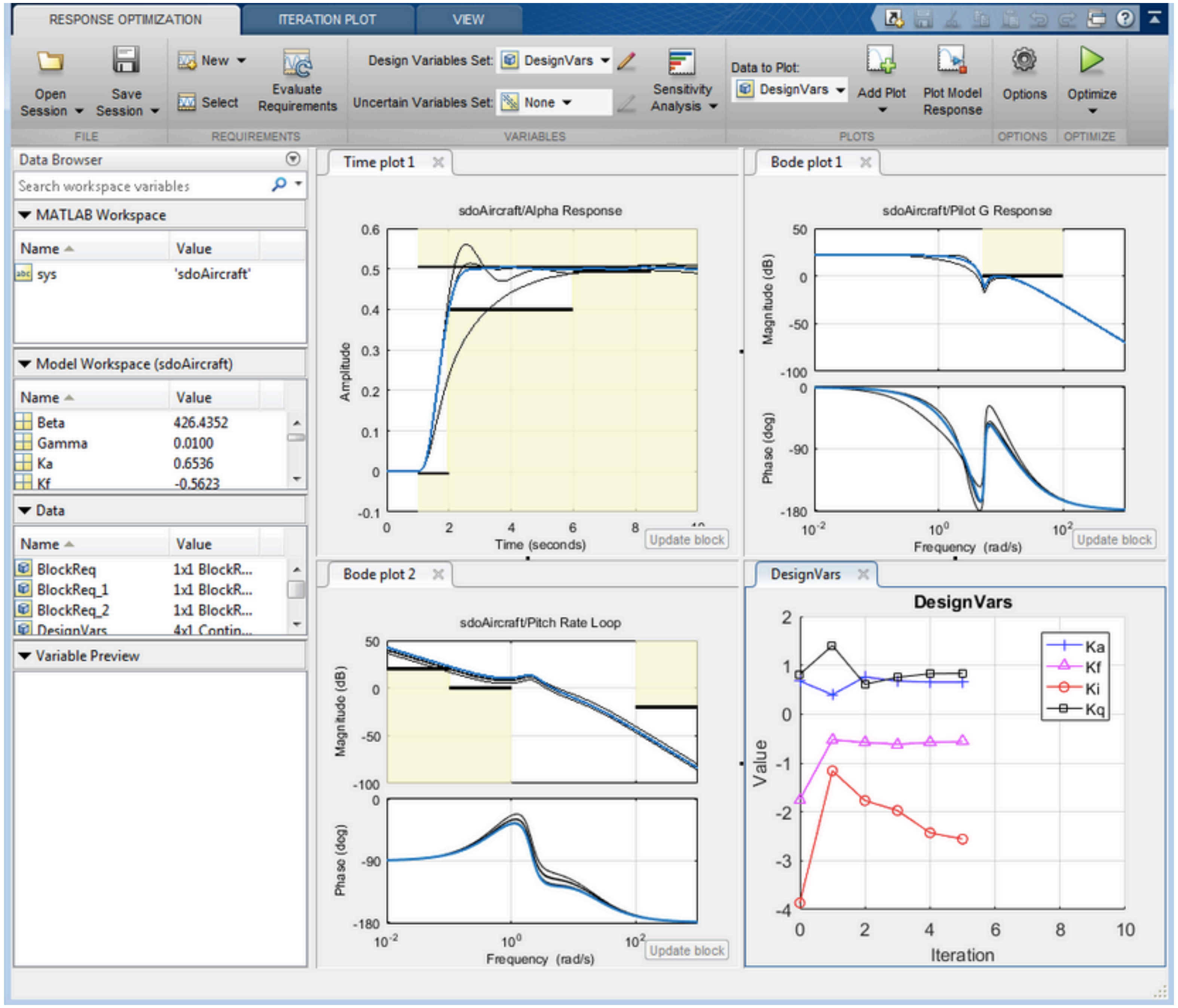
ME40064: System Modelling & Simulation
ME50344: Engineering Systems Simulation
Lecture 19

Dr Andrew Cookson
University of Bath, October 2018

LECTURE 19
Optimisation

- Understand basic theory of optimisation
- Appreciate how parameter space studies can be performed efficiently

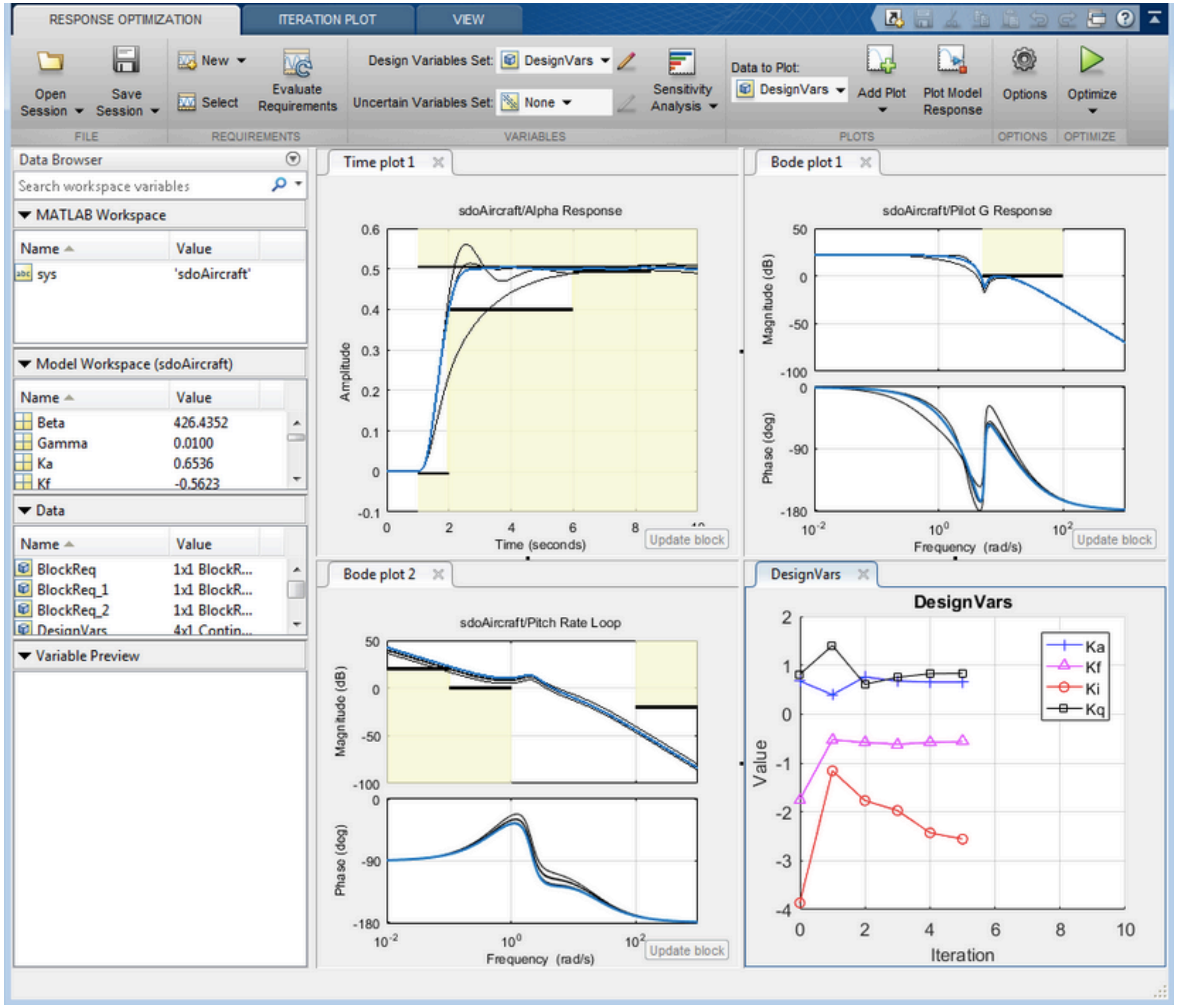
MOTIVATION
System Response Optimisation In Simulink



Design optimisation within Simulink

http://uk.mathworks.com/help/sldo/getting-started-with-simulink-design-optimization.html

MOTIVATION
System Response Optimisation In Simulink



Topic of Lecture 20 and Tutorial 10...

