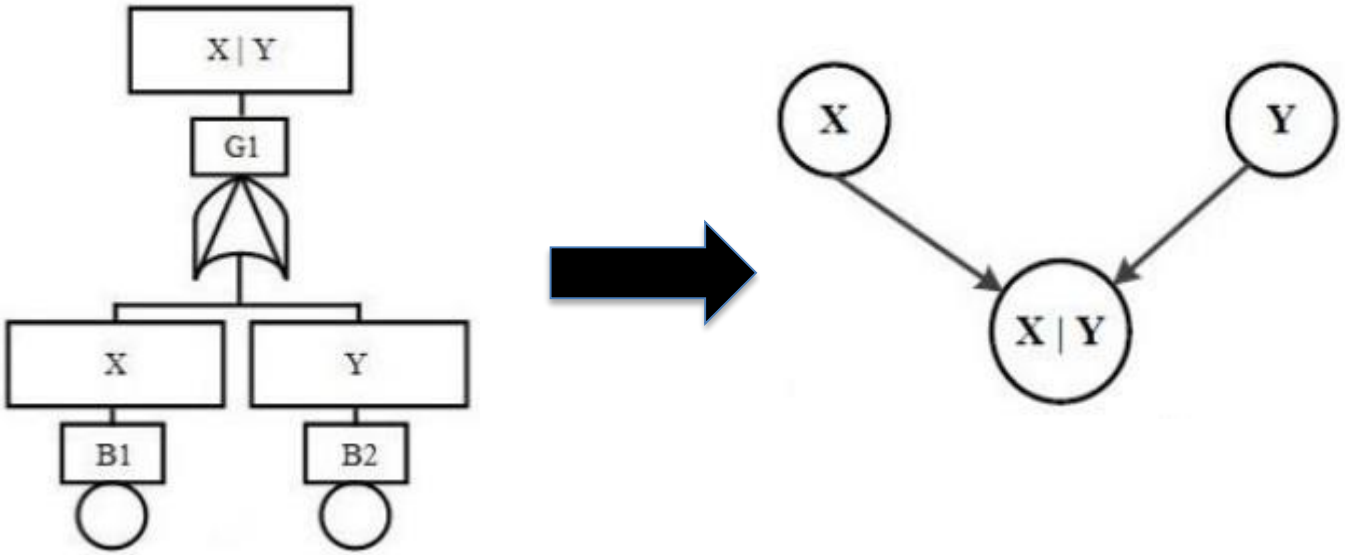
A	B	P(B|A)
true	true	0.3
true	false	0.7
false	true	0.99
false	false	0.01

