Multidisciplinary System Design Optimization (MSDO)

Problem Formulation

Lecture 2

Anas Alfaris

Today's Topics

- MDO definition
- Optimization problem formulation
- MDO in the design process
- MDO challenges

MDO Definition

What is MDO ?

- A methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena
- Optimal design of complex engineering systems which requires analysis that accounts for interactions amongst the disciplines (= parts of the system)
- “How to decide what to change, and to what extent to change it, when everything influences everything else.”

Ref: AIAA MDO website http://www.aiaa.org (Click Inside AIAA, Technical Committees)

Engineering Design Disciplines

<u>Aircraft:</u>
Aerodynamics
Propulsion
Structures
Controls
Avionics/Software
Manufacturing
others

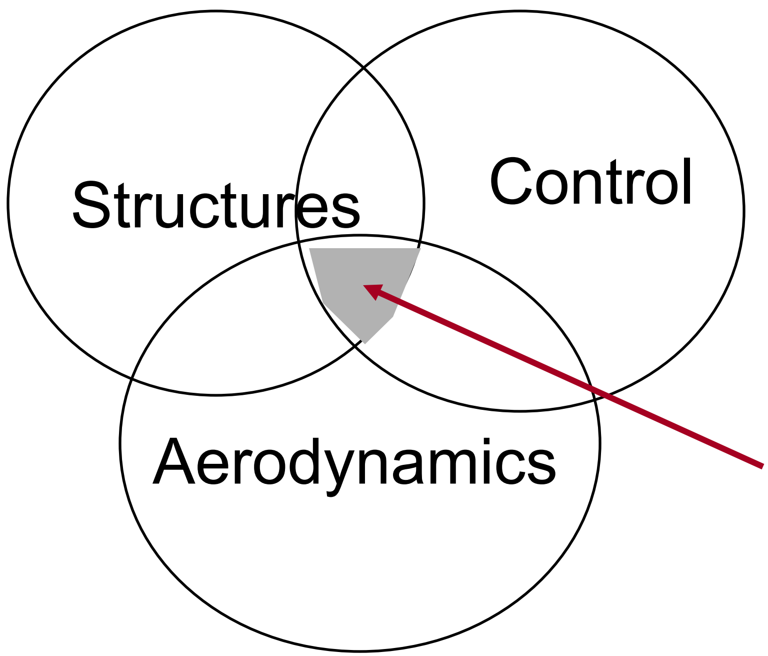
<u>Spacecraft:</u>
Astrodynamics
Thermodynamics
Communications
Payload & Sensor
Structures
Optics
Guidance & Control

<u>Automobiles:</u>
Engines
Body/chassis
Aerodynamics
Electronics
Hydraulics
Industrial design
others

⇨ Fairly mature, but advances in theory, methodology, computation and application foster substantial payoffs

Multidisciplinary Aspects of Design

Emphasis is on the multidisciplinary nature of the complex engineering systems design process. Aero-space vehicles are a particular class of such systems.



Emphasis in recent years has been on advances that can be achieved due to the inter-action of two or more disciplines.

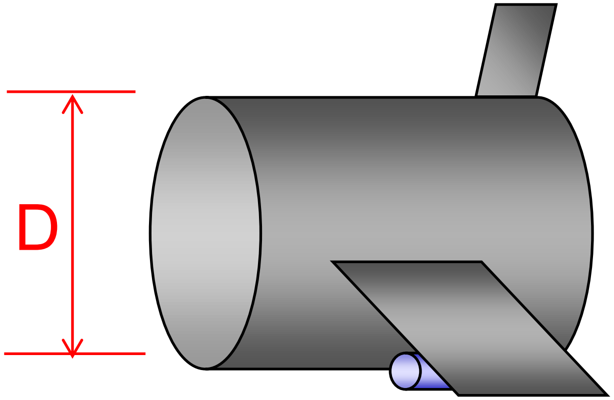
System Level Optimization

Why system-level, multidisciplinary optimization ?

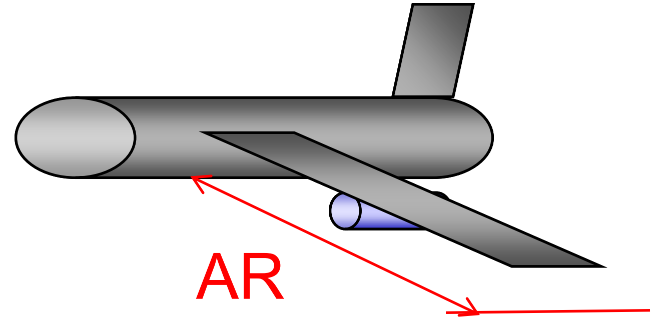
- Disciplinary specialists tend to strive towards improvement of objectives and satisfaction of constraints in terms of the variables of their own discipline