OPTIMISATION
An Overview

Optimisation is a key activity in modern engineering design
- often the basic concept is known but must be tailored to a specific problem case

It is the maximising/minimising of some performance metric by systematically varying the values of the input design variables

Computational methods are in widespread use for conducting this optimisation

Often in practical engineering problems it is a multi-objective optimisation problem
- a design must be made that is both light and stiff

Robust design optimisation an important sub-discipline
- aim for an optimal design that performs well under a range of conditions

OPTIMISATION
The Key Components

The Objective Function
This represents our design target, and is the quantity we wish to optimise for

Generally written as a minimisation problem i.e. minimise the difference between actual and target performance

Or, to maxmise $F(\mathbf{x})$, set problem as minimise $-F(\mathbf{x})$

Parameters
These are the design variables that we can alter in order to optimise the design

Constraints
These are parameter value limits that the optimal solution must satisfy
- physical dimensions
- monetary cost
- performance limits

A Simple Minimisation Problem

Consider the following quadratic objective function that we want to minimise:

$$F(x) = x^2 - 6x + 10$$

Can find the minimum point by finding where the gradient is equal to zero:

$$\frac{dF(x)}{dx} = 0$$

For this objective function, the gradient is:

$$\frac{dF(x)}{dx} = 2x - 6$$

The solution to this is simply:

$$\frac{dF(x)}{dx} = 0 \quad at \quad x = 3$$

OPTIMISATION
A Minimisation Problem

Consider the following quadratic objective function that we want to minimise:

$$F(x) = x^2 - 6x + 10$$

Can find the minimum point by finding where the gradient is equal to zero:

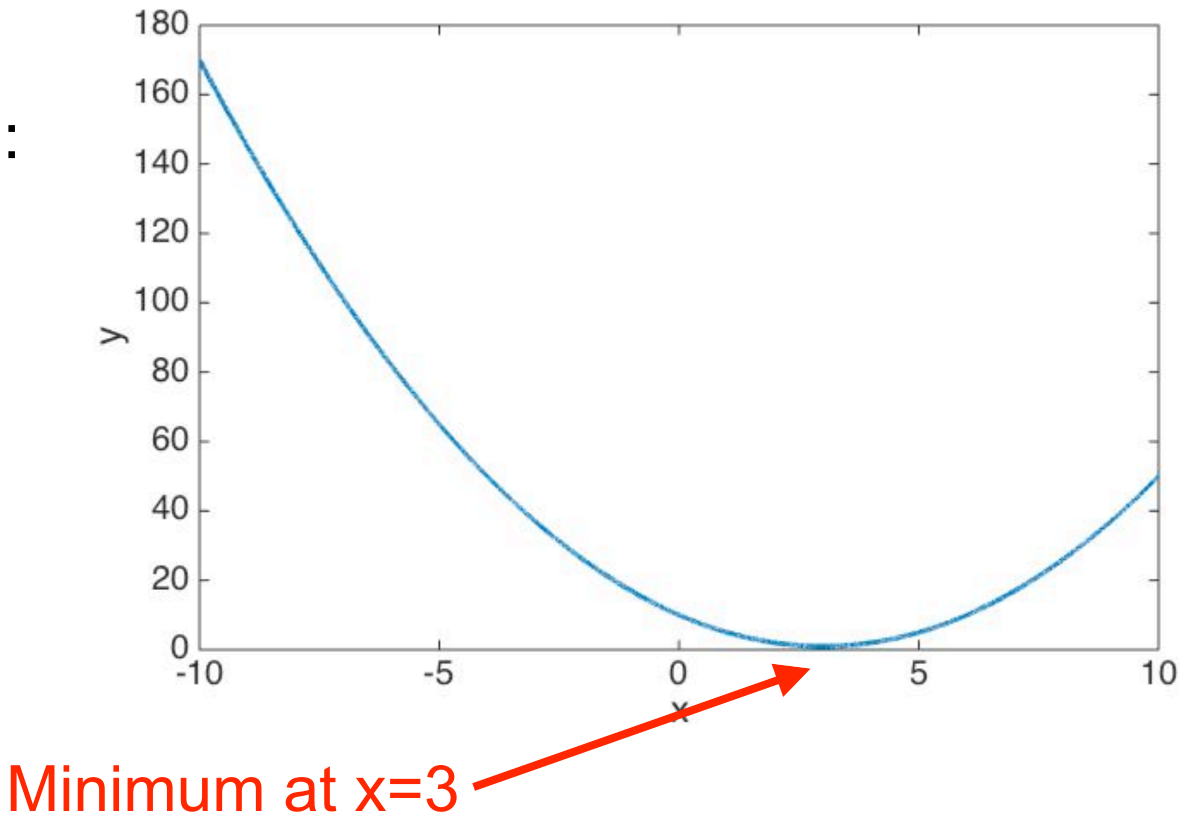
$$\frac{dF(x)}{dx} = 0$$

For this objective function, the gradient is:

$$\frac{dF(x)}{dx} = 2x - 6$$

The solution to this is simply:

$$\frac{dF(x)}{dx} = 0 \quad at \quad x = 3$$



OPTIMISATION
Some Strategies

For real problems the objective function is not so simple and the minimum point can not be found so easily

Objective function **not usually explicitly known** - i.e. it emerges as the result of a complex simulation

Many different optimisation algorithms exist for these problems, such as:

Genetic Algorithms
- objective function is taken to be the evolutionary fitness of a design
- design traits are evolved by analogy to genes, with mutation, cross-over and selection

Gradient-descent methods
- uses gradient information to derive the new guess value for the parameters

Newton's Method
- uses information about both the gradient and second derivative
- approaches optimum faster than gradient-descent methods

OPTIMISATION
Local Vs. Global Minima

Minima of objective function denoted $F^* = F(\mathbf{x}^*)$ for minimiser value $\mathbf{x}^*$

Vector variable

Local minima - smallest value of F within some small neighbourhood

$$F^* \leq F(\mathbf{x}) \quad \text{for all } |\mathbf{x} - \mathbf{x}^*| \leq \delta \text{ for some small region } \delta > 0$$

Global minimum - smallest value of F for all feasible values of $\mathbf{x}$

$$F^* \leq F(\mathbf{x}) \quad \text{for all } \mathbf{x} \text{ in the domain}$$