B	C	P(C|B)
true	true	0.1
true	false	0.9
false	true	0.6
false	false	0.4

B	D	P(D|B)
true	true	0.95
true	false	0.05
false	true	0.98
false	false	0.02

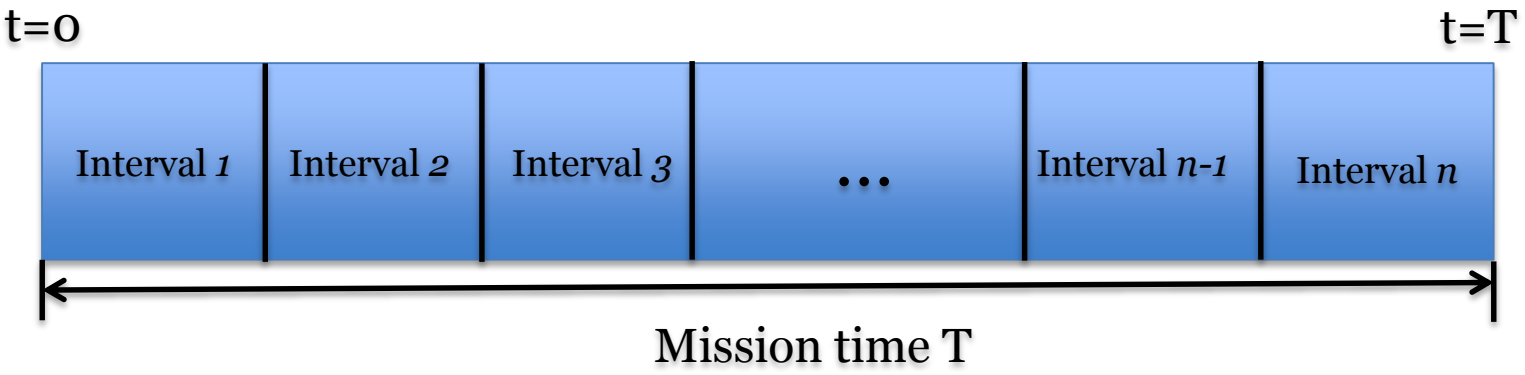
Proposed Methodology

- Step1: Translate Pandora TFT into equivalent discrete-time BN
  - Each root node of BN represents a basic event of TFT
  - Each intermediate node represents a logic gate of TFT



Proposed Methodology

- Step2: Divide the mission *T* into *n* equal intervals



n must be at least equal to the number of input events of a temporal gate which has highest number of input events in the whole TFT

Why restriction on the minimum value of *n*

If $n = 2$

$$(X = 1) \wedge (Y = 2) \wedge (Z = 3) = 3$$

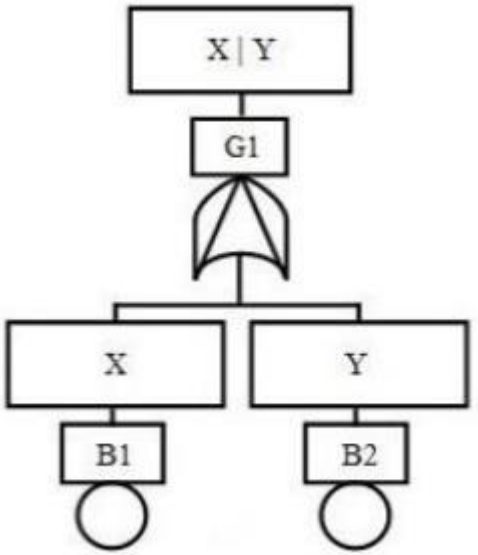
X	Y	Z	X < Y< Z
0	0	0	0
0	0	1	0
0	0	2	0
0	1	0	0
0	1	1	0
0	1	2	0
0	2	0	0
0	2	1	0
0	2	2	0

X	Y	Z	X < Y< Z
1	0	0	0
1	0	1	0
1	0	2	0
1	1	0	0
1	1	1	0
1	1	2	0
1	2	0	0
1	2	1	0
1	2	2	0

X	Y	Z	X < Y< Z
2	0	0	0
2	0	1	0
2	0	2	0
2	1	0	0
2	1	1	0
2	1	2	0
2	2	0	0
2	2	1	0
2	2	2	0

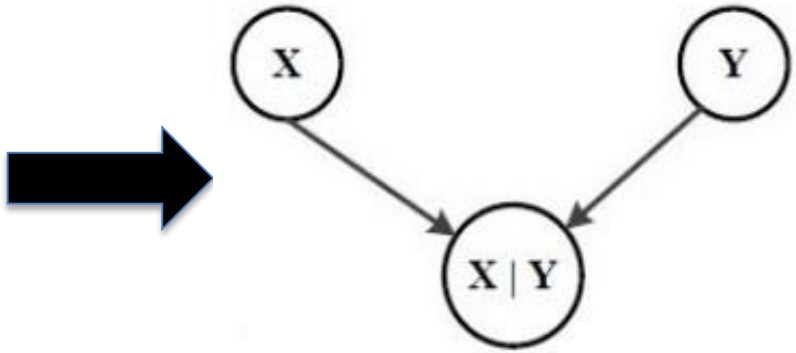
Proposed Methodology

- Step3: Define prior probability table for each of the root nodes based on the failure rates of the basic event represented by the nodes



