Grid-Sampling Optimisation of Safety Systems

Prof. John Andrews & Dr Lisa Bartlett

Loughborough University, UK

Content Of Presentation

- Traditional Approach to Safety System Design.
- Process for Optimal System Design.
- Application Safety System - HIPS.
- Grid-Sampling Optimisation.
- Methodology & Results.
- Conclusions

Traditional Approach to Safety System Design

- Preliminary Design

  ↓

  Analysis ← Redesign

  ↓

  Appraisal

- Acceptance Criteria - Probability of failure below preset level.

- Adequate design not optimal.

Process for Optimal System Design

- Obtain the best system performance within resource limits available.
-

Initial Design

Design Variables

Analysis (Fault Trees -> BDDs)

Optimisation (Optimisation Program)

Optimal Design

Purpose of Research - Latest Developments

- Optimal performance achieved using fault trees in 1994, Andrews.
- Approach improved to use Binary Decision Diagrams (BDDs) instead of Fault Trees to analyse availability of system (1997).
- Optimisation achieved through using Genetic Algorithms (1999).
- THIS PAPER:
  Optimisation using technique referred to as GRID-SAMPLING, incorporating use of BDDs.

Aim of Research

- Application Safety System.

  → HIPS

- Determine design parameters.

  → 12 Variables

- Use method to determine availability of each design.

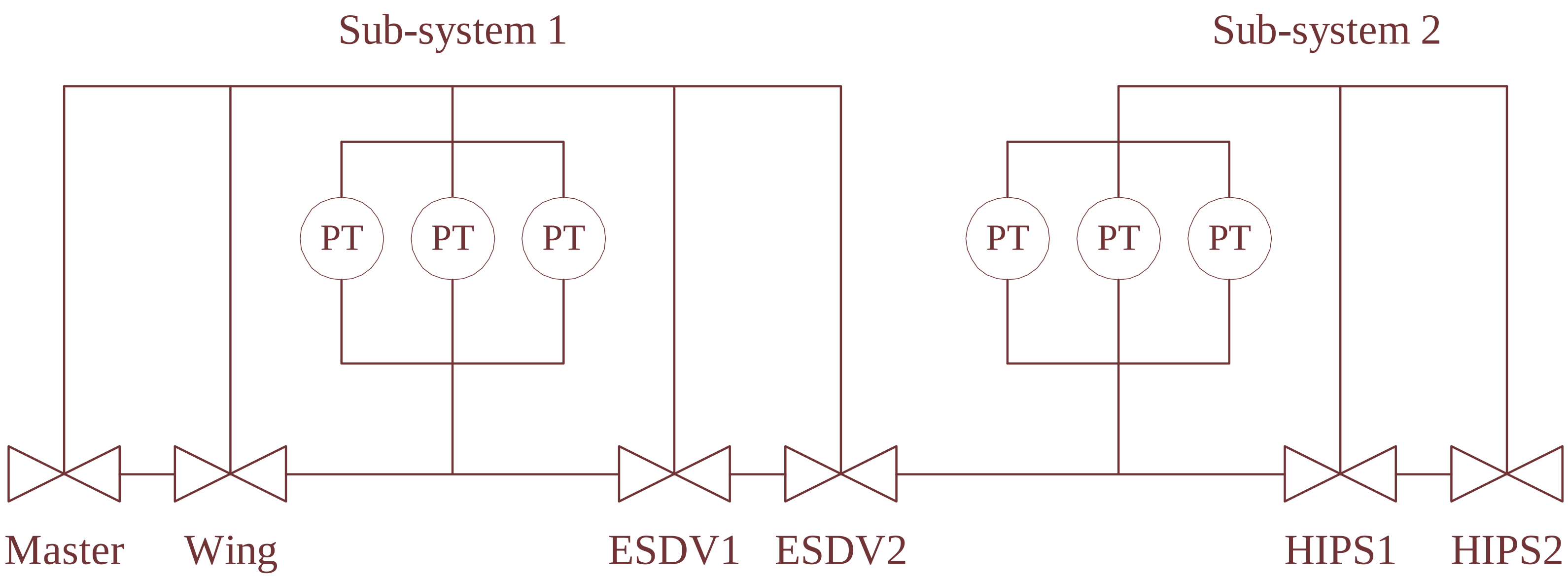
  → BDD

- Find optimisation method to find optimal design not adequate design.