There can be many local minima that are not global minima

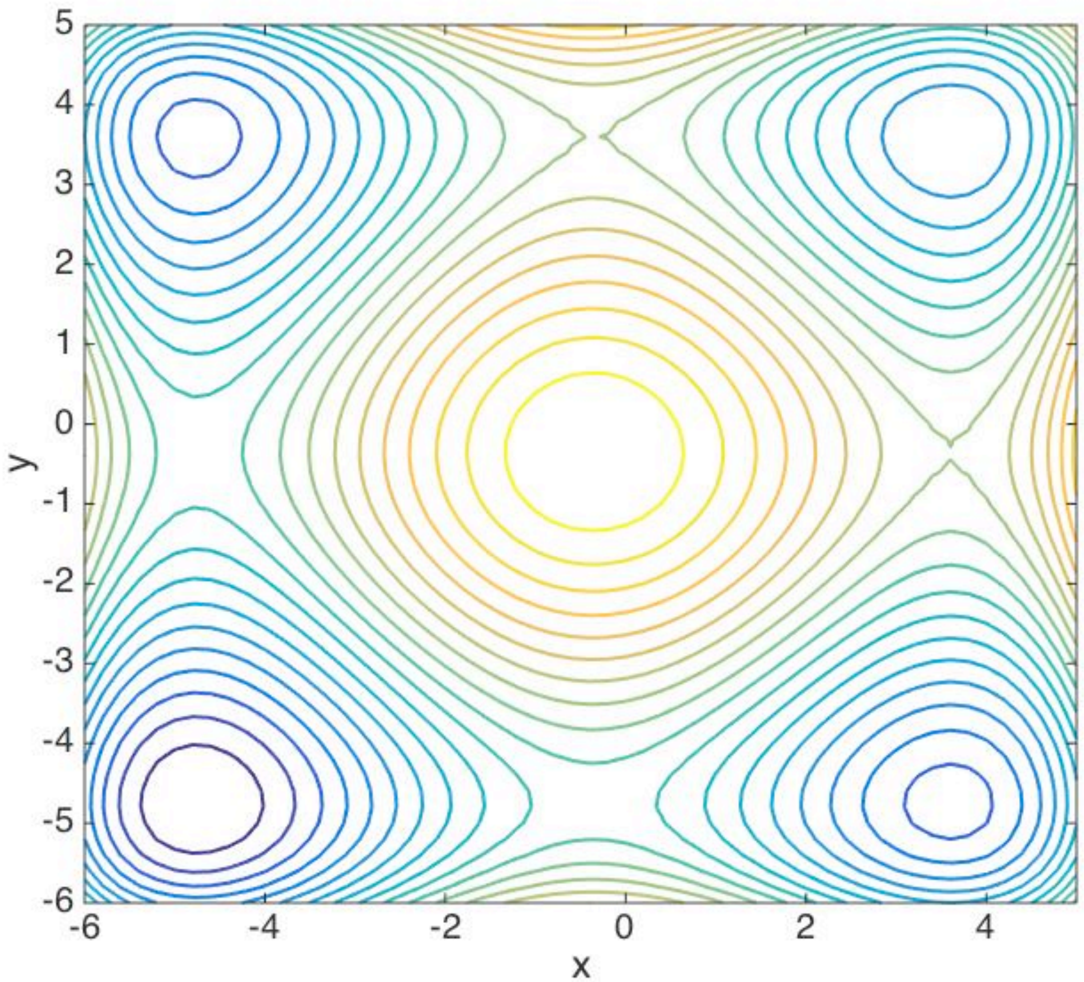
In general, global optimisation is a “hard” problem computationally

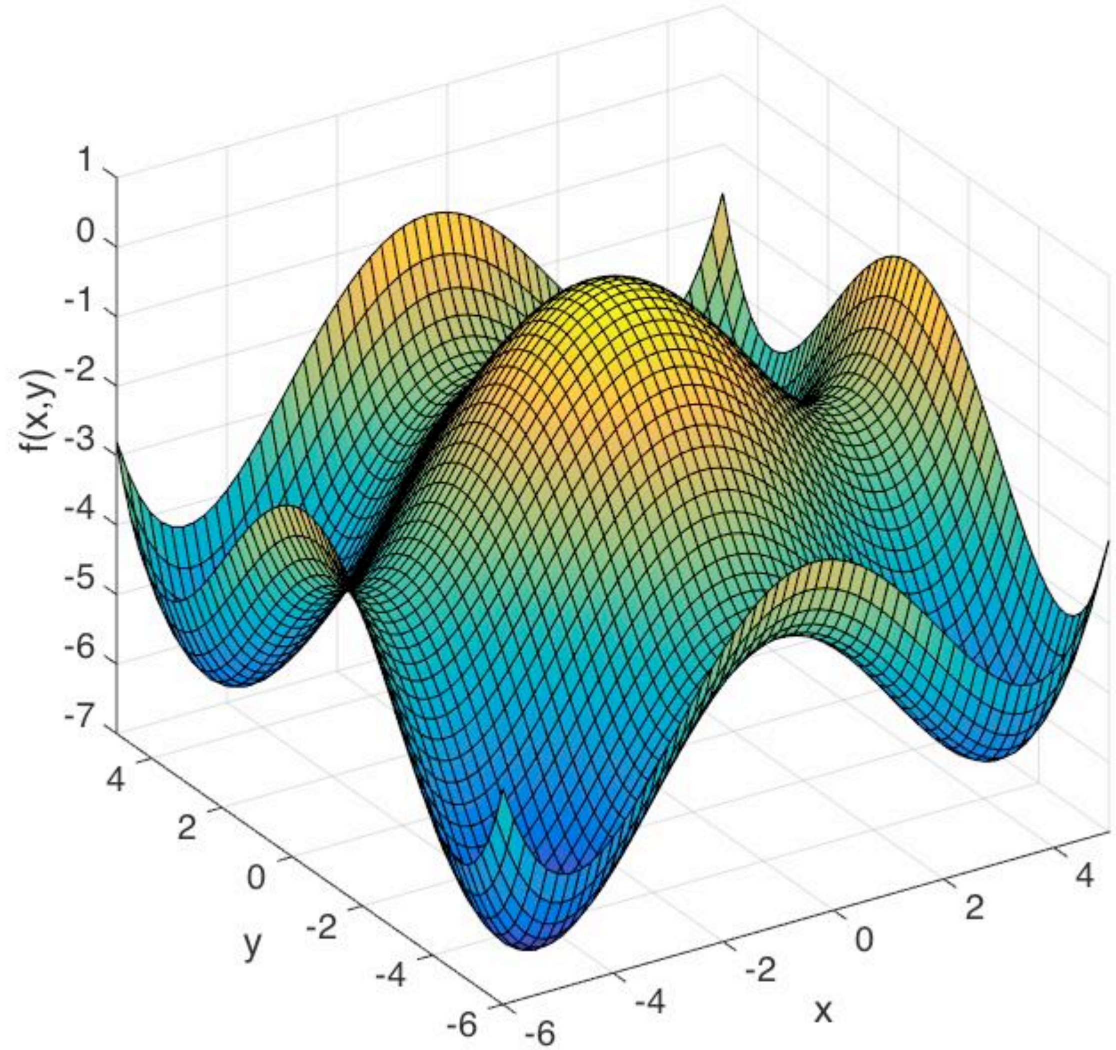
Local Vs. Global Minima Example

Consider the following objective function:

$$f(x,y) = [(x+0.5)^4 - 35x^2 - 25x + (y+0.5)^4 - 35y^2 - 25y]/100$$

Plotted as a surface & contour plot:





Multiple local minima for this objective function

Gradient Descent: The Theory

Multiple algorithms implemented within Simulink - a commonly used one uses a **gradient descent** based method

Consider an objective function: $F(\mathbf{x})$

We assume that this function is differentiable and want to minimise it for $\mathbf{x}$

This method can be understand by analogy to finding the shortest path down a hill
- want to walk in the direction of the steepest gradient
- after walking some distance in that direction, the slope may have changed
- therefore recalculate the new gradient and continue walking in the new direction
- repeating this process until the bottom of the hill is reached i.e. the gradient becomes zero
- trade-off between cost of gradient calculation with accuracy of smaller step size

OPTIMISATION
The Theory

This method can be expressed mathematically in the following way:

1. Initial starting point is defined: $\mathbf{x_0}$

2. For iteration, *n*, compute the gradient of the objective function: $\nabla F(\mathbf{x_n})$

3. Compute the step to be taken, $\alpha_n \nabla F(\mathbf{x_n})$, such that the value of F will decrease: $F(\mathbf{x_n}) \geq F(\mathbf{x_{n+1}})$

4. Compute the new ***x*** value based on this step:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - \alpha_n \nabla F(\mathbf{x_n}), n \geq 0$$