- In doing so they generate side effects - often unknowingly- that other disciplines have to absorb, usually to the detriment of the overall system performance

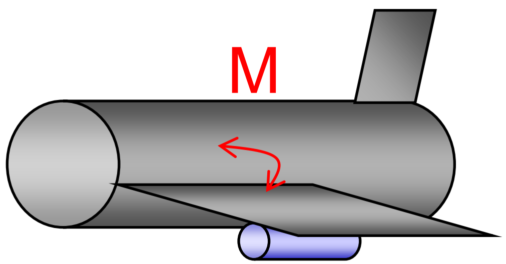
Aircraft “Optimization”



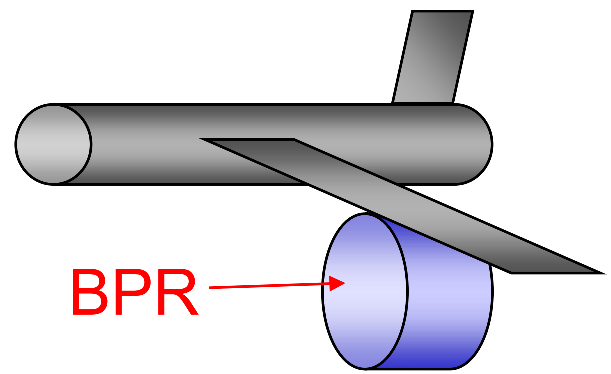
Marketing: maximize passenger volume
→ Cabin diameter



Aero: maximize L/D
→ Aspect Ratio



Structures: minimize structural mass
→ Wing-root moment



Propulsion: minimize specific fuel consumption (SFC)→ Bypass Ratio

System-level Optimization

Bréguet Range Equation

Marketing

Aero

All

$$R = \frac{V(L/D)}{g \cdot SFC} \cdot \ln \frac{W_{initial}}{W_{final}}$$

Propulsion

Structures

R = Range [m]
V = Flight velocity [m/s]
SFC = Specific Fuel Consumption [kg/s/N]
L/D = Lift-over-Drag ration [N/N]
g = gravitational acceleration [m/s^2]
$W_{initial}$ = Initial (takeoff) weight [N]
W_{final} = Weight at end of flight [N]
$W_{fuel}=W_{initial}-W_{final}$ Fuel quantity [N]

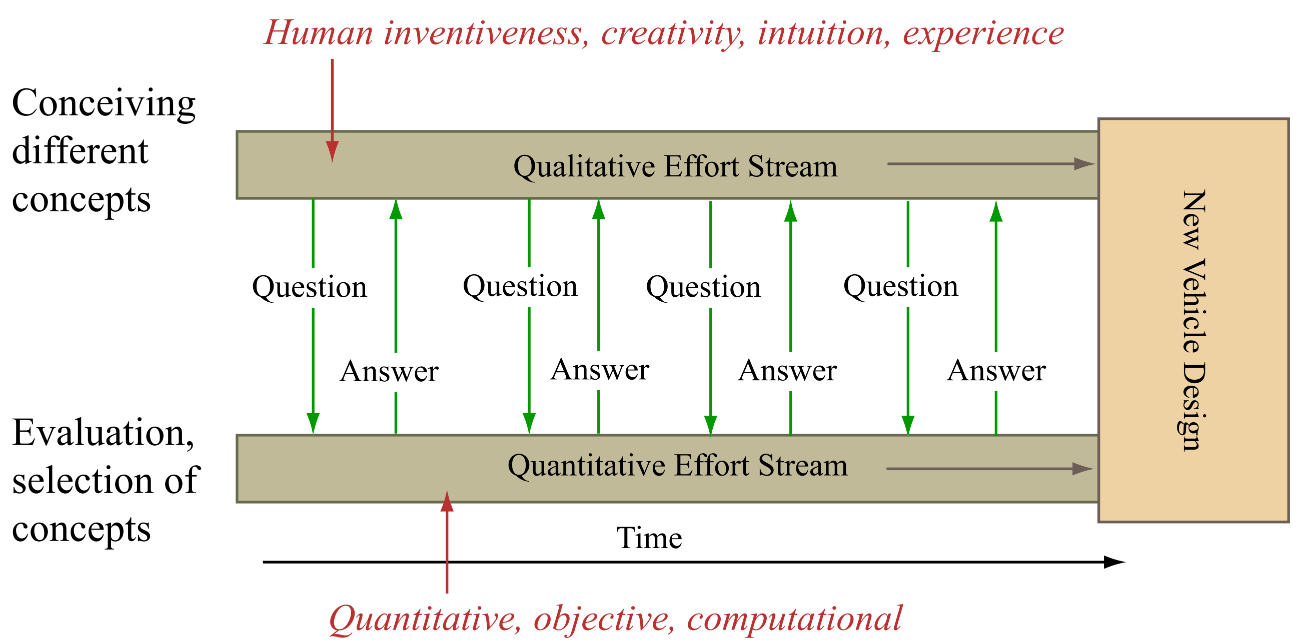
Human Interface Aspects of Design

It is wrong to think of MDO as “automated” or “push-button” design:

- The human strengths (creativity, intuition, decision-making) and computer strengths (memory, speed, objectivity) should complement each other
- The human will always be the Meta-designer
- Challenges of defining an effective interface – continuous vs. discrete thinking
- Challenges of visualization in multidimensional space, e.g. search path from initial design to final design

⇒ Human element is a key component in any successful system design methodology

Quantitative vs. Qualitative



Parallel, qualitative, and quantitative efforts in design.

Image by MIT OpenCourseWare.

Human mind is the driving force in the design process. MDO is a way of formalizing the quantitative tool to apply the best trade-offs.

Architecture vs. Design

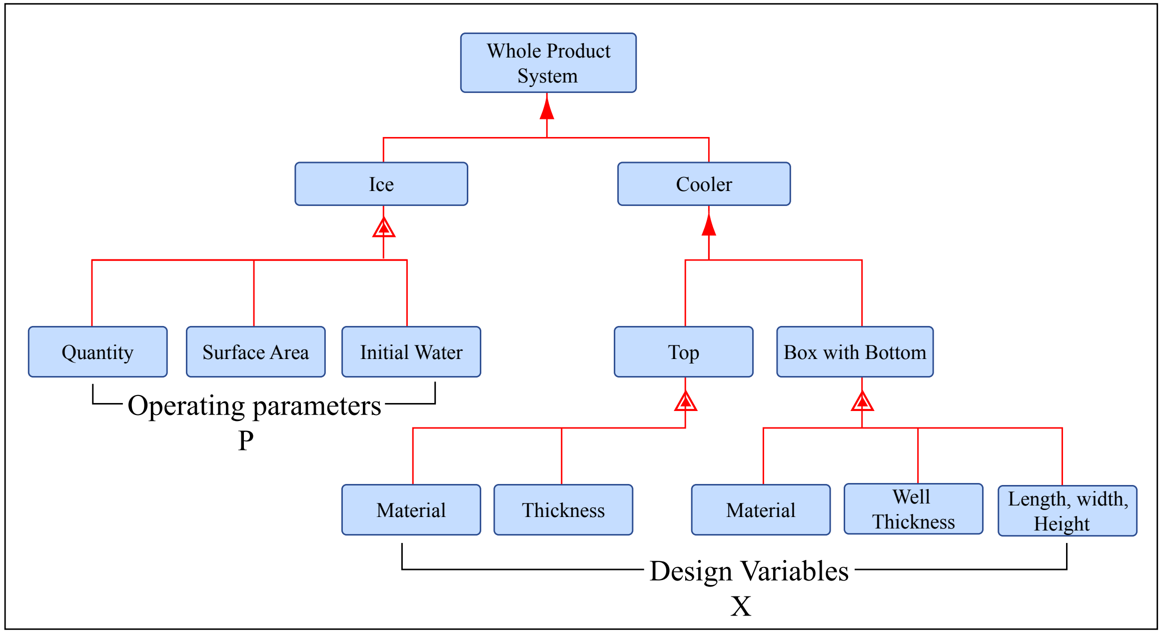


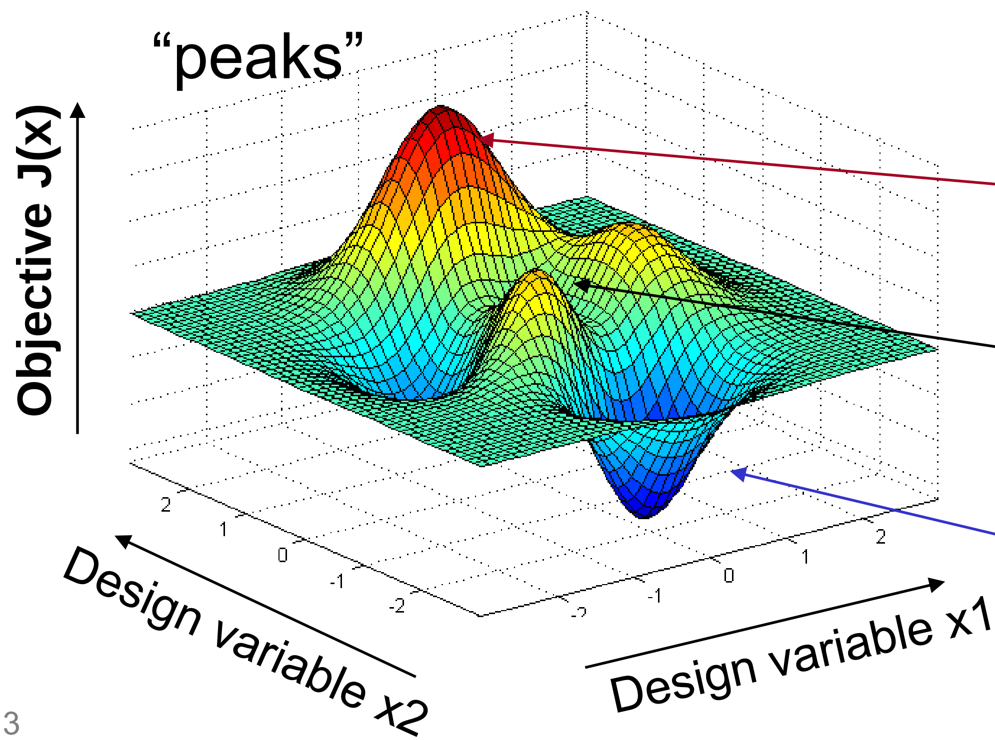
Image by MIT OpenCourseWare.