  → GRID SAMPLING

Application Safety System

- Function: prevent high pressure surge passing through system.



Design Paramaters

• How many ESD valves are required (0, 1, 2)?	E (integer)
• How many HIPS valves are required (0, 1, 2)?	H (integer)
• How many pressure transmitters for each subsystem (0, 1, 2, 3, 4)?	N1, N2 (integer)
• How many transmitters required to trip?	K1, K2 (integer)
• Which of two possible ESD/HIPS valves to select?	V1, V2 (Boolean)
• Which of two possible PTs to select?	P1, P2 (Boolean)
• Maintenance test interval in weeks for each subsystem (1 week - 2 yrs)?	$\theta 1$, $\theta 2$ (in practice integer)

TOTAL = 42, 831, 360

Component Data

Component	Dormant Failure Rate	Dormant Mean Repair Time	Spurious Failure Rate	Spurious Mean Repair Time	Cost	Test time
Wing Valve	1.14×10^{-5}	36.0	1×10^{-6}	36.0	100	12
Master Valve	1.14×10^{-5}	36.0	1×10^{-6}	36.0	100	12
HIPS1	5.44×10^{-6}	36.0	5×10^{-7}	36.0	250	15
HIPS2	1×10^{-5}	36.0	1×10^{-5}	36.0	200	10
ESDV1	5.44×10^{-6}	36.0	5×10^{-7}	36.0	250	15
ESDV2	1×10^{-5}	36.0	1×10^{-5}	36.0	200	10
Solenoid Valve	5×10^{-6}	36.0	5×10^{-7}	36.0	20	5
Relay Contacts	0.23×10^{-6}	36.0	2×10^{-6}	36.0	1	2
PT1	1.5×10^{-6}	36.0	1.5×10^{-5}	36.0	20	1
PT2	7×10^{-6}	36.0	7×10^{-5}	36.0	10	2
Computer Logic	1×10^{-5}	36.0	1×10^{-5}	36.0	20	1

Analysing the Design

- Criterion must be determined to quantify how "good" each system design actually is.

- System to work on demand => Minimise system unavailability

$$\min Q_{SYS} = f(E, H, N1, K1, N2, K2, P1, P2, V1, V2, \theta 1, \theta 2)$$

- Consideration also to available resources - must not exceed:
  - Cost (≤ 1000 units)
  - Maintenance downtime (≤ 130 hours)
  - Spurious trip frequency (≤ 1 per/year)

Assessing Performance of Potential System

- Depends on four parts:

  - The probability of system failure, Q_{SYS}.
  - Penalty for exceeding total cost constraint, Cpen.
  - Penalty for exceeding the total MDT constraint, MDTpen.
  - Penalty for exceeding spurious trip constraint, STpen.

- Penalised System Unavailability =

$$Q'_{SYS} = Q_{SYS} + Cpen + MDTpen + STpen \qquad (1)$$

- A means to evaluate each term in (1) is required.

