University of Southampton

SESA6085 – Advanced Aerospace Engineering Management

Lecture 13

2024-2025

Dr David Toal

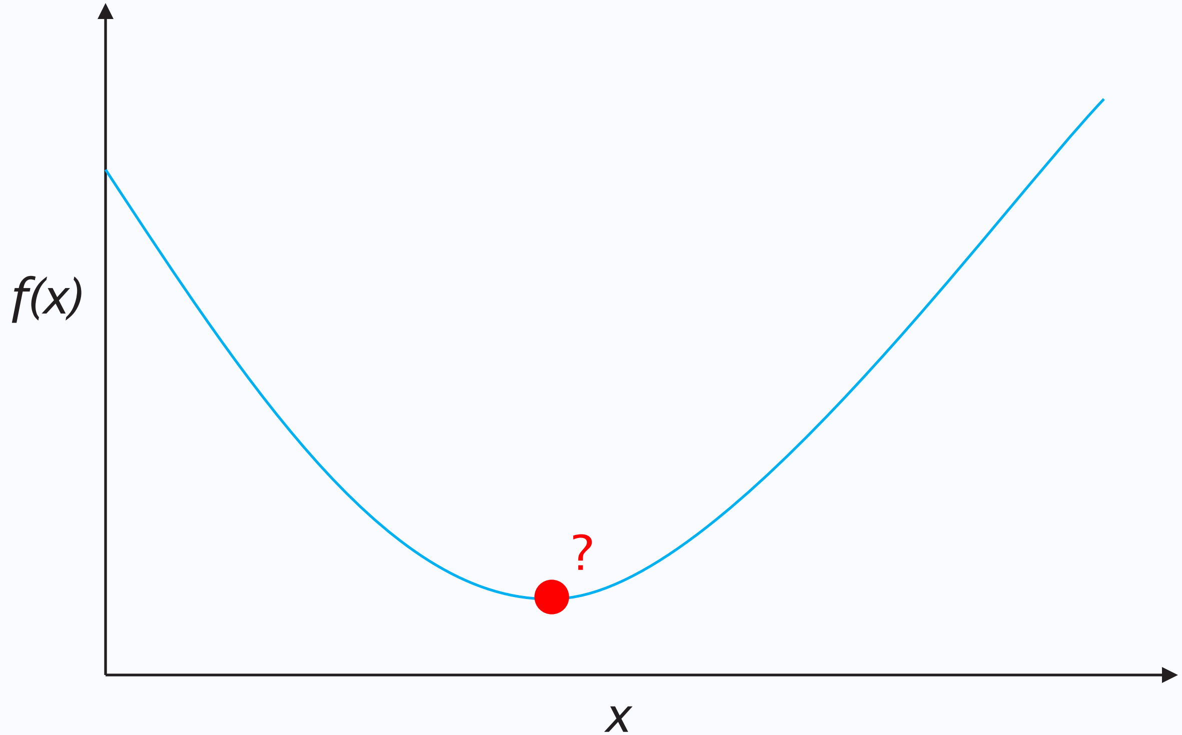
Design Optimisation

Design Optimisation

- What do we mean by "design optimisation"?
  - Generally, it's the process of iteratively improving a design based on some performance metric
  - Could be automated but not necessarily
- Given an arbitrary function, *f(x)*, how would you optimise it?
  - Global / local optimisers, GA, SA, BFGS, Simplex etc.
  - Random search
  - Solve it analytically

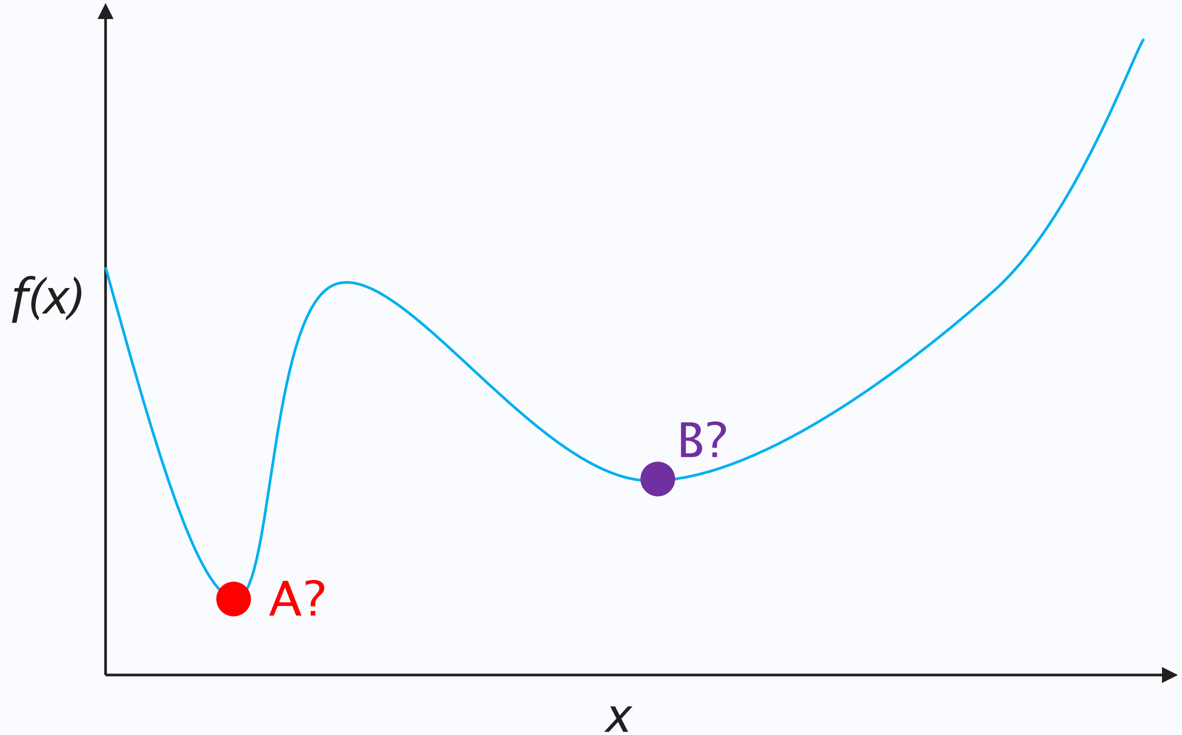
Optimisation Example #1

- Minimise *f(x)*
- What is the optimum and why?



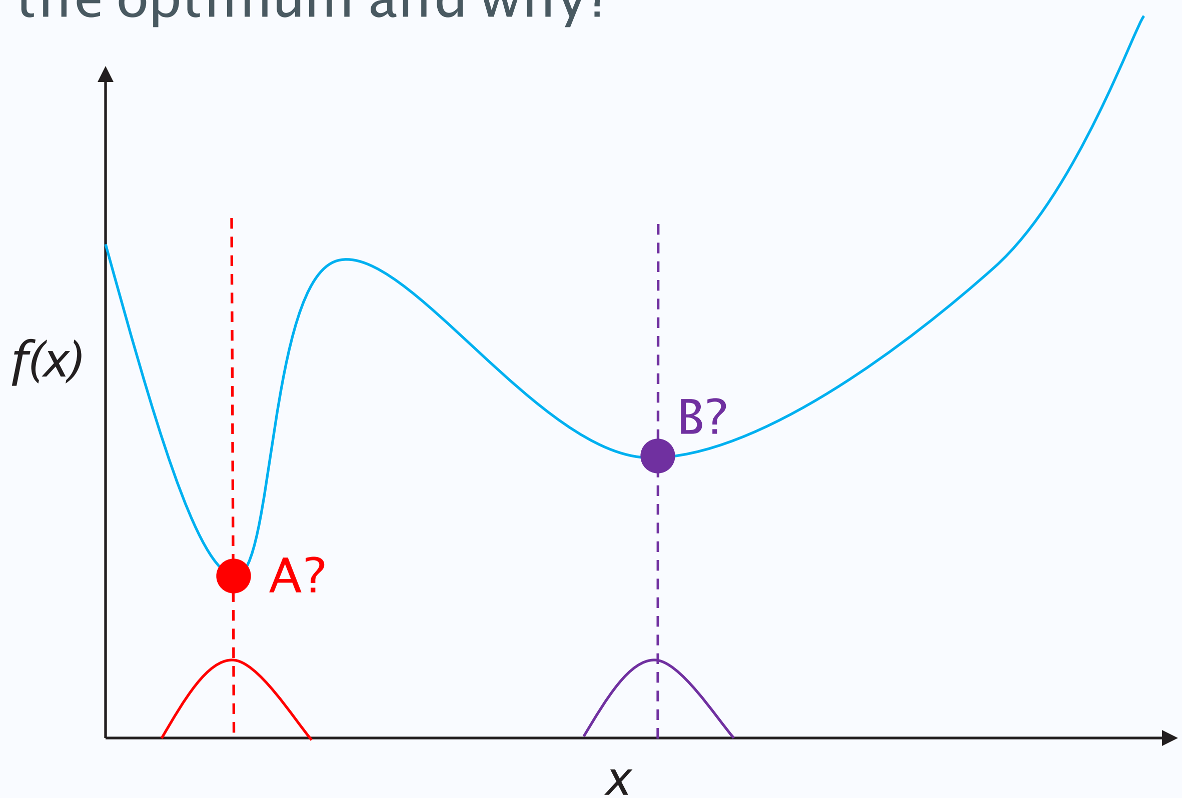
Optimisation Example #2

- Minimise *f(x)*
- What is the optimum and why?