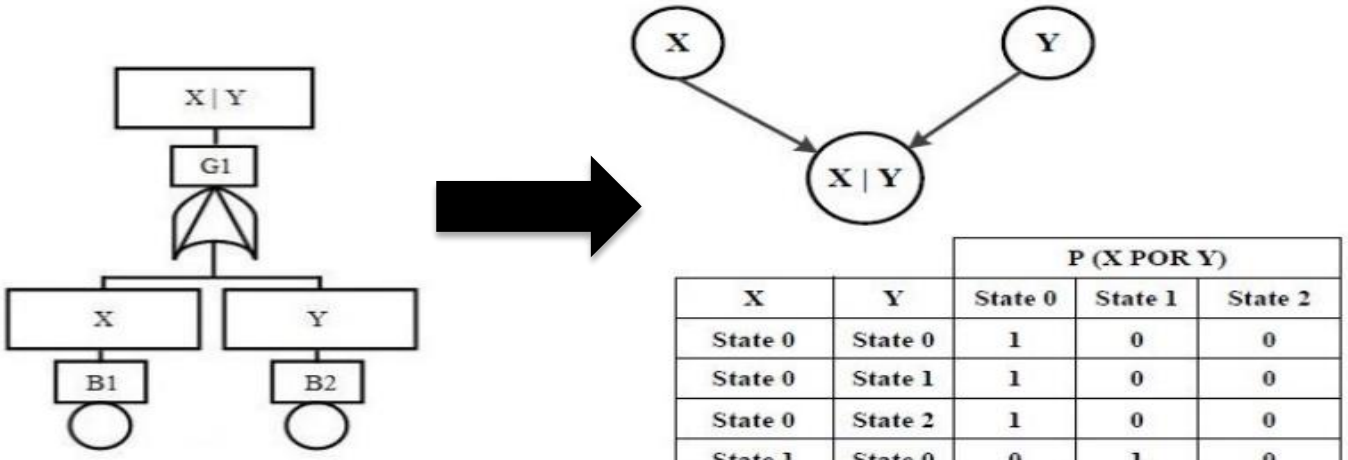
X	State 0	State 1	State 2
P (X)	P0	P1	P2

Y	State 0	State 1	State 2
P (Y)	P0	P1	P2



Proposed Methodology

- Step 4: Define conditional probability table of all the intermediate nodes based on the behaviour of the gates they are representing
  - As the outcomes of the gates are deterministic so each entry in the CPT is either '0' or '1'

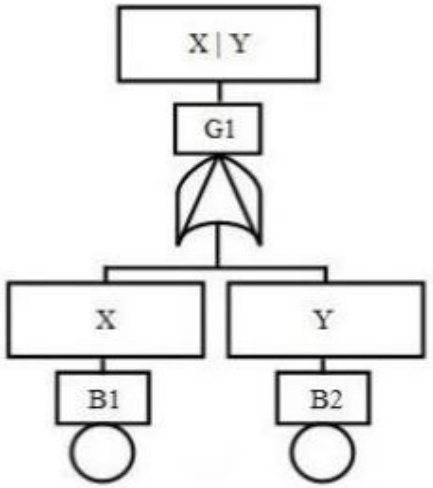


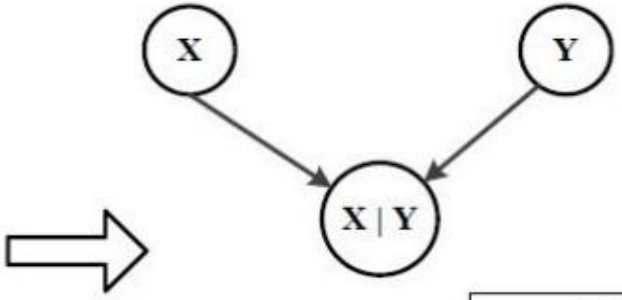
		P (X POR Y)		
X	Y	State 0	State 1	State 2
State 0	State 0	1	0	0
State 0	State 1	1	0	0
State 0	State 2	1	0	0
State 1	State 0	0	1	0
State 1	State 1	1	0	0
State 1	State 2	0	1	0
State 2	State 0	0	0	1
State 2	State 1	1	0	0
State 2	State 2	1	0	0