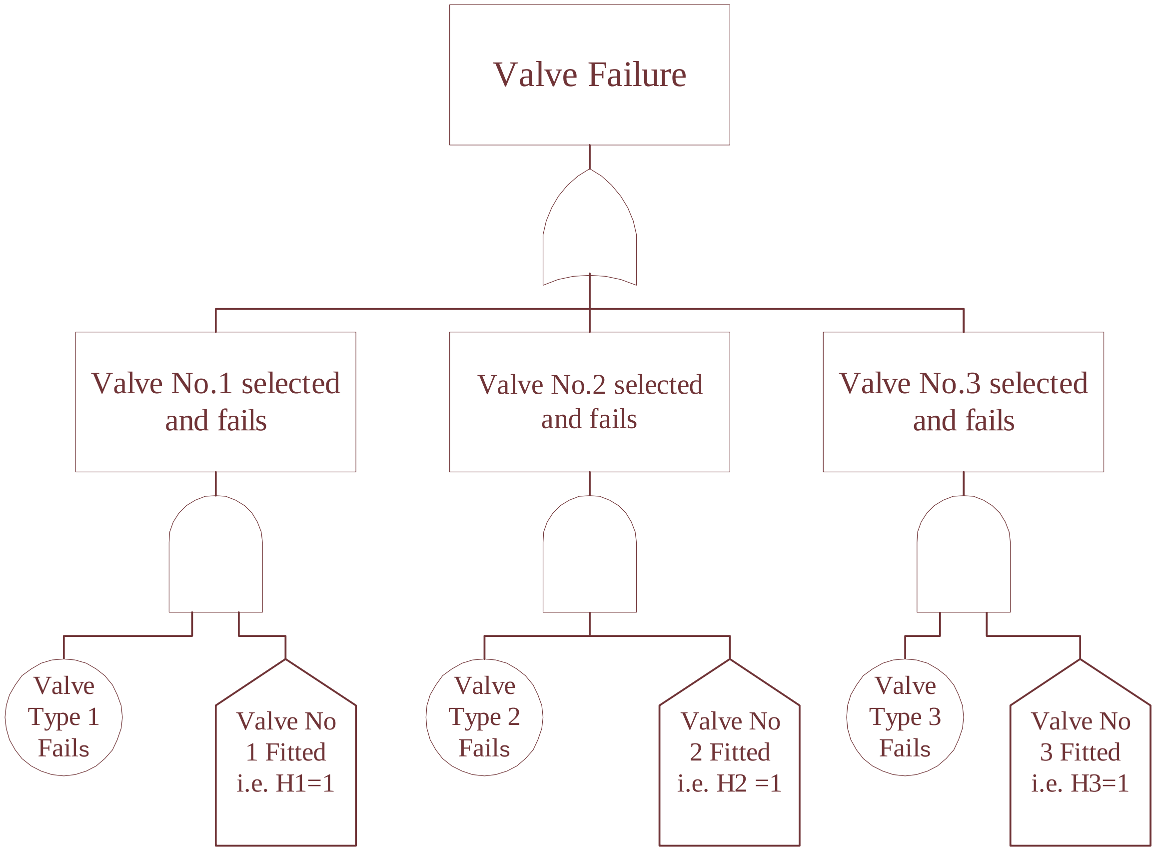
Loughborough University

Evaluating System Unavailability

- No objective function can be formulated.
- Fault trees used to quantify unavailability of each potential design.
- Time consuming to construct fault tree for each potential design.
- Construct one fault tree, incorporating House Events, for all possible designs.

House Events

- Either TRUE or FALSE, are utilised to turn on or off different branches of the tree.



Overall System Fault Tree

- Fault tree to describe all possible design options:
    - 88 primary events
    - 169 gates
- Of 88 primary events:
    - 44 basic events
    - 44 house events
- This fault tree converted to equivalent BDD.

Analysis Using Binary Decision Diagram

- Large fault trees often necessary to use approximations for quantification.
- BDD - latest development in assessing fault tree.
- Process involves converting to BDD format.
- Format considers each basic event in fault tree in turn, considering effect on system when working and failed.
- Failure and repair data for each component used in procedure to calculate unavailability.

Cost and Maintenance Down Time Evaluation

- Cost of system and MDT are both a function of the design variables.

- Cost = Cost(Subsystem1) + Cost(Subsystem2) $\leq$ 1000

- Cost of Sub system 1 =

  $$E(V1C_{V1} + V2C_{V2} + C_S) + N1(P1C_{P1} + P2C_{P2}) + 261$$

  261 is a fixed cost of parts.

Cost and Maintenance Down Time Evaluation

- Cost of Sub system 2 =

  $H(V1C_{V1} + V2C_{V2}+C_S) + N2(P1C_{P1}+P2C_{P2}) + 21$

  21 is a fixed cost of parts.