Optimisation Example #2b

- What if the value of x was uncertain?
- What is the optimum and why?



Robust Design or Design Under Uncertainty

Robust Design Optimisation

- Similar to standard optimisation i.e. trying to find a set of parameters to give the best value of a design metric
- However…
  - *f(x)* usually represents a model or approximation of the real world therefore there may be a mismatch between the model's optimum and the true optimum
  - Even if there is no mismatch manufacture of the optimum design may be difficult resulting in errors
  - Standard optimisation does not allow for fluctuations
  - Optimised systems may be sensitive to small changes

Robust Design Optimisation

- In a robust design optimisation we therefore search for "robust" solutions to our optimisation problem
- Also sometimes called "Design Under Uncertainty"
- As we will see this has a number of similarities to designing for reliability

Sources of Uncertainties

- Design uncertainties can come from a variety of sources
  - Changing environmental & operating conditions
  - Production tolerances
  - Uncertainties in system outputs (modelling uncertainties)
  - Feasibility uncertainties concerning how well constraints are met

Wing Example

- Consider the design optimisation of a wing
- Operating condition & environment uncertainties?
  - Angle of attack, Mach no., altitude etc.
- Production tolerance uncertainties?
  - Shape accuracy, presence of rivets etc.
- Uncertainties in model outputs?
  - Turbulence model, RANS (unsteadiness ignored) etc.
- Feasibility constraints?
  - Pitching moment & lift coefficients etc.

Aleatory & Epistemic Uncertainties

- A slightly different classification of uncertainties is often used in the literature
  - Uncertainties are differentiated into objective & subjective uncertainties
- Objective (*aleatory*) uncertainties:
  - Uncertainties due to physical nature (temperature, material parameters etc.)
  - Intrinsically have a stochastic nature
  - Cannot be removed, the designer has to live with them
  - Typically modelled using some form of PDF

Aleatory & Epistemic Uncertainties

- Subjective (*epistemic*) uncertainties:
  - Reflect the designers lack of knowledge
  - Includes uncertainties about the model used to describe the physics of the system
  - Includes errors from numerical methods

Wing Example

- Let's consider our wing design example again, are the following aleatory or epistemic uncertainties?
- Angle of attack?
  - Aleatory – known variation from the flight characteristics, can be modelled with a known PDF
- Turbulence model settings?
  - Epistemic – exact settings everywhere are unknown without further investigation

Modelling Uncertainties

- Uncertainties can be modelled in different ways:

1. Deterministic – parameter domains are defined in which uncertainties can vary
   - Epistemic or aleatory uncertainties
   - E.g. due to the discretization the error is ± 2
2. Probabilistic – defines probability or likelihood of a certain event occurring
   - Aleatory uncertainties
   - E.g. the variation in height is normally distributed with mean 5 and standard deviation 0.1

Solving Robust Design Problems

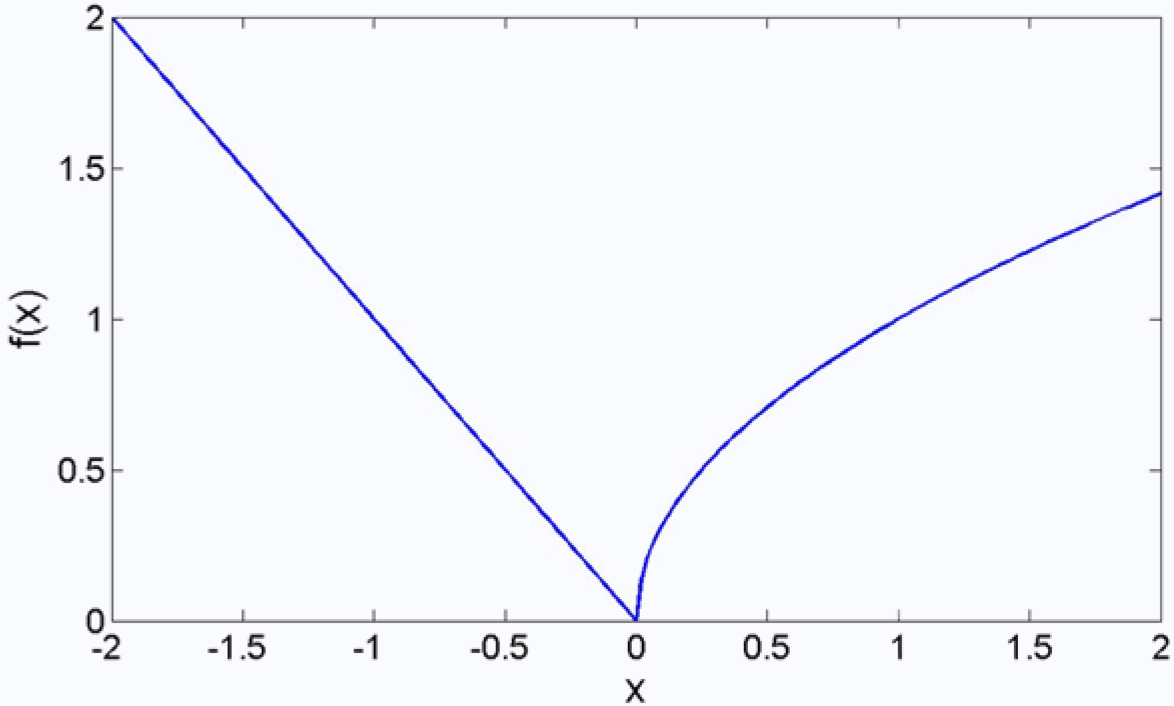
- To solve robust design problems a robust counterpart of *f(x)* is usually derived
- Techniques include:
  - Robust regularisation
  - Expectancy measures (aggregation or induced distribution function evaluation)

Robust Regularisation

- Robust regularisation considers the max of $f(x)$ in some range ε around the point in question
- This can be used with deterministic uncertainties where the bounds are known
- Generally considered as a "worst case" strategy
- Selecting too large a range can give poorly performing designs e.g.
  - An aircraft is very strong but is too heavy

Robust Regularisation

- Find the robust optimum for the following function:
- $f(x) = \begin{cases} -x, & \text{if } x < 0 \\ \sqrt{x}, & \text{if } x \geq 0 \end{cases}$ subject to $\varepsilon = 0.25$



Robust Regularisation

- Using robust regularisation *f(x)* now becomes:

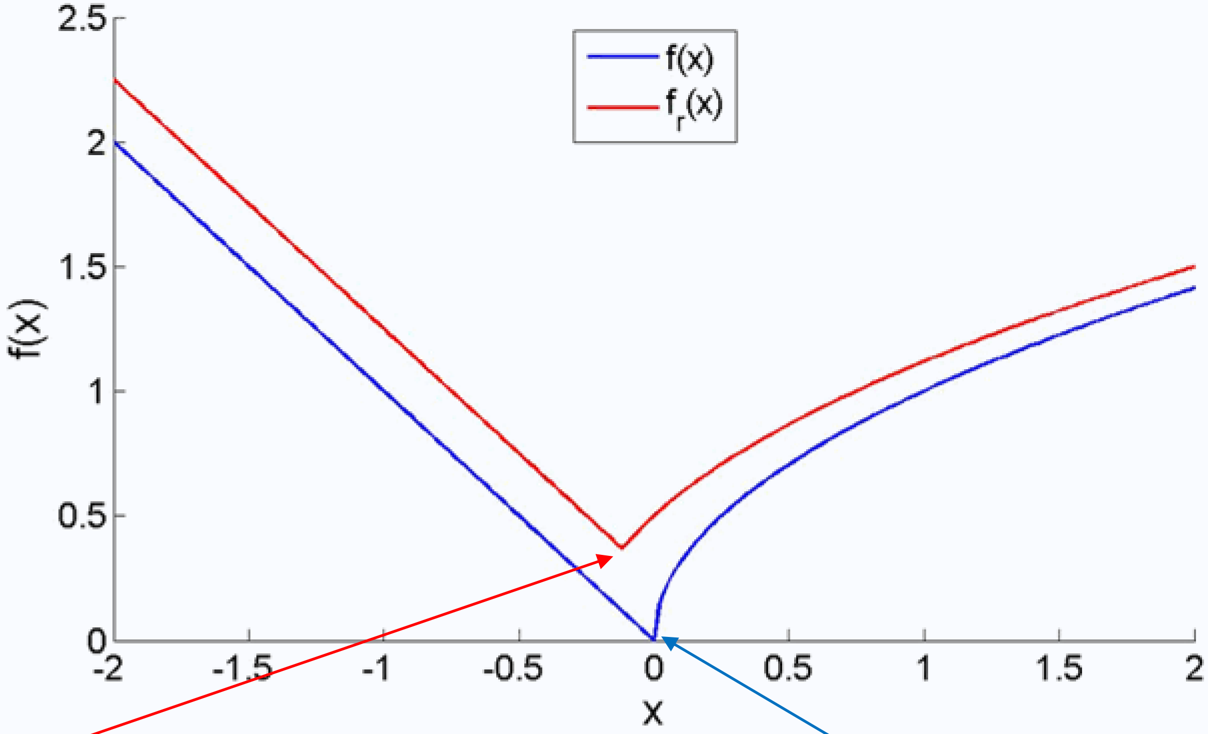
$$f_r(x) = \max[f(x \pm \varepsilon)]$$

- Therefore, becoming a "minimax" optimisation:

$$\min[\max[f(x \pm \varepsilon)]]$$

- N.B. For this particular problem it is possible to derive an equation for $f_r(x)$, however, this may not always be the case

Robust Regularisation



Robust optimum (ε=0.25)

Non-robust optimum

Expectancy Measures of Robustness

- Given the issues with robust regularisation it might be better to consider robustness based on a probability
- In such a case the inputs become random and based on a PDF which reflects prior knowledge
- As the inputs are random the function *f(x)* becomes random
- The resulting function can be dealt with in two ways:
    1. Aggregation
    2. Randomised approaches

Aggregation

- The process of aggregation produces one or more integral measures of robustness e.g.

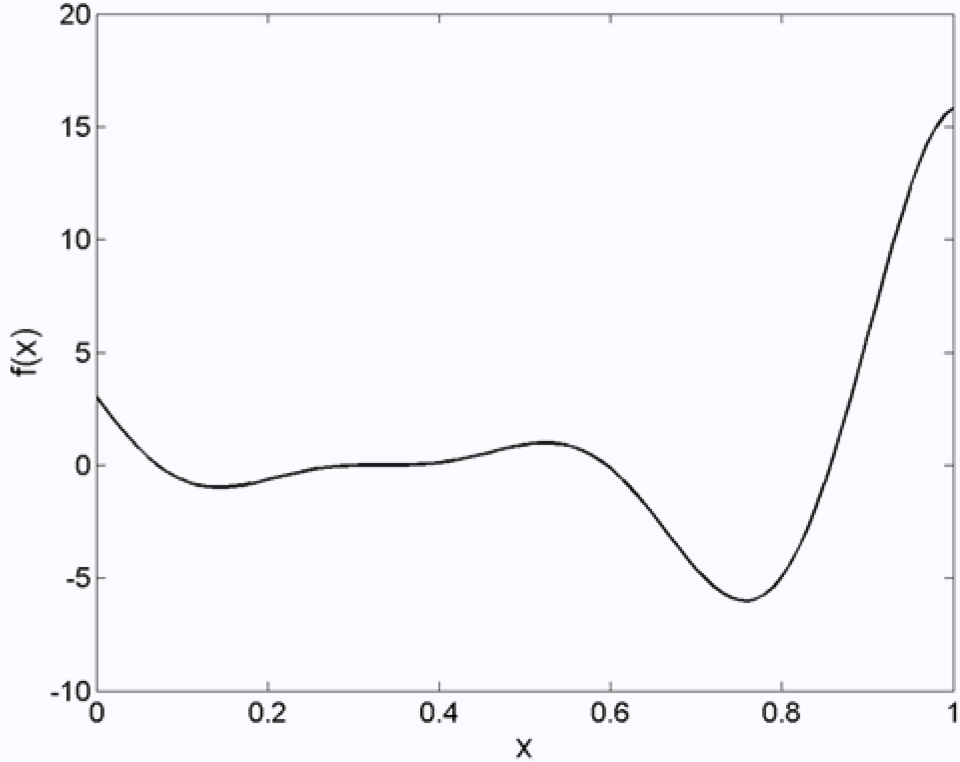
$$F_1(x) = \int f(x+\varepsilon)P(\varepsilon)d\varepsilon$$

- This is sometimes called the "effective fitness" and is almost like a mean value of *f(x)* over the distribution
- Alternatively if we are interested in plateau-like regions we may wish to use some form of dispersion metric e.g.

$$F_2(x) = \int (f(x+\varepsilon) - f(x))^2 P(\varepsilon)d\varepsilon$$

Aggregation Example

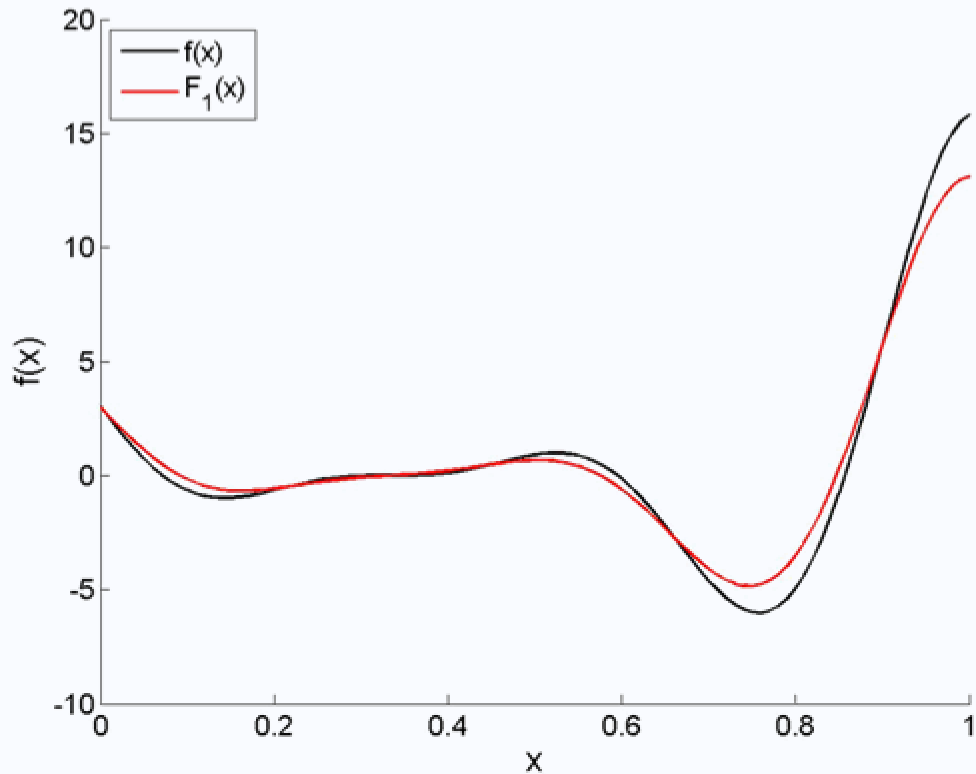
- Let's consider the robust optimisation of the following function
- Where might the optimum of F_1 and F_2 be?