5. Note the minus sign, as we want to move <u>down</u> the gradient $\nabla F(\mathbf{x_n})$

GRADIENT DESCENT
Demonstration

Use gradient descent to find minimum of the objective function:

$$F(x,y) = x^2 + xy + 3y^2$$

Gradient of this objective function is:

$$\nabla F(x,y) = (2x + y, x + 6y)$$

Test performance of the method for varying α

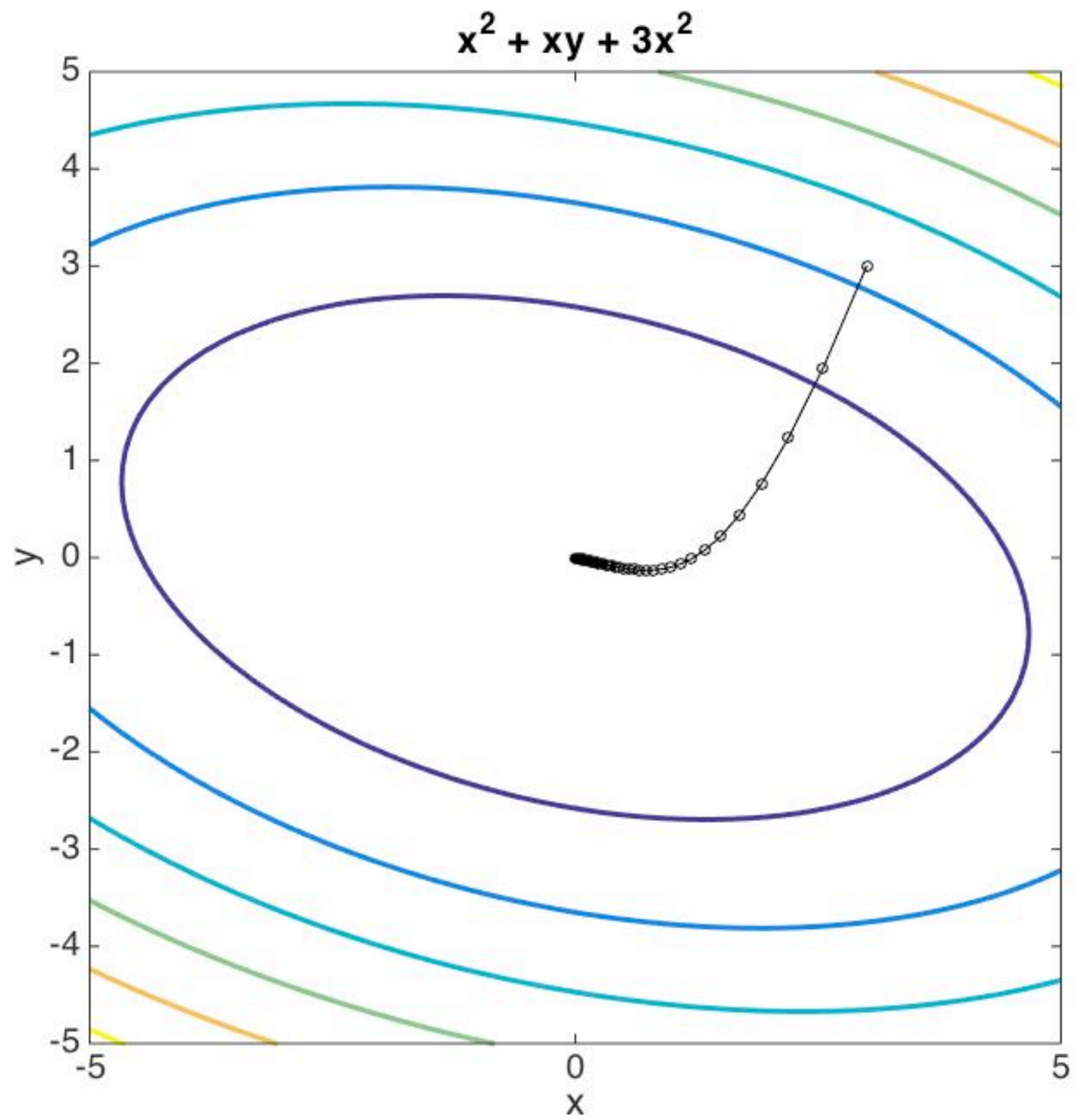
GRADIENT DESCENT
Demonstration

Use gradient descent to find minimum of the objective function:

$$F(x, y) = x^2 + xy + 3y^2$$

Setting $\alpha = 0.05$

Number of iterations = **133**



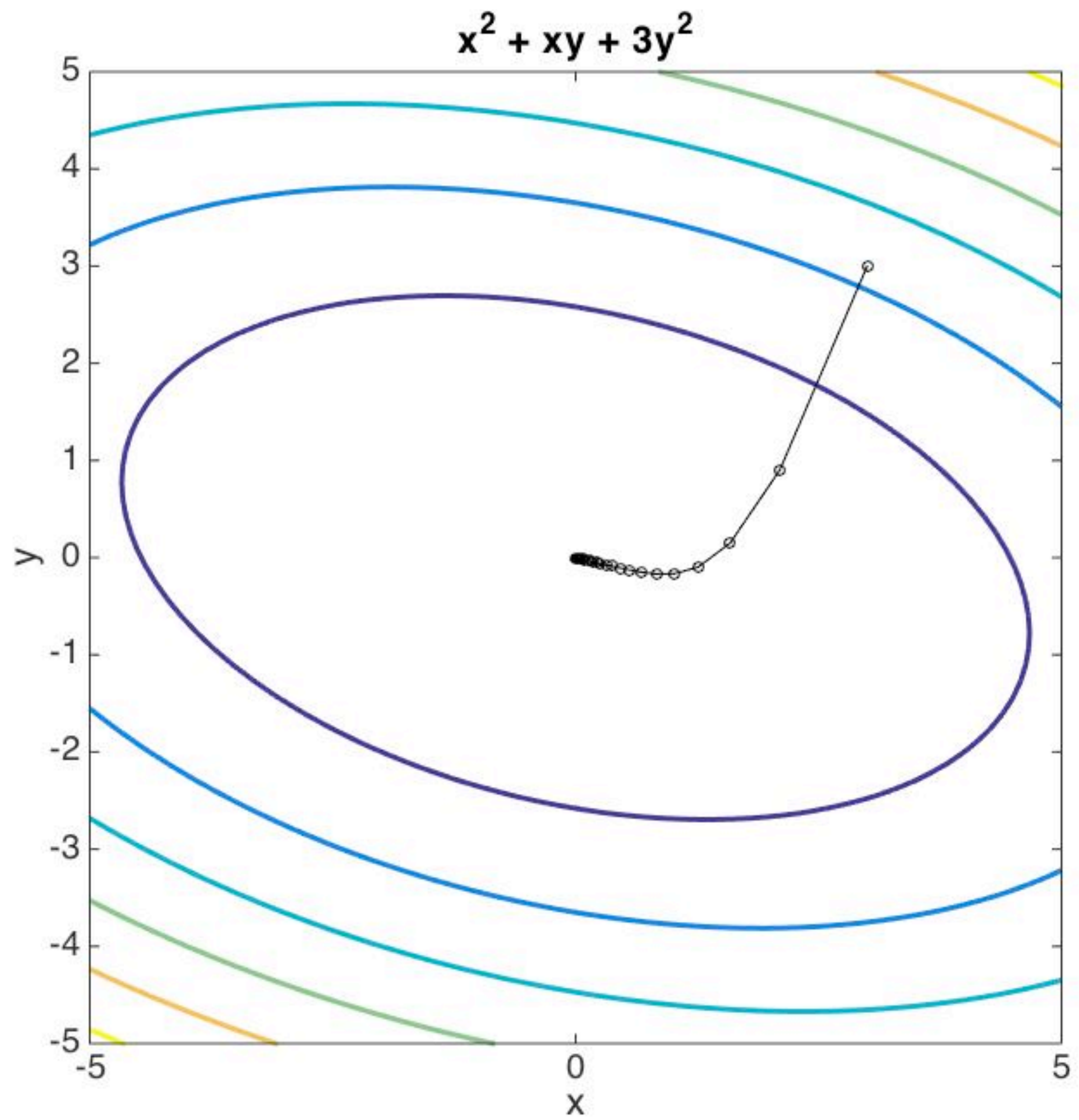
GRADIENT DESCENT
Demonstration

Use gradient descent to find minimum of the objective function:

$$F(x,y) = x^2 + xy + 3y^2$$

Setting $\alpha = 0.1$

Number of iterations = **67**



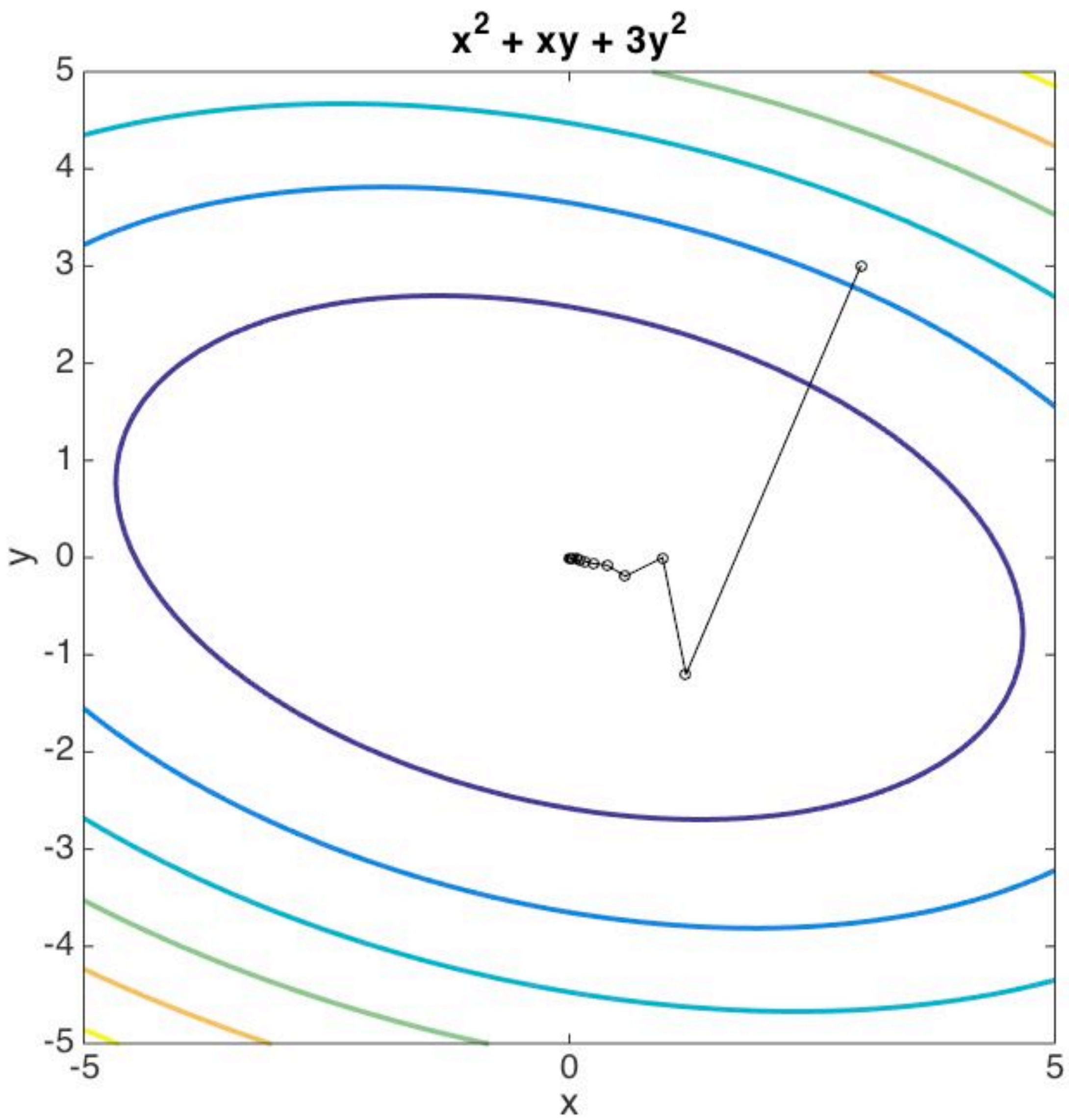
GRADIENT DESCENT
Demonstration

Use gradient descent to find minimum of the objective function:

$$F(x, y) = x^2 + xy + 3y^2$$

Setting $\alpha = 0.2$

Number of iterations = **32**



GRADIENT DESCENT
Demonstration

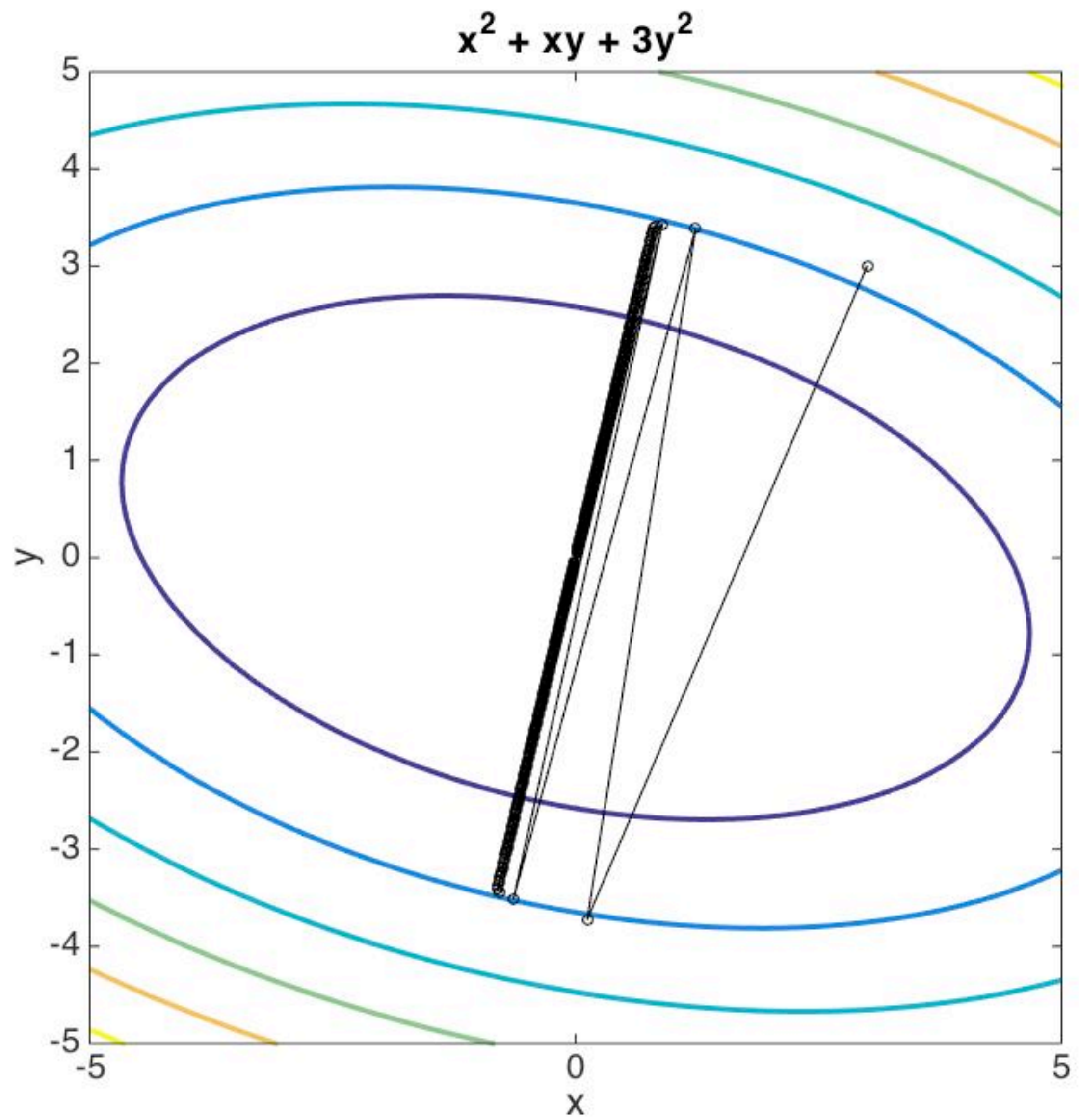
Use gradient descent to find minimum of the objective function:

$$F(x, y) = x^2 + xy + 3y^2$$

Setting $\alpha = 0.32$

Number of iterations = **1000**

i.e. maxed out
iteration counter
without converging



OPTIMISATION
Practical Points

Relatively simple method to understand and code

Computing each step is relatively inexpensive

However may need more steps to converge compared to other methods

In some problems the objective function is very shallow
- the method then takes steps that are too big even for small α_n
- each iteration then zig-zags around the solution space
- as a result convergence is very slow