- Depending on number and type of components used, the total cost of the system can be calculated by substituting the relevant costs into formula.

Cost and Maintenance Down Time Evaluation

- Similarly,

  $$MDT = MDT(Subsystem1) + MDT(Subsystem2) \leq 130$$

- MDT of Sub system 1 =

  $$\frac{52}{\theta 1}\,[E(V1M_{V1} + V2M_{V2} + M_S) + N1(P1M_{P1} + P2M_{P2}) + 47]$$

  47 is MDT of fixed parts in sub-system 1.

Cost and Maintenance Down Time Evaluation

- MDT of Sub system 2 =

  $$\frac{52}{\theta 2}\left[H(V1M_{V1} + V2M_{V2}+M_S) + N2(P1M_{P1}+P2M_{P2}) + 13\right]$$

  13 is the MDT for fixed parts of subsystem 2.

- Depending on number and type of components used, the total MDT of the system can be calculated by substituting the relevant test times into formula.

- Penalties if cost and MDT constraints exceeded.

Spurious Trip Evaluation

- Can not be expressed as function of design variables.
- Evaluated by full system analysis.
- One fault tree constructed to incorporate each potential design using House Events.
- Resulting fault tree analysed using BDD methodology.

Penalties for Exceeding Constraints

- Penalties calculated such that:
  - Small excess penalized small amounts.
  - Further away from limit more penalty.
  - Non-linear penalty function.
- Penalties need to be consistent across constraints.
- All penalties related to cost.

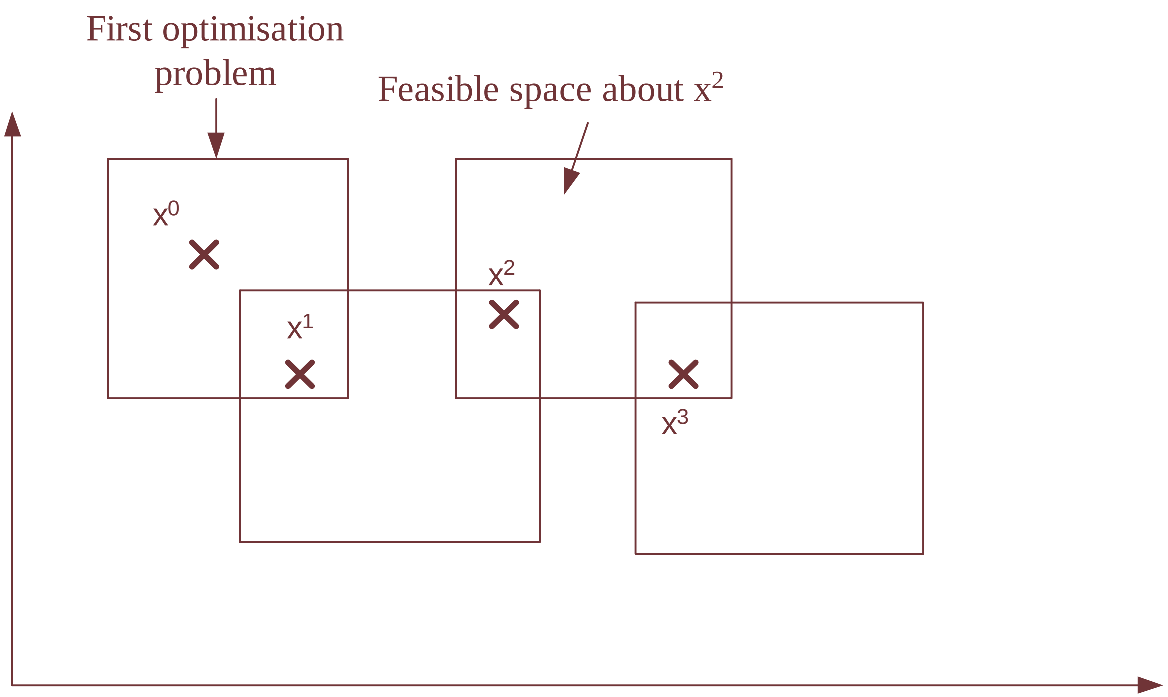
Grid-Sampling Optimisation - General

- Optimisation problem involves integer and Boolean variables, thus some traditional techniques not feasible.
- Method of Grid-Sampling is an iterative scheme, approaches optimum by solving a sequence of optimisation problems.
- Each iteration improves performance.
- When no longer improved due to limitations on system, procedure terminates.