Two input POR to equivalent BN Example

X	State 0	State 1	State 2
P (X)	P0	P1	P2

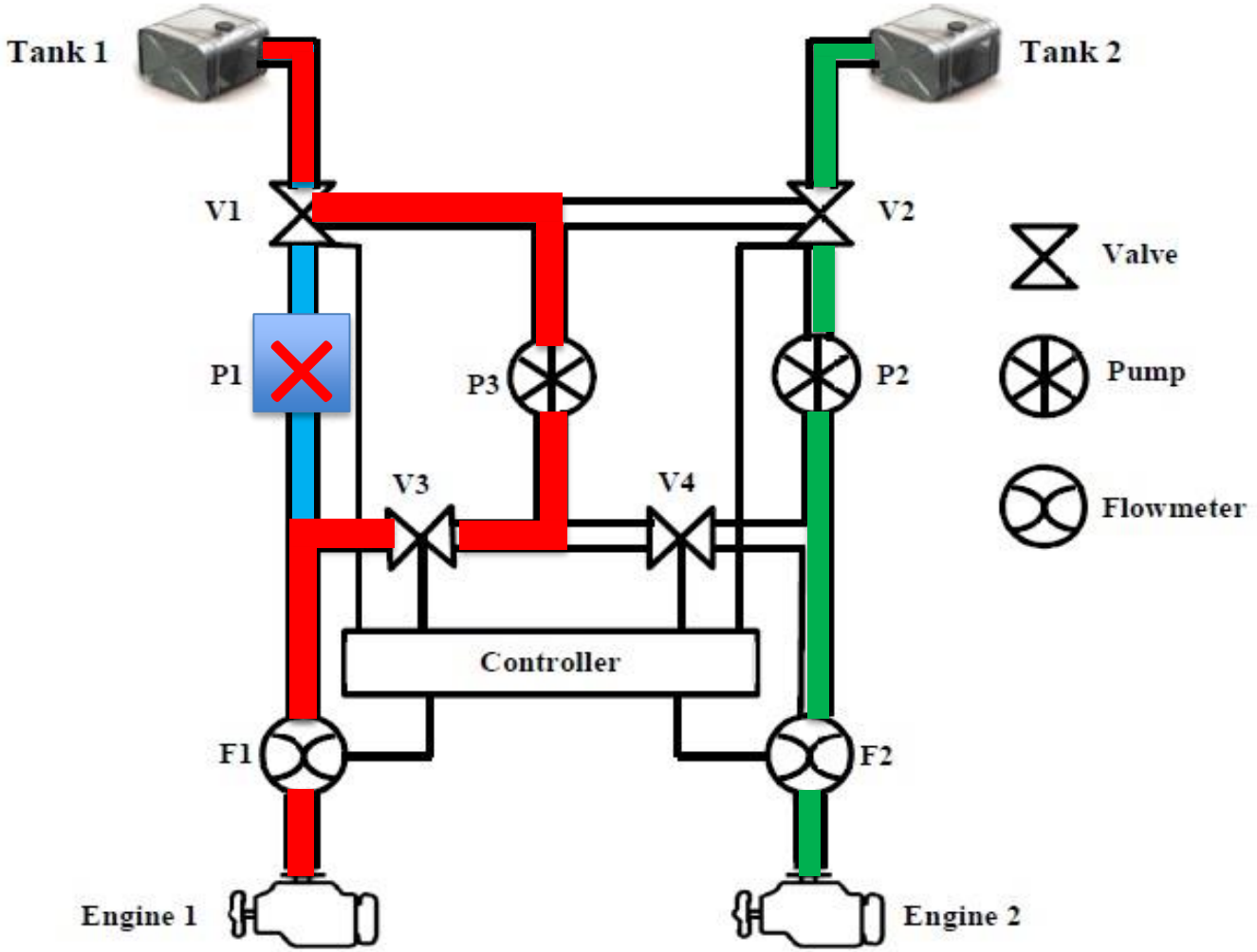
Y	State 0	State 1	State 2
P (Y)	P0	P1	P2