Optimisation algorithms within Simulink

https://uk.mathworks.com/help/sldo/ug/how-the-optimization-algorithm-formulates-minimization-problems.html

GRADIENT DESCENT IN MATLAB
Fmincon - A Versatile Function

Within MATLAB gradient descent is implemented as part of the function: `fmincon`

Objective function: $F(\mathbf{x})$

Upper/lower bounds: $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$

Linear constraints:

$A\mathbf{x} = b$ Equality constraints

$A\mathbf{x} \leq b$ Inequality constraints

Non-linear constraints:

$c(\mathbf{x}) = 0$ Equality constraints

$c(\mathbf{x}) \leq 0$ Inequality constraints

For full information see: https://uk.mathworks.com/help/optim/ug/fmincon.html

GRADIENT DESCENT IN MATLAB
Fmincon - A Versatile Function

Within MATLAB gradient descent is implemented as part of the function: `fmincon`

Objective function:	$F(\mathbf{x})$		`fun`
Upper/lower bounds:	$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$		`lb,ub`
Linear constraints:	$A\mathbf{x} = b$	Equality constraints	`A,b`
	$A\mathbf{x} \leq b$	Inequality constraints	`Aeq,beq`
Non-linear constraints:			
	$c(\mathbf{x}) = 0$	Equality constraints	`nonlcon`
	$c(\mathbf{x}) \leq 0$	Inequality constraints	

```
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
```

For full information see: https://uk.mathworks.com/help/optim/ug/fmincon.html

PARAMETER SPACE STUDY
A Complementary Activity

Parameter space studies are a complementary activity to optimisation

Optimisation processes may provide the optimal design to the problem but not necessarily any understanding of what is important

Parameter space study can provide that intuitive understanding of the system's behaviour

Optimisation algorithms will (usually) work well if there is a single global optimal solution

Understanding the behaviour of the objective function will help reveal if there are multiple local minima/optimal point
- can inform the optimisation process such that a suitable algorithm can be chosen
- helps improve confidence in optimisation process

PARAMETER SPACE STUDY
The Problem

For systems with one parameter simply vary that parameter and observe the effect

For systems with two (or more) parameters could perform a **one-factor-at-a-time** strategy
- vary one parameter, while keeping all others at the baseline value
- doesn't capture any interactions that arise from multiple parameters varying together

PARAMETER SPACE STUDY
The Problem

For systems with one parameter simply vary that parameter and observe the effect

For systems with two (or more) parameters could perform a **one-factor-at-a-time** strategy
- vary one parameter, while keeping all others at the baseline value
- doesn't capture any interactions that arise from multiple parameters varying together

Alternatively, for systems with two (or more) parameters could perform a **factorial** strategy
- test all combinations of the parameters for a range of values

However for systems with many parameters (particularly given limited experimental samples or computationally expensive models) cannot simply test every single parameter combination - use a **fractional factorial strategy** instead

PARAMETER SPACE STUDY
Design Of Experiments

Q. How do we pick a sensible subset of samples for testing?

PARAMETER SPACE STUDY
Design Of Experiments

Q. How do we pick a sensible subset of samples for testing?

A. Design of experiments

i.e. the design of a set of tests that will reveal the variation of a system’s behaviour under a range of parameter inputs

PARAMETER SPACE STUDY
Design Of Experiments

Q. How do we pick a sensible subset of samples for testing?

A. Design of experiments

i.e. the design of a set of tests that will reveal the variation of a system's behaviour under a range of parameter inputs

Experimental design involves:
- appropriate sampling of the parameter space
- consider any correlations between the parameters
- consider the likely variation of other factors (e.g. those outside direct control, such as operating conditions or natural variation)
- examine the statistical correlations between the input parameters and the output variables

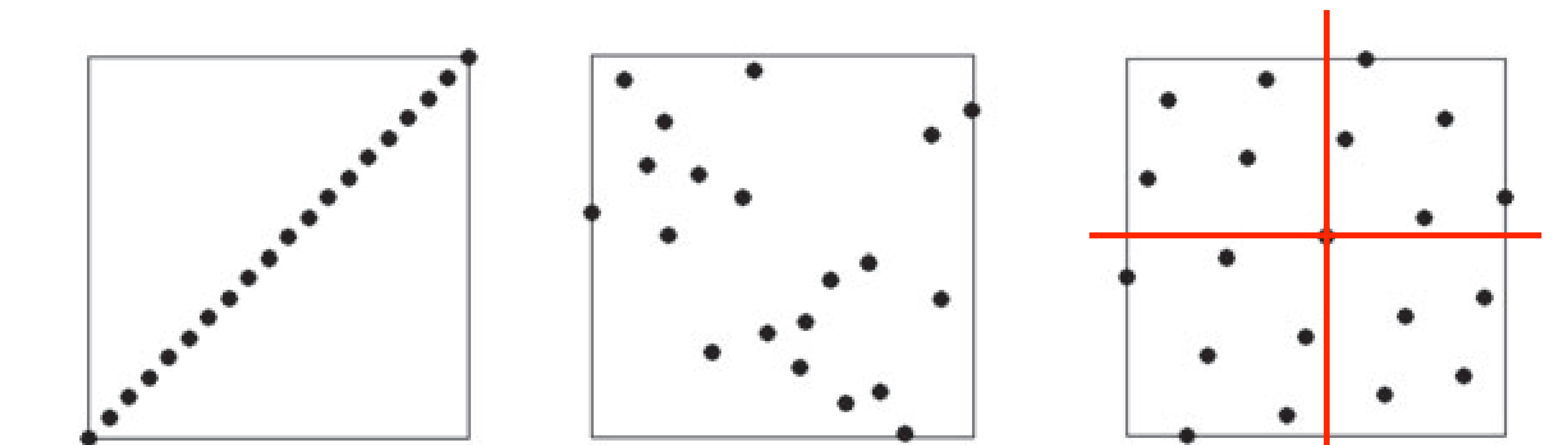
To illustrate the idea we will consider one method called Latin squares (though other sampling designs are available)

PARAMETER SPACE STUDY
Latin Squares

Given a two vectors of parameter values $\mathbf{x_1}, \mathbf{x_2}$, Latin squares are a set of sample points that don't duplicate a sample value in any direction:

$$\mathbf{x_1} = [x_1^{(1)}, x_1^{2)}, x_1^{(3)}, ..., x_1^{(n)}], \quad \mathbf{x_2} = [x_2^{(1)}, x_2^{2)}, x_2^{(3)}, ..., x_2^{(n)}]$$