Aggregation Example

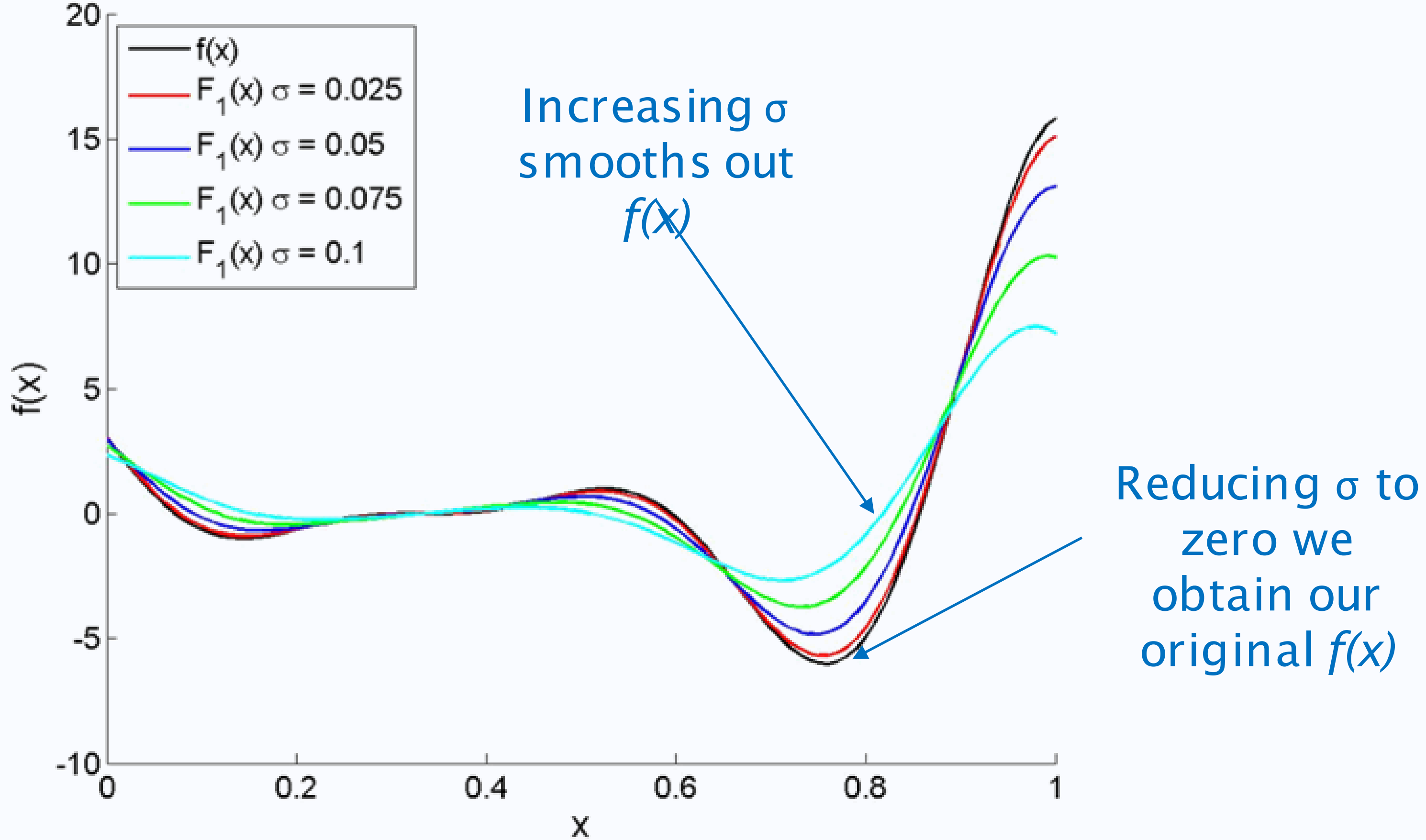
- Let's assume that x varies according to a normal distribution with mean x and standard deviation 0.05

$$F_1(x) = \int f(x+\varepsilon)P(\varepsilon)d\varepsilon$$



Aggregation Example

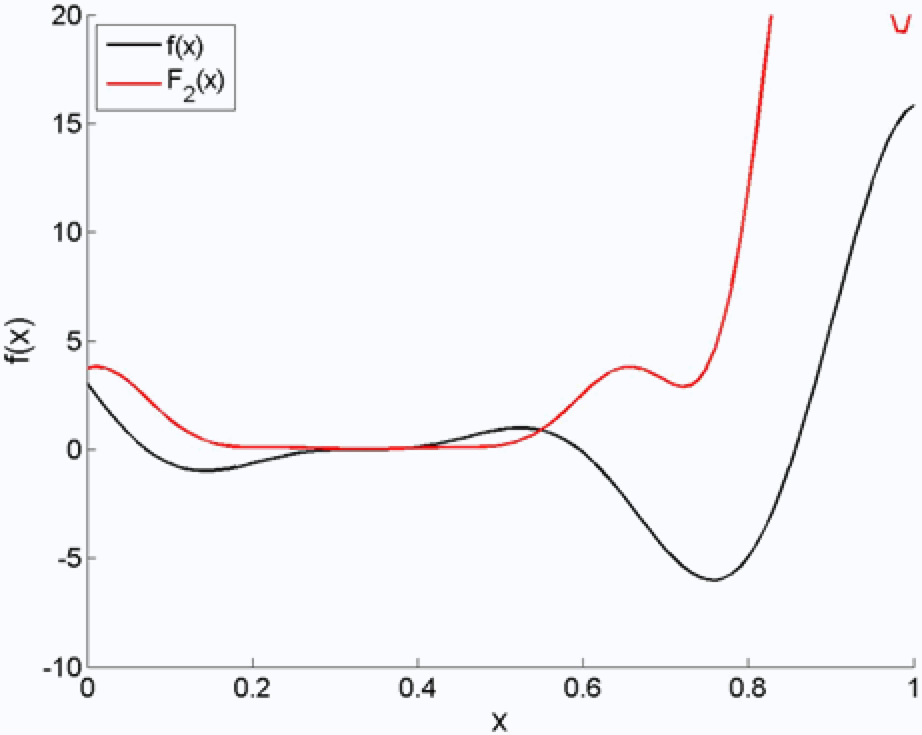
- What will happen as the standard deviation changes?



Aggregation Example

- What about F_2 if the standard deviation is 0.05?

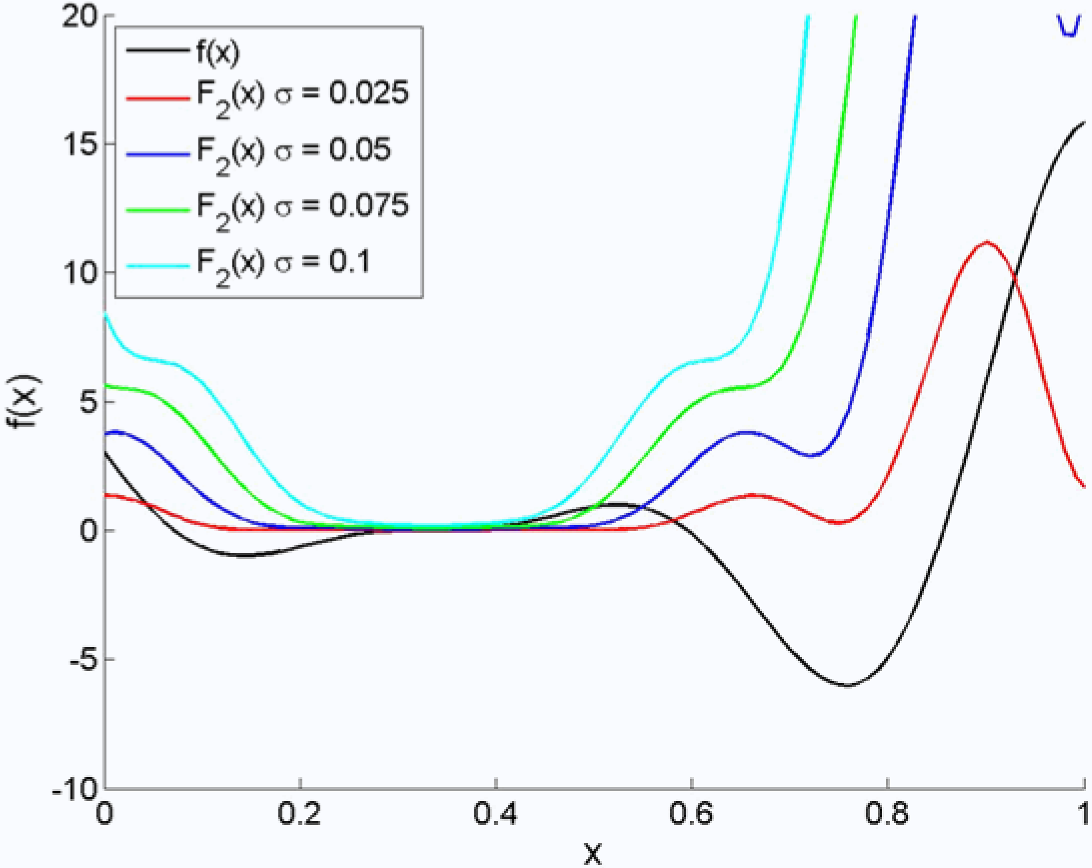
$$F_2(x) = \int (f(x+\varepsilon) - f(x))^2 P(\varepsilon) d\varepsilon$$



Aggregation Example

- What will happen as the standard deviation changes?