		P (X POR Y)		
X	Y	State 0	State 1	State 2
State 0	State 0	1	0	0
State 0	State 1	1	0	0
State 0	State 2	1	0	0
State 1	State 0	0	1	0
State 1	State 1	1	0	0
State 1	State 2	0	1	0
State 2	State 0	0	0	1
State 2	State 1	1	0	0
State 2	State 2	1	0	0

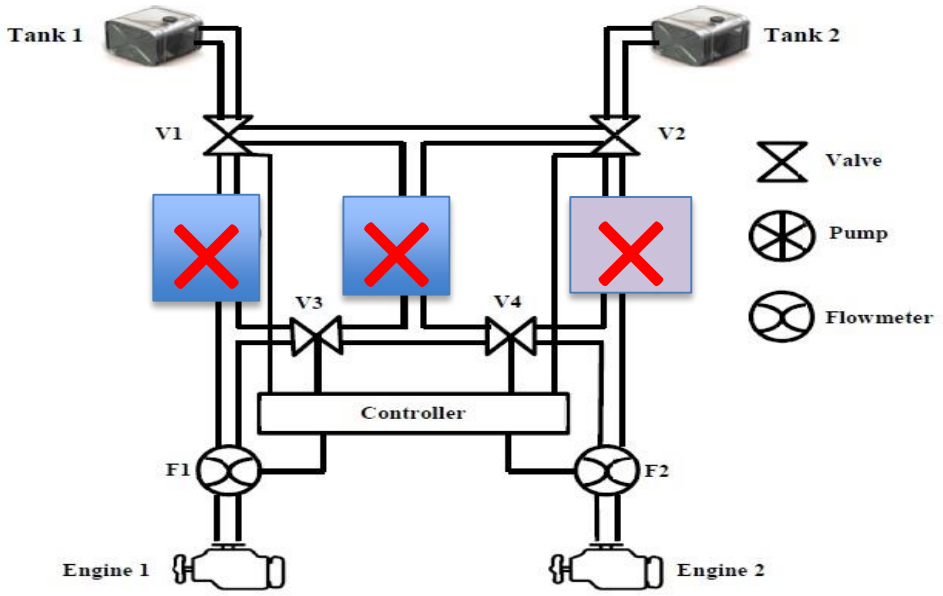
Case Study: fault tolerant fuel distribution system