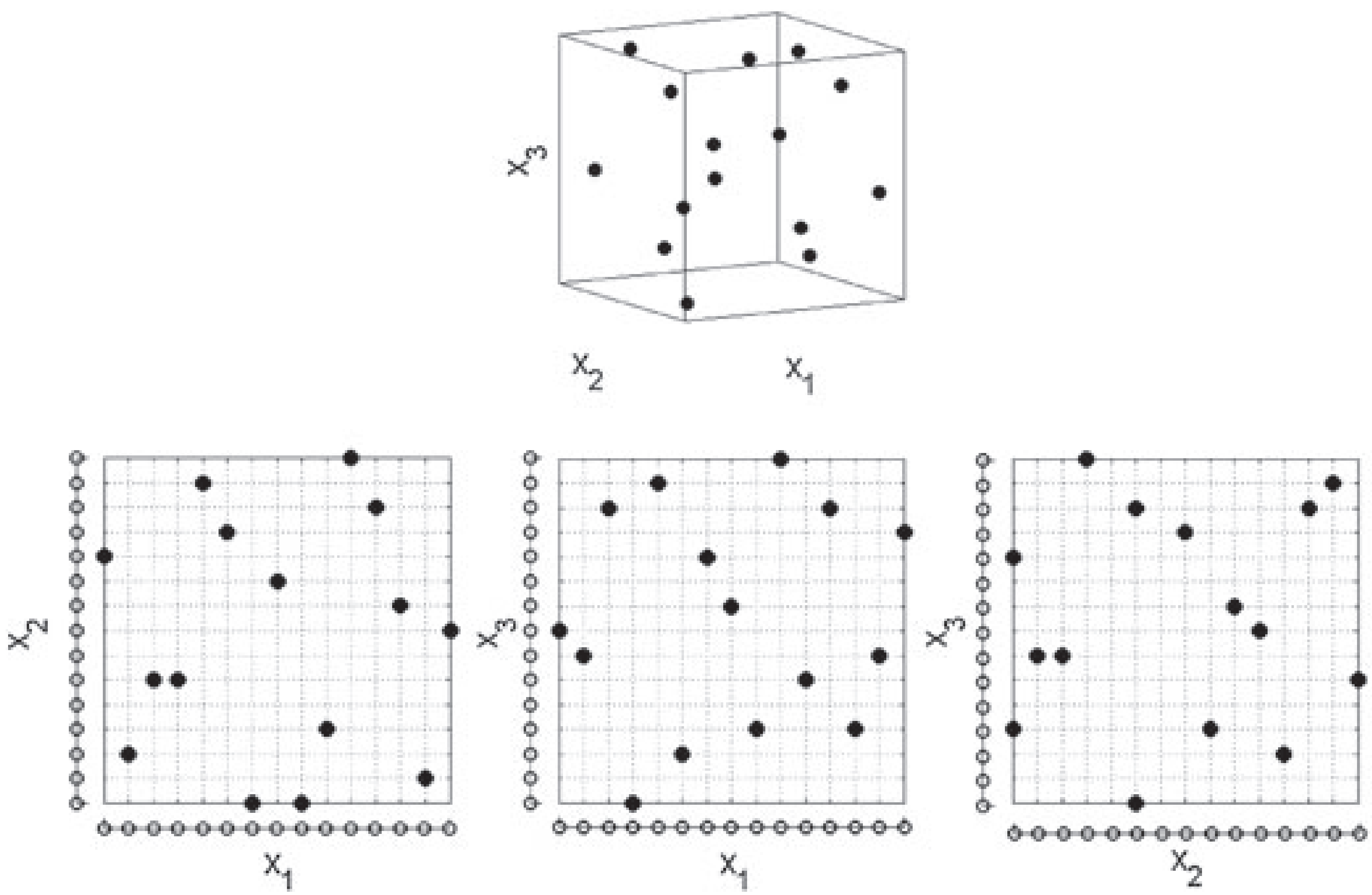
You can think of this like a Sudoku puzzle, where each row must contain only one instance of each number:



(a) Design with very poor space filling properties. (b) Randomized design. (c) Design with good space filling properties.

Qual. Reliab. Engng. Int. 2016, 32 1975–1985

This can be extended to three parameters $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$, as a cube:



(a) Latin hypercube design with 15 points.

Qual. Reliab. Engng. Int. 2016, 32 1975–1985

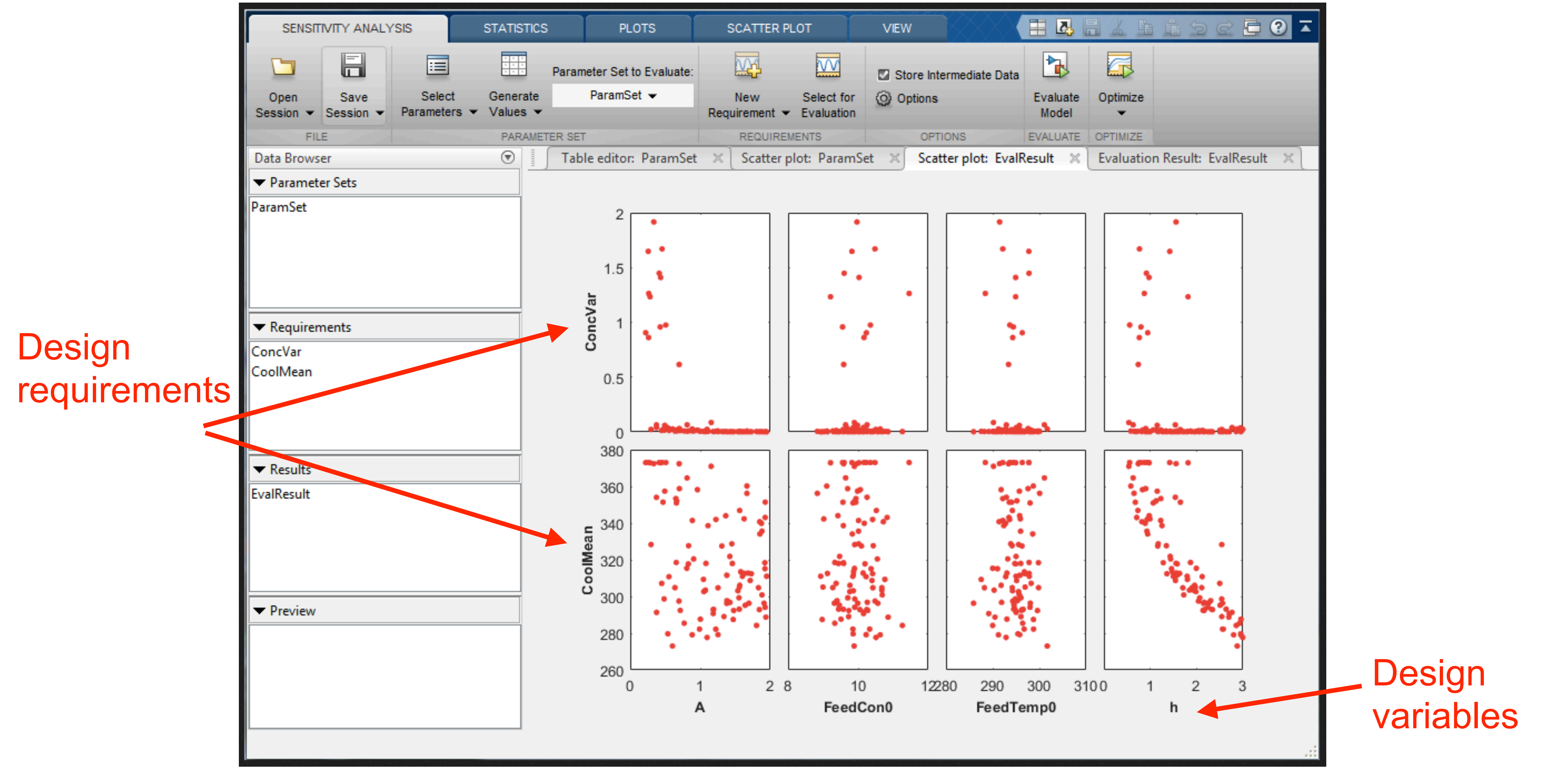
PARAMETER SPACE STUDY
Latin Hypercubes

For 4 or more parameters a hypercube (i.e. a higher dimensional cube is formed)

Grande Arche de la Défense, Paris (Image courtesy of Coldcreation)