Grid-Sampling Optimisation - Basic Principles

- Assumes some form of objective function for system unavailability.
- Region is defined over which function is considered accurate.
- Initial design chosen.
- Each point is analysed within restricted space to obtain enclosed optimal design.
- New neighbourhood is then constructed around new design.
- Process repeated until optimal design over whole region located.

Grid-Sampling Optimisation - General



- Optimal solution approached by solving a sequence of optimsation problems.

Formulation of Objective Function

- $\min Q_{SYS} = f(E, H, N1, K1, N2, K2, P1, P2, V1, V2, \theta 1, \theta 2)$

- Consider area surrounding a design point.

- Expand Taylors series around current design point.

$$Q_{SYS}(\mathrm{x} + \Delta \mathrm{x}) = Q_{SYS}(\mathrm{x}) + g^{T} \Delta \mathrm{x} + \frac{1}{2} \Delta \mathrm{x}^{T} H \Delta \mathrm{x} + \ldots$$

$x =$ current design vector

$\Delta \mathrm{x} =$ change in design vector

$g^{T} = \nabla Q_{SYS} = \left[\frac{\partial Q_{SYS}}{\partial x_1}, \frac{\partial Q_{SYS}}{\partial x_2}, \ldots.. \right]$

$H =$ Hessian matrix

Loughborough University

Formulation of Objective Function

- Truncate after linear term.

$$Q_{SYS}(\mathrm{x}+\Delta \mathrm{x}) = Q_{SYS}(\mathrm{x}) + g^{T}\Delta \mathrm{x}$$

- Use finite differences to evaluate differential terms.
  - Central differences
  - Forward differences
  - Backward differences

Finite Differences

- Central Differences

$$\frac{\partial Q_{SYS}}{\partial x_i} = \frac{Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i + dx_i, x_{i+1}, \ldots, x_n) - Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i - dx_i, x_{i+1}, \ldots, x_n)}{2dx_i}$$

- Forward Differences

$$\frac{\partial Q_{SYS}}{\partial x_i} = \frac{Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i + dx_i, x_{i+1}, \ldots, x_n) - Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)}{dx_i}$$

- Backward Differences

$$\frac{\partial Q_{SYS}}{\partial x_i} = \frac{Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) - Q_{SYS}(x_1, x_2, \ldots, x_{i-1}, x_i - dx_i, x_{i+1}, \ldots, x_n)}{dx_i}$$

Defining Region

- Truncating Taylors expansion gives approximation.
- Objective function assumed accurate in a small neighbourhood.
- Range limited by

$$x_i^j - \Delta x_{iL} \le x_i^T \le x_i^j + \Delta x_{iU}$$

- Objective function evaluated using Taylors expansion for each point in restricted design, and optimum point selected.

Steps of Optimisation Algorithm

1) Construct fault tree by which the use of house events is capable of representing the causes of dormant failure for each possible system design.

2) Construct a fault tree representing the causes of spurious trip for each possible system design, using house events.

3) Select some feasible initial design. Convert system fault tree to alternative BDD representation. Set corresponding house events and use relevant component data to determine system unavailability. Repeat procedure for spurious trip fault tree and check that the spurious trip rate constraint is met.

Steps of Optimisation Algorithm

3) Example initial design.

E	K_1	N_1	H	K2	N2	V	P	θ_1	θ_2
1	2	2	1	1	1	1	1	40	50

Determine Q_{SYS} : 3.95×10^{-3}

Determine F_{SYS} : 0.420

Feasible if other constraints met:

MDT 101.66

Cost 882

If initial design not feasible, select another.