Failure Behaviour Analysis

MCSQ for example system

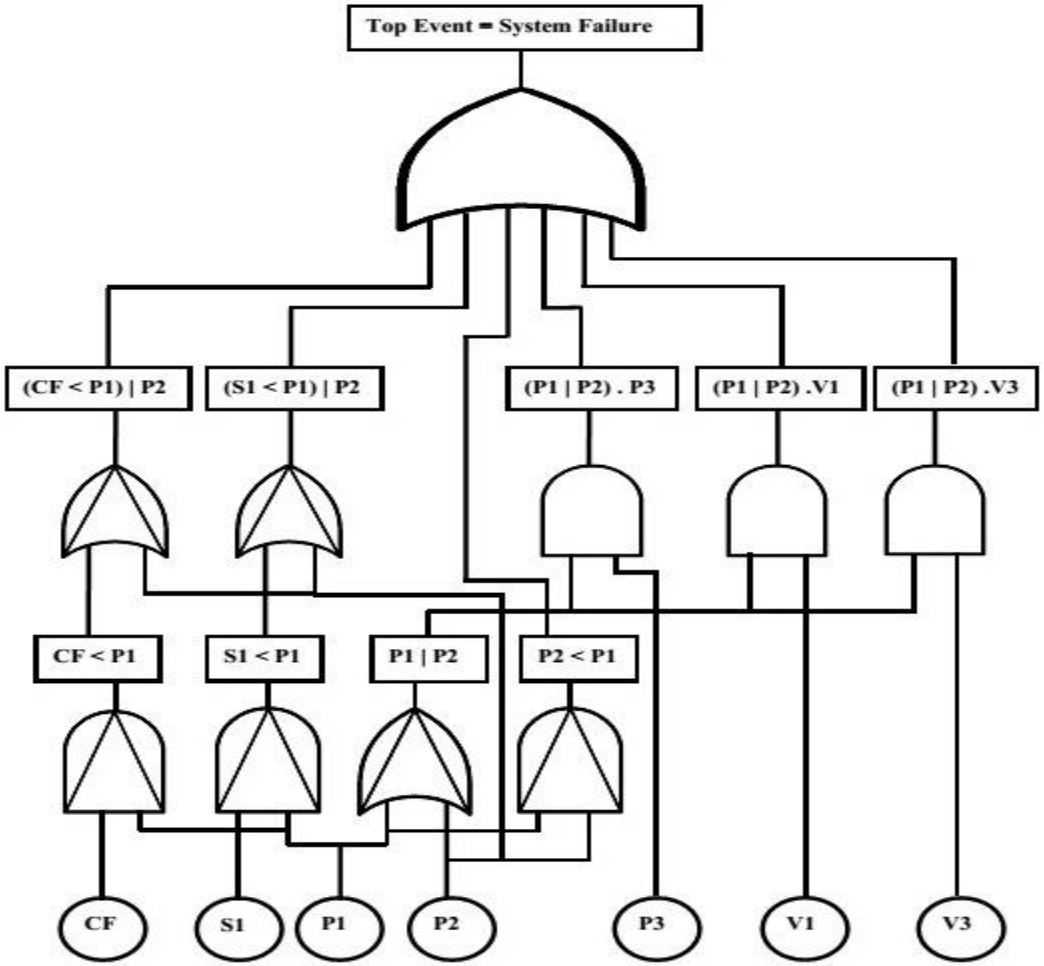
O-Engine 1
(P1|P2).P3
(P1|P2).V1
(P1|P2).V3
(S1<P1)|P2
(CF<P1)|P2
P2<P1