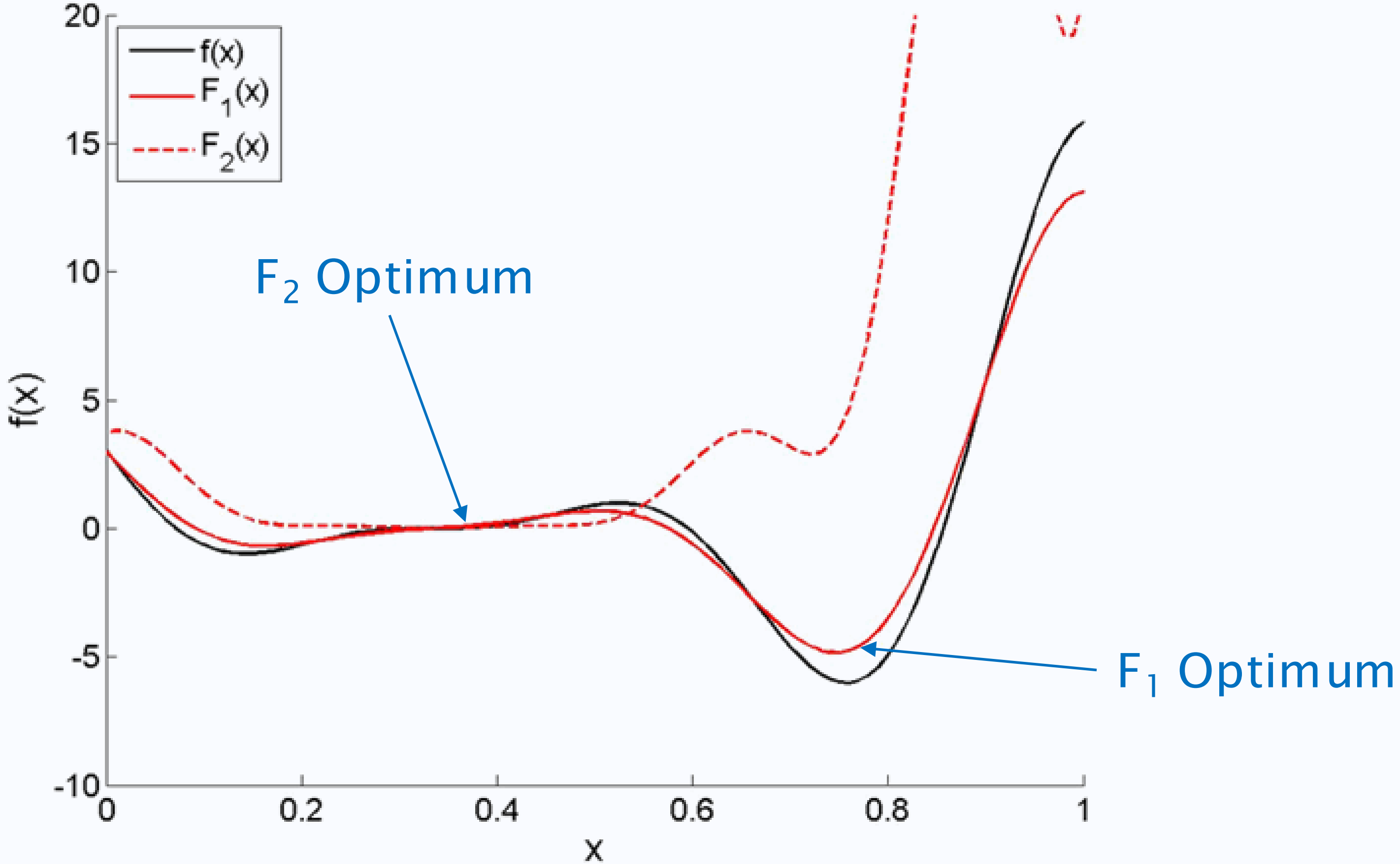
Increasing σ increasingly penalises areas in *f(x)* with any variation



As σ goes to zero F_2 moves towards a constant value of zero i.e. no variation

Aggregation Example

- Plotting both F_1 and F_2 on the same graph what do we notice?

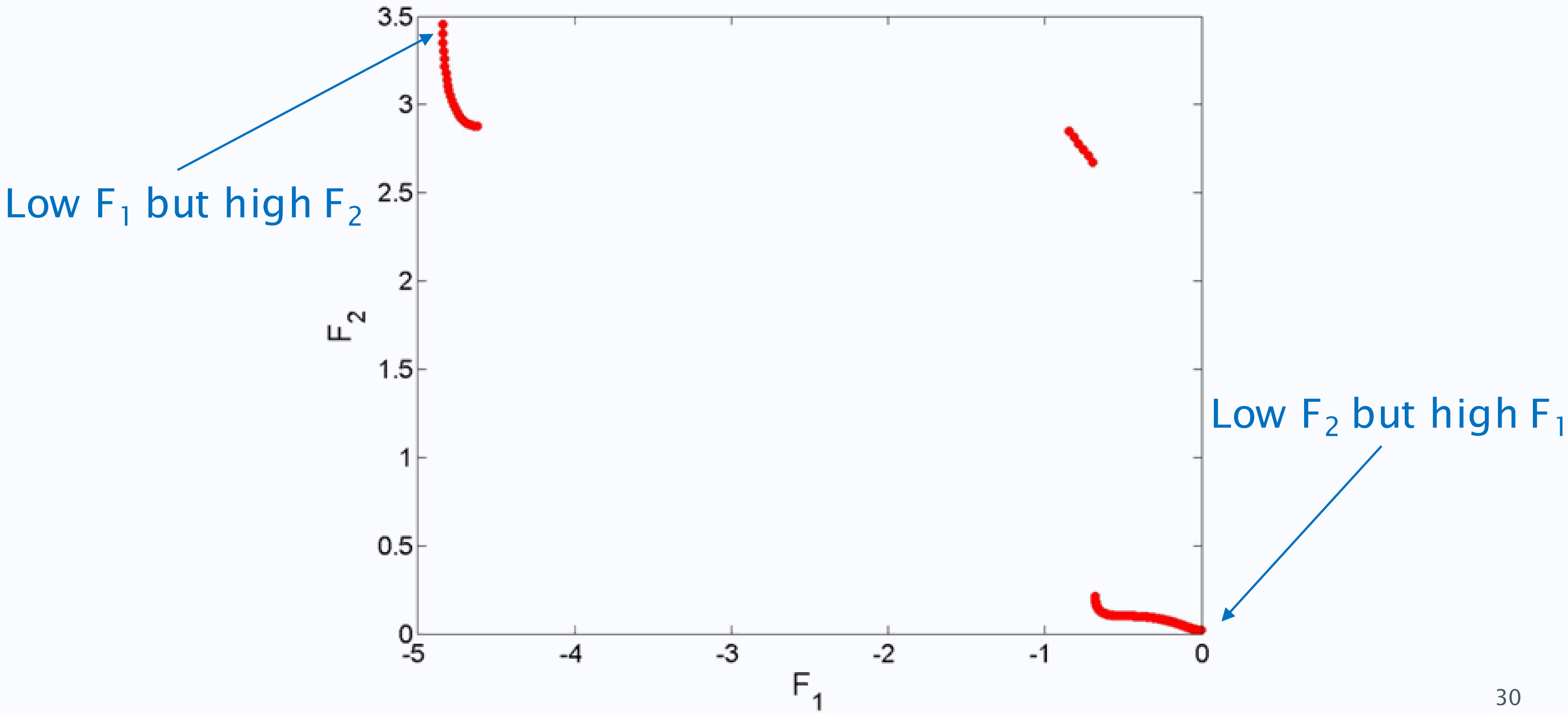


Aggregation

- If we want the best “mean” performance and low variance we want some combination of F_1 and F_2
- Often this leads to multiple conflicting criteria
  - Optimising for F_1 may lead to a design with a high F_2
- How do we combat this?
  - We could use some form of weighted sum of the two
  - Or we apply a multi-objective optimisation to create a Pareto front of F_1 and F_2