Steps of Optimisation Algorithm

4) Choose the form of objective function that can be used to represent Q_{SYS} in the neighbourhood of the current design vector.

5) When each design derivative has been evaluated the objective function is given by:

$$Q_{SYS}(x + \Delta x) = Q_{SYS}^{j} + \sum_{i=1}^{n} \left(\frac{\partial Q_{SYS}}{\partial x_i} \right)^{j} dx_i$$

Loughborough University

Steps of Optimisation Algorithm

Example: Calculate each derivative

Parameter	E
Initial Design	1
Range (±1)	0, 2
Difference	Central

Calculate Q_{SYS} with E = 0, other parameters as initial design
Calculate Q_{SYS} with E = 2, other parameters as initial design
Subtract two values and divide by 2.

→ Partial derivative Q_{SYS} with respect to E.

Repeat for all other variables.

Loughborough University

Steps of Optimisation Algorithm

6) Minimise system unavailability over current design space.

Example:
Option 1: E = 2, N1 = 2, other parameters as initial design.

Therefore Unavailability =

Change in variables (1)

$$Q_{SYS}(x+\Delta x) = Q_{SYS}^{j} + \frac{\partial Q_{SYS}}{\partial x_E} dx_E + \frac{\partial Q_{SYS}}{\partial x_{N1}} dx_{N1}$$

Unavailability Initial Design

Partial Derivatives

Repeat for each design possibility in design space.

Steps of Optimisation Algorithm

7) Check optimal design.

Exact system unavailability of design needs to be checked.

Relevant fault trees set up and evaluated.

If difference found between Taylors series approximation and exact, indicates that objective function is not true in neighbourhood selected.

If unavailability less fit then point rejected.

Steps of Optimisation Algorithm

8) Locate optimal design in design region.

If new design vector is better than initial design given all constraints met, accept and repeat steps 4 onwards.

Else, reduce neighbourhood around initial design, repeat to find optimal design in this reduced neighbourhood. Keep reducing neighbourhood until:

> If locate optimal repeat steps 4 onwards.
> Else conclude best is initial design.

Process terminates when no design in neighbourhood can be found with lower unavailability.

Results

- Approach tested on eight initial designs.
- Best designs for each were:

Test No	E	K1/N1	H	K2/N2	V	P	θ1	θ2	Q_{SYS}	F_{SYS}	Cost	MDT
1	0	2/3	2	1/3	2	1	27	36	7.97×10^{-4}	0.847	842	129
2	0	1/2	2	1/2	2	1	34	26	7.23×10^{-4}	0.977	802	129.6
3	0	2/2	2	1/2	2	1	31	29	9.34×10^{-4}	0.847	822	130
4	0	1/3	0	0/0	1	1	16	0	1.43×10^{-2}	0.411	301	126.7
5	0	1/2	2	1/2	2	1	38	24	7.5×10^{-4}	0.977	802	129.2
6	0	1/3	2	2/3	2	1	33	28	7.57×10^{-4}	0.847	842	129.9
7	0	2/3	1	1/3	1	1	40	40	2.51×10^{-3}	0.67	672	85.5
8	2	1/1	1	1/1	1	1	40	50	4.2×10^{-3}	0.807	982	108.2

- Proves very effective if appropriate initial design chosen.
- Tests 2 & 5 very similar results.
- Lowest availability in test 2.
- Constraints met in all cases.

Conclusions - The Technique

1) Very effective optimisation procedure if an appropriate initial design is chosen.

2) Problems arise with interactions of parameters.

3) Allows full use of resources, MDT distributed across both systems.

4) Allows number of designs to be evaluated without having to calculate exact unavailability.

5) Appropriate technique used in combination with algorithm to find good region, this can hunt out best in optimal space.

Conclusions - The Process

1) Better use can be made of techniques such as the BDD in the design process.

2) Utilised as part of a design optimisation technique better use of resources can be achieved to improve system performance.

3) Algorithm is flexible and any type of design variation can be incorporated.

4) Best designs as opposed adequate designs are achieved.