PARAMETER SPACE STUDY
Analysing The Results

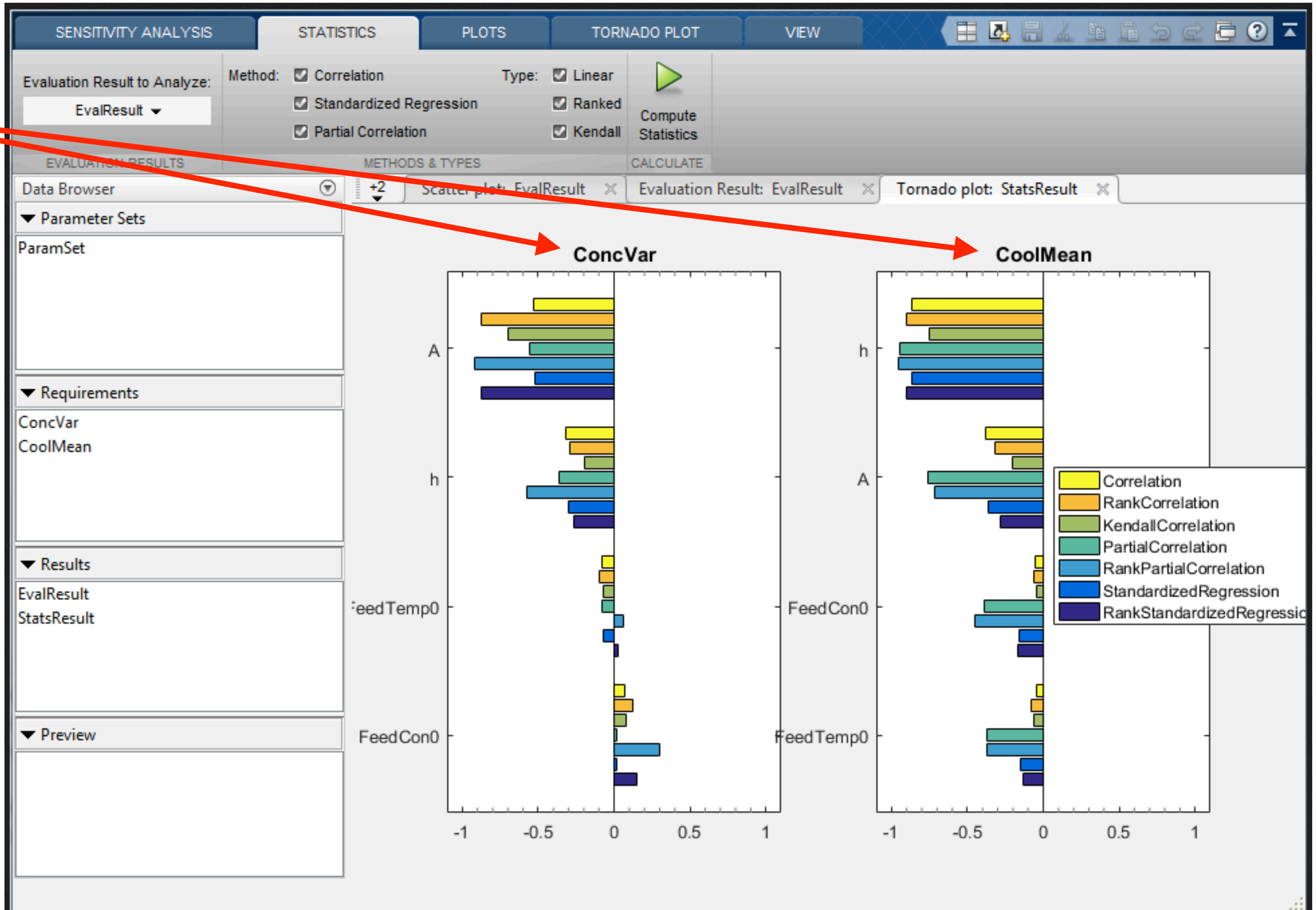
Having run the model for this parameter space sample, need to display and analyse the data. Scatter plots of design requirement vs design parameters reveal useful info



PARAMETER SPACE STUDY
Analysing The Results

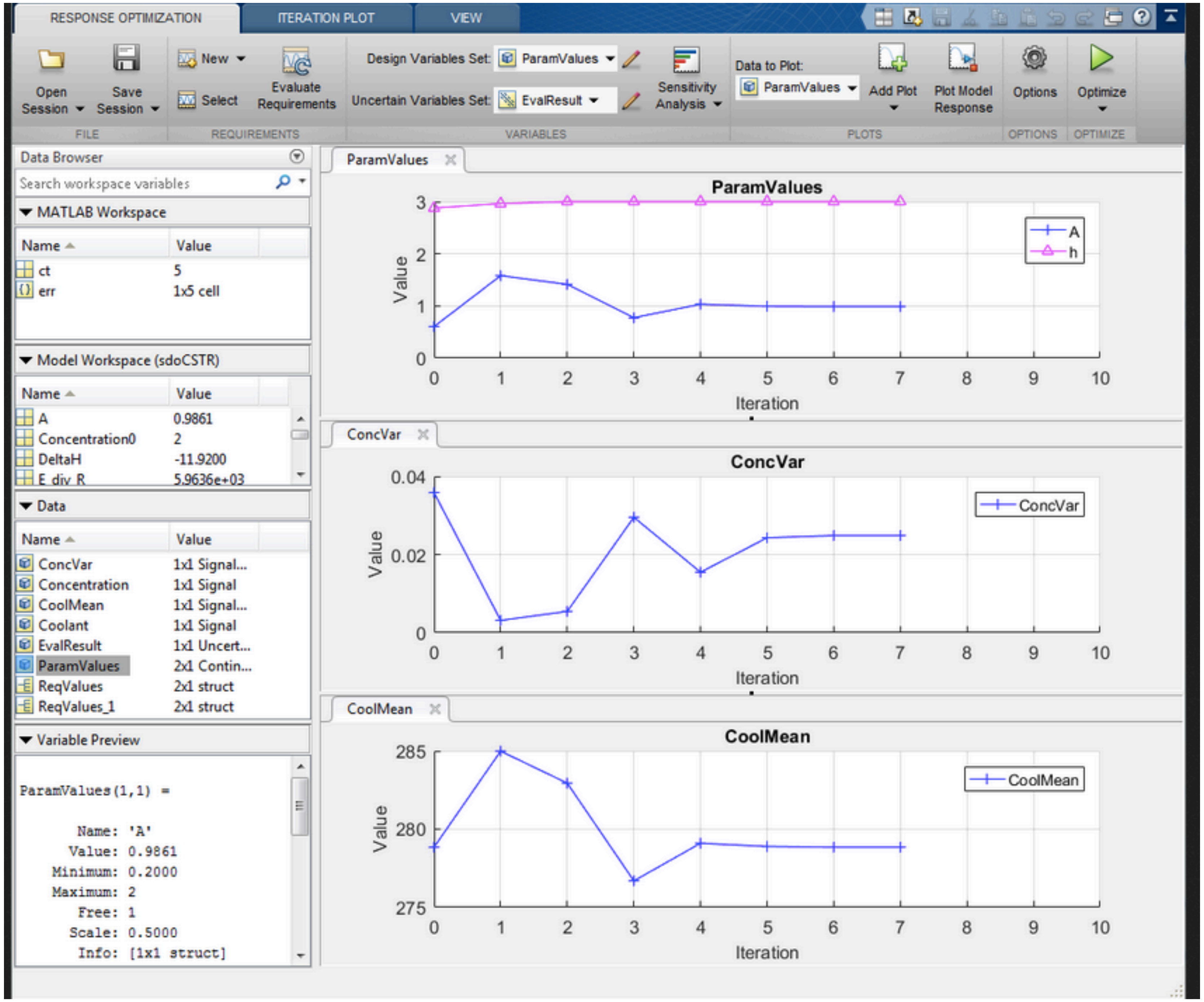
Statistical correlations can be derived for design requirement and design variables:

Design requirements



PARAMETER SPACE STUDY
Coupling With An Optimisation Process

This information can be fed into an optimisation process to provide a good initial guess:



PARAMETER SPACE STUDY
Further Reading For Interest

Sampling parameter spaces in Simulink

http://uk.mathworks.com/help/sldo/ug/sampling-parameters-and-states.html

Design exploration using parameter sampling in Simulink

http://uk.mathworks.com/help/sldo/ug/design-exploration-using-parameter-sampling-gui.html