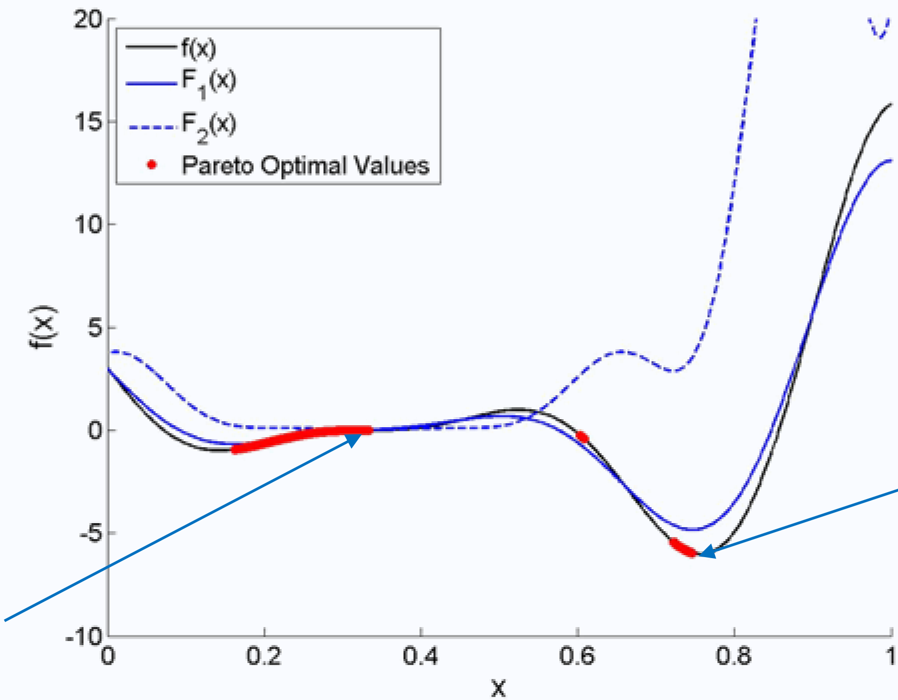
Pareto Front Example

- Performing a multi-objective optimisation on the previous example would create the following Pareto front



Pareto Front Example

- Given the Pareto front a trade-off can be performed by designers
- The following plot indicates where the Pareto optimal points are located along the design variable, *x*



Low F_1 but high F_2

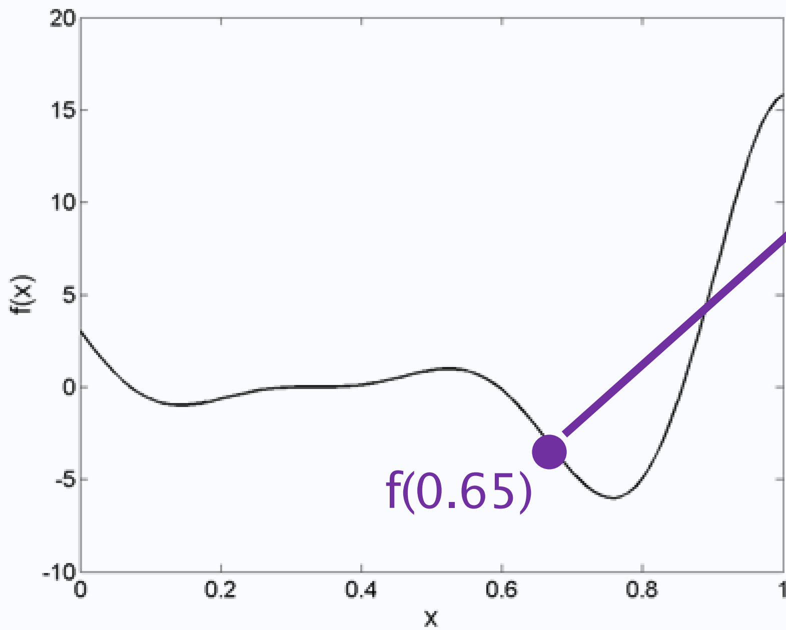
Low F_2 but high F_1

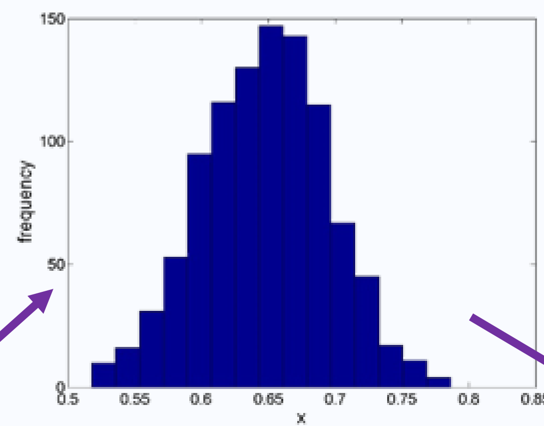
Randomised Approaches

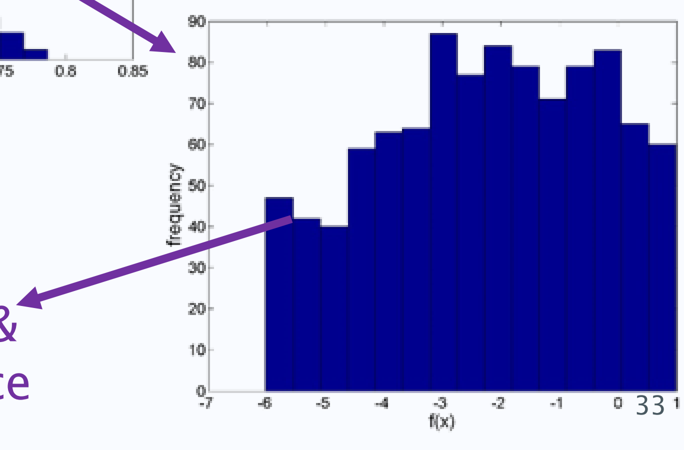
- Aggregation may not, however, be possible in a lot of engineering applications
  - There is no simple analytical expression for $f(x)$ and/or the aggregated version
- How then do we solve such problems?
- Given that our uncertainties are probabilistic then the objective function $f(x)$ becomes a random function
- We can then use the results of this random function in our optimisation

Monte Carlo Strategies

- Given a design point of interest, *x*, we use our random function to calculate a mean, variance etc.
- We can then use these values as inputs to our optimisation algorithm(s)





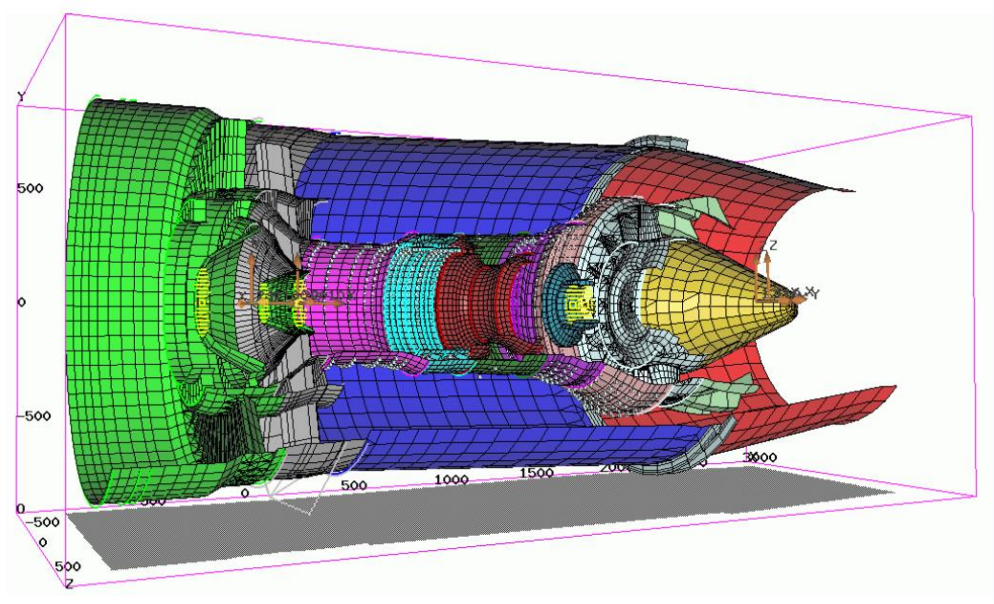


Surrogate Modelling

- Monte Carlo (MC) strategies whilst very general can still be expensive to perform
- Surrogate models (otherwise known as meta models or response surfaces) can be used as an approximation to:
  - The true function, $f(x)$, which the MC can evaluate
  - The results of the MC i.e. the mean and variance etc.
  - Or a combination of the two