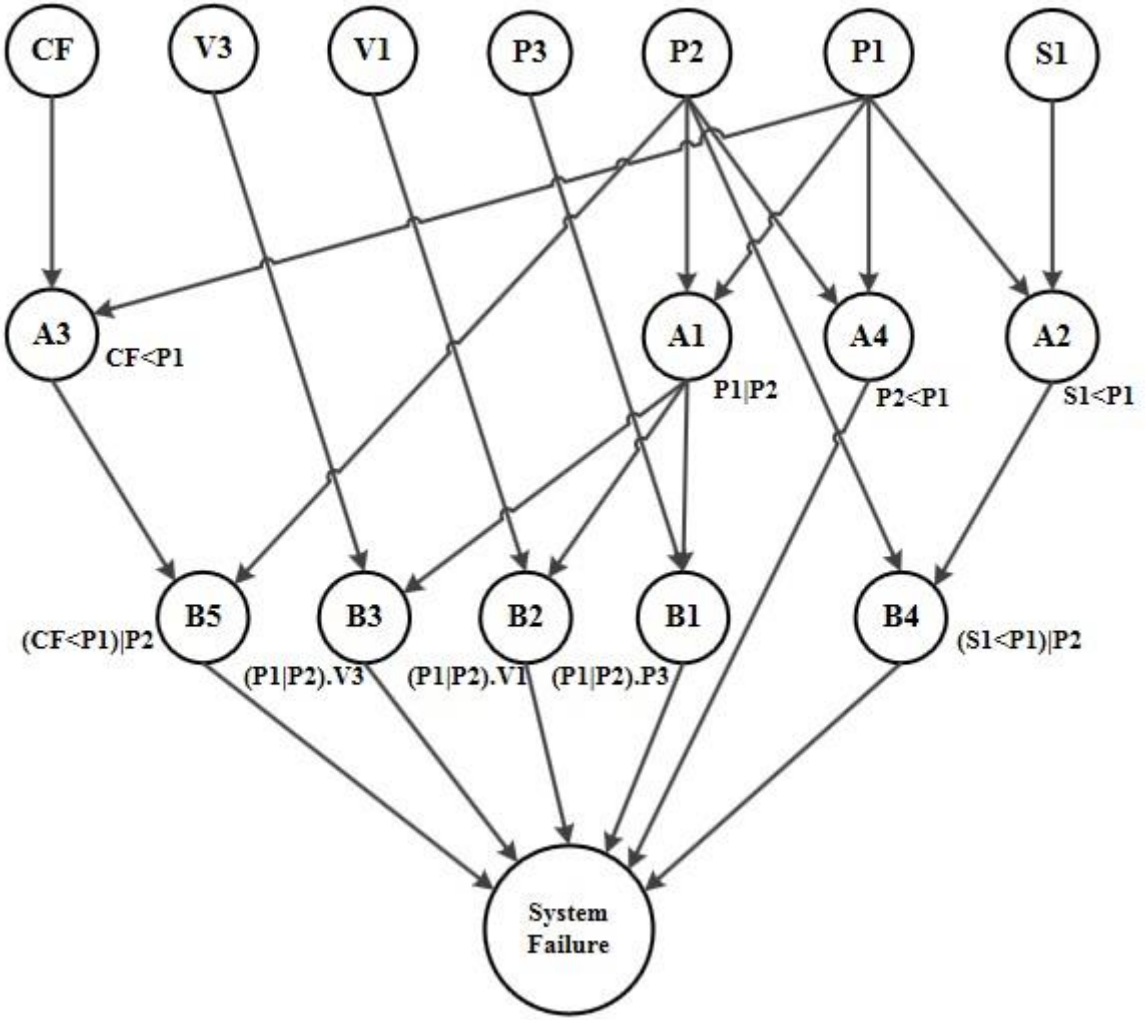
failure rates of components for fuel distribution system

Component	Failure rate/hour
Tanks	1.5E-5
Valve1 , Valve 2	1E-5
Valve3 , Valve 4	6E-6
Pump1 ,Pump2 , Pump3	3.2E-5
Flowmeter Sensor	2.5E-6
Controller	5E-7

Pandora TFT of the fuel distribution system



Equivalent BN



Result of Quantitative Analysis

Top event probability for different values of *n* with *T=10000 hours*

n	Top Event Probability	Average Execution Time (ms)
3	0.1114	15.6
4	0.1159	25.2
5	0.1187	40.6
6	0.1205	74.8
7	0.1218	149.6
8	0.1227	306.0
9	0.1235	493.0
10	0.1242	1011.2
12	0.1250	3070.0
15	0.1260	12879.4
17	0.1264	29162.8
20	0.1268	84889.0

Discussion and Summary

- This methodology provides a probabilistic method of reasoning under uncertainty
- It can work with different distribution of basic event failure rates
- The accuracy of approximation increases as the value of *n* increases
- The value of top event probability converges towards a ceiling value as the value of *n* increases
- Execution time increases by a factor of approximately 1.6 for every additional interval
- Users can make trade-off between accuracy and execution time

Future Works

- Optimising the size of the CPTs to reduce memory consumption
- Formally proving the equivalence of BNs
- Extend this work by performing diagnostic (backward) analysis
- Performing analysis by putting observation about the states of the system components

Thank You