Case Study #1 ‡

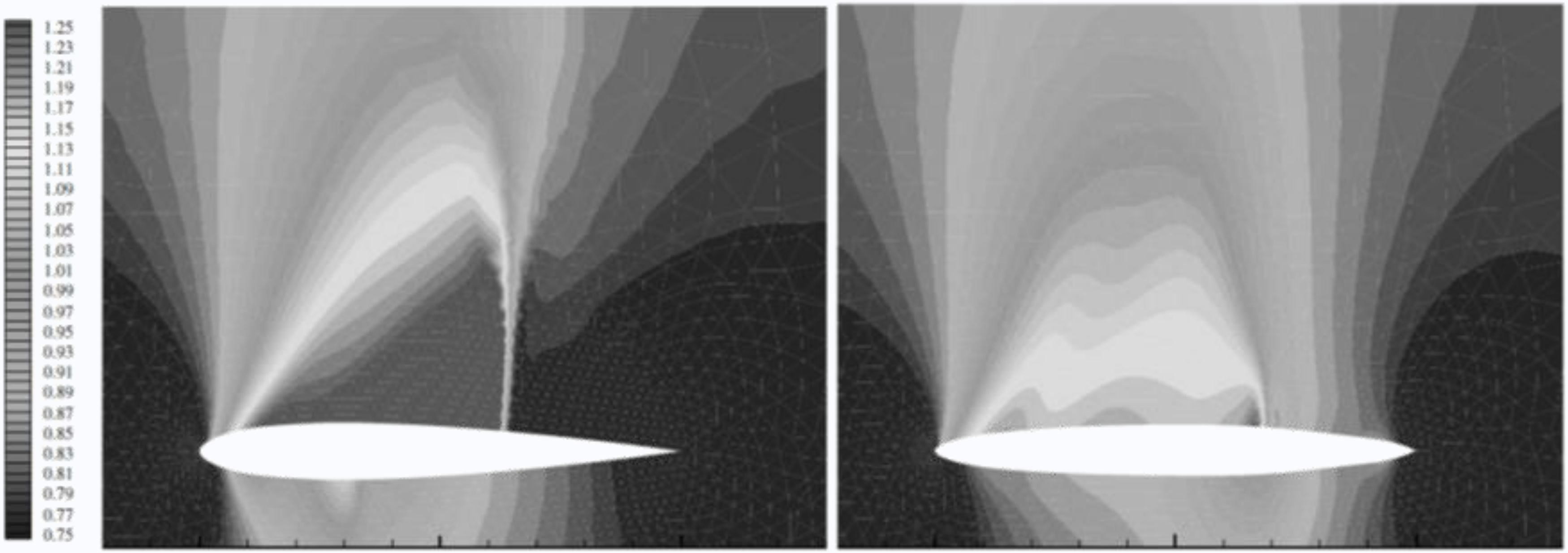
- Employs a surrogate model and a multi-objective evolutionary algorithm to optimise for mass, SFC, and mean and standard deviation of the internal stresses
- Considers each design over a range of applied loads



‡Voutchkov, Keane & Fox, “Robust Structural Design of a Simplified Jet Engine Model Using Multi-Objective Optimization”, AIAA Multidisciplinary Analysis & Optimization Conference, 2006

Case Study #2‡

- Considered the robust design of an aerofoil with respect to changing Mach number



(a) NACA-0012 (b) Profile Optimization

‡Li, Huyse & Padula, "Robust Airfoil Optimization to Achieve Drag Reduction Over a Range of Mach Numbers", Structural Multidisciplinary Optimisation, 2002

Threshold Measures of Robustness

- An alternative to the previous methods is to consider the robustness of a design with respect to a threshold
- In such a case a threshold is defined, q, and we aim to ensure the majority of designs are below/above this value
- I.e. we want to maximise,

$$\Pr[f \leq q] \qquad \text{or} \qquad \Pr[f \geq q]$$

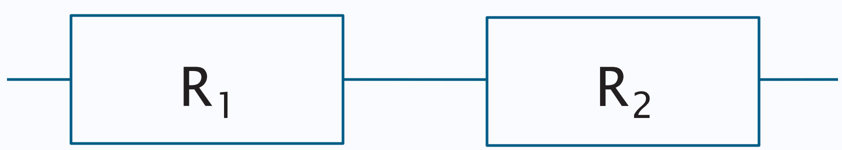
- Is this familiar?

Similarities & Differences

- Generally robust design and reliability design can be considered as two sides of the same coin
- Both consider performance of a system with respect to some form of PDF:
  - Robust design considers behaviour over the whole PDF, mean, standard deviation etc.
  - Reliability considers behaviour at the tails of the PDF, where the system will fail
- Both can be considered as design objectives or constraints and traded against other objectives if necessary

Reliability Optimisation Example

- Let's consider a simple worked example
- A UAV subsystem consists of a speed controller and an electric motor
- The designer of this subsystem must maximise reliability whilst constraining cost and weight to within the limits defined at the system level
- The system can be defined by the following RBD:



Reliability Optimisation Example

- The designer is free to choose the values of R_1 and R_2 by selecting a different model of speed controller or motor but the cost and weight must be less than £120 & 320g respectively
- What is the optimisation problem?

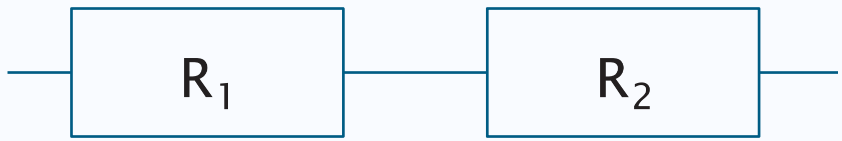
$$Max(R_{sys})$$

Subject to

$$C_{sys}<120 \qquad \& \qquad M_{sys}<320$$

Reliability Optimisation Example

- Given we know the RBD and R_1 and R_2 are parametric it is possible to define an equation for R_{sys}



- Which is?

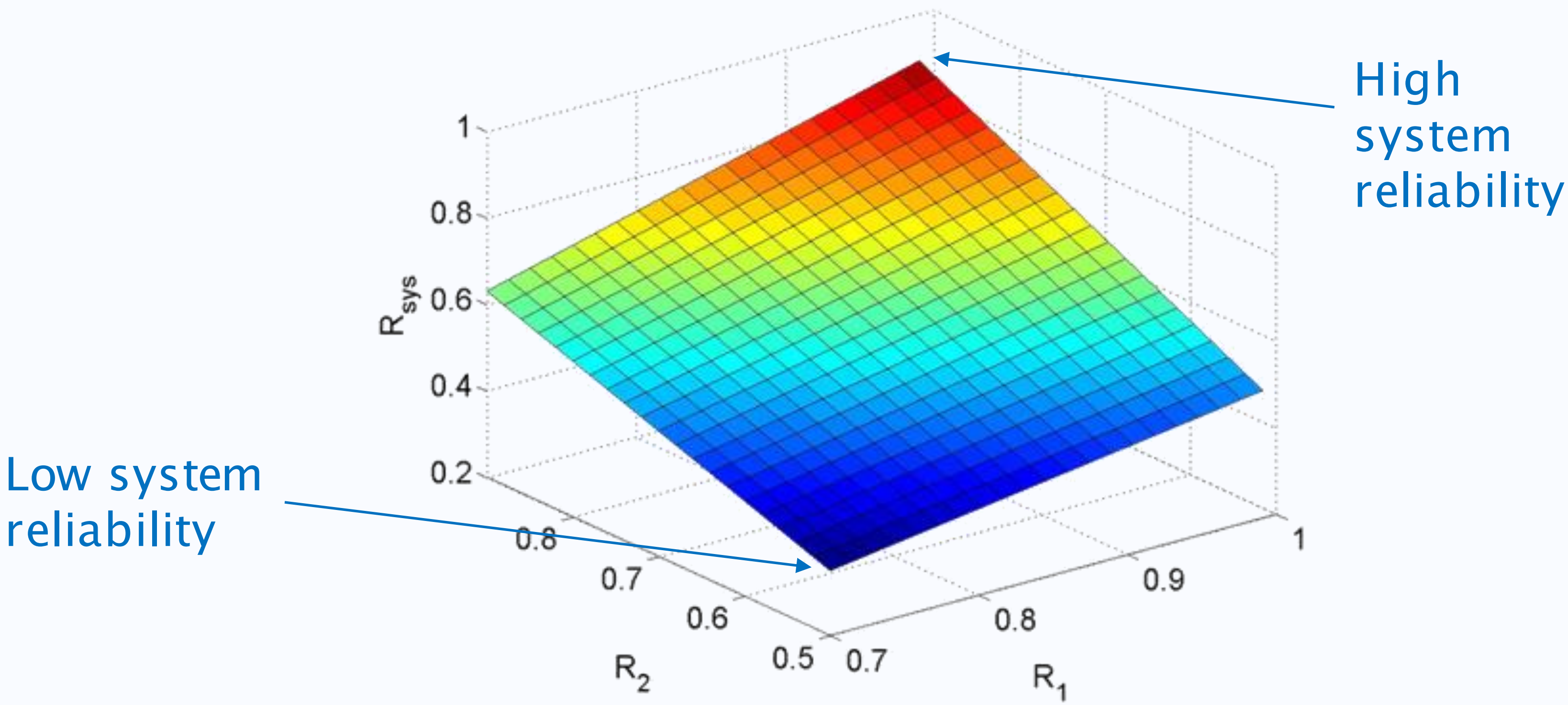
$$R_{sys} = R_1 \times R_2$$

- Similarly because R_1 and R_2 define changes in components we can define corresponding costs and masses

Reliability Optimisation Example

- For this case study we'll assume that we can only have components with R_1 and R_2 within a certain range

$$R_1 \in [0.7,0.99] \quad \& \quad R_2 \in [0.5,0.9]$$

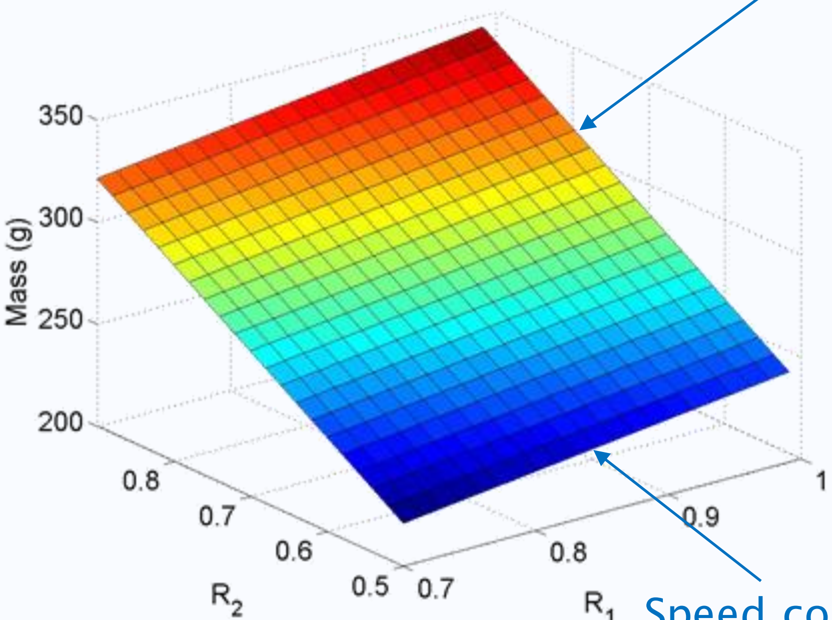


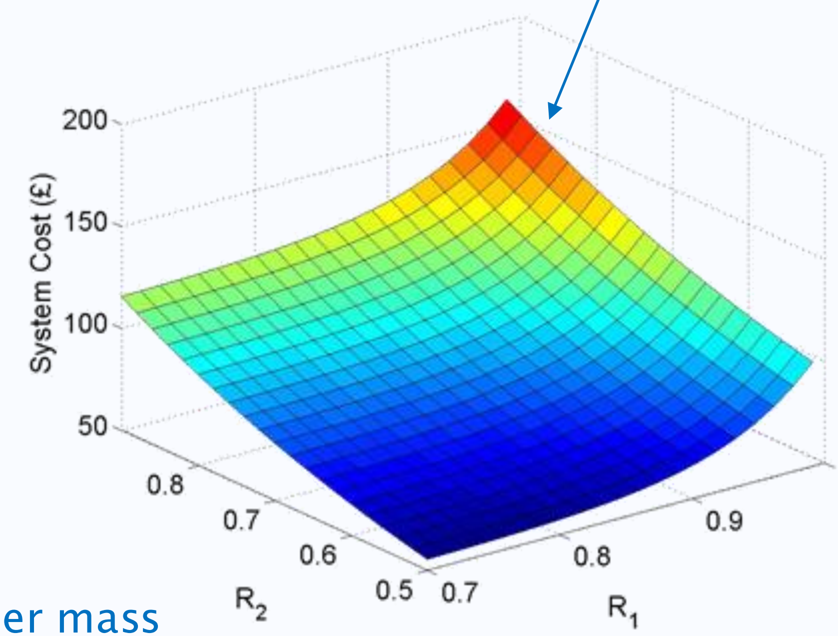
Reliability Optimisation Example

- Similarly we can construct plots for the system mass and cost

Motor is a mechanical system, we assume that mass is more sensitive to reliability

Generally more reliable systems are assumed to cost more

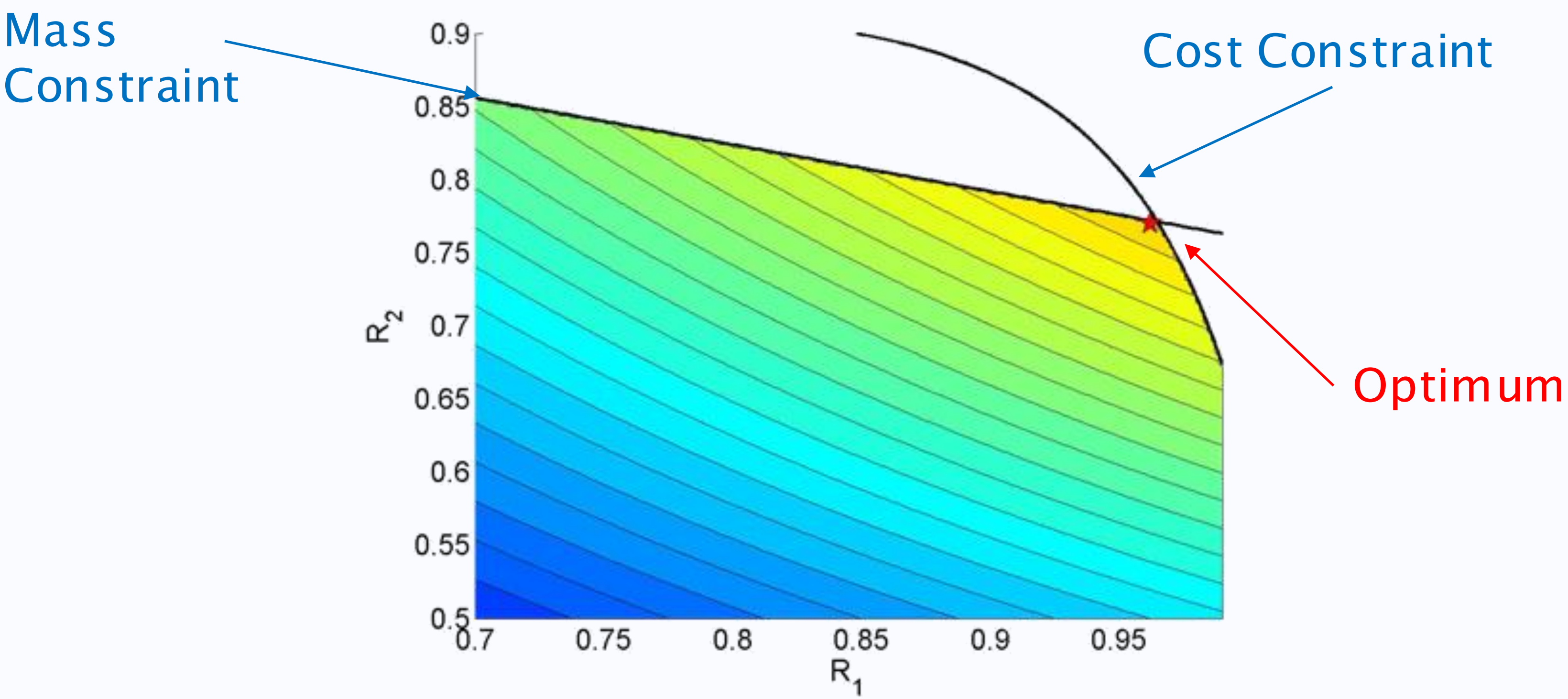




Speed controller mass increases little with increasing reliability

Reliability Optimisation Example

- How then do we solve this reliability optimisation problem?
  - Any of the DSO type methods e.g. evolutionary approaches, surrogate modelling, local optimisers etc.



University of
Southampton