- Architecture selects the concept, decomposition and mapping of form to function
- Architecture establishes the vector of design and operating parameters
- Design selects the values of the vector of variables
- This is what optimization is good for
- Some work in "architecture" is just an exhaustive search over the design of one architecture

Optimization Problem Formulation

Optimization Aspects of Design

- Optimization methods have been combined with design synthesis and parametric analysis for ca. 40 years
- Traditionally used graphical methods to find maximum or minimum of a multivariate function ("carpet plot"), but….



Graphics break down above 3-4 dimensions

Where is max J(x) ?

Caution: local extrema !

Where is min J(x) ?

Combinatorial Explosion

- Any design can be defined by a vector in multidimensional space, where each design variable represents a different dimension
- For n > 3 a combinatorial "explosion" takes place and the design space cannot be computed and plotted in polynomial time
- Numerical optimization offers an alternative to the graphical approach and "brute force" evaluation

⇨ During past three decades much progress has been made in numerical optimization

Formal Notation

Quantitative side of the design problem may be formulated as a problem of Nonlinear Programming (NLP)

$$\min \mathbf{J}(\mathbf{x},\mathbf{p})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{p}) \le 0$$

$$\mathbf{h}(\mathbf{x},\mathbf{p}) = 0$$

$$x_{i,LB} \le x_i \le x_{i,UB} \quad (i = 1,\ldots,n)$$

This is the problem formulation that we will discuss this semester.

where

$$\mathbf{J} = \left[J_1(\mathbf{x}) \quad \cdots \quad J_z(\mathbf{x}) \right]^T$$

$$\mathbf{x} = \left[x_1 \quad \cdots \quad x_i \quad \cdots \quad x_n \right]^T$$

$$\mathbf{g} = \left[g_1(\mathbf{x}) \cdots g_{m_1}(\mathbf{x}) \right]^T$$

$$\mathbf{h} = \left[h_1(\mathbf{x}) \cdots h_{m_2}(\mathbf{x}) \right]^T$$

Objectives

The objective can be a vector **J** of *z* system responses or characteristics we are trying to maximize or minimize

$$\mathbf{J} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_i \\ \vdots \\ J_z \end{bmatrix} = \begin{bmatrix} \text{cost} \quad [\$] \\ \text{range} \quad [\text{km}] \\ \text{weight} \; [\text{kg}] \\ \text{data rate} \; [\text{bps}] \\ \vdots \\ \text{ROI} \quad [\%] \end{bmatrix}$$

Often the objective is a scalar function, but for real systems often we attempt multi-objective optimization:

$$\mathbf{x} \mapsto \mathbf{J}(\mathbf{x})$$

Some objectives can be conflicting.

Design Variables

16.888
ESD.77

Design vector **x** contains *n* variables that form the design space

During design space exploration or optimization we change the entries of **x** in some rational fashion to achieve a desired effect

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \text{aspect ratio [-]} \\ \text{transmit power [W]} \\ \text{\# of apertures [-]} \\ \text{orbital altitude [km]} \\ \vdots \\ \text{control gain [V/V]} \end{bmatrix}$$

x_i can be .....

Real: $x_i \in \mathcal{R}$

Integer: $x_i \in \mathcal{I}$

Binary: $x_i \in \{0, 1\}$

Boolean: $x_i \in \{\text{true}, \text{false}\}$

⇒ Design variables are "controlled" by the designers

Parameters

Parameters *p* are quantities that affect the objective **J**, but are considered fixed, i.e. they cannot be changed by the designers.

Sometimes parameters *p* can be turned into design variables x_i to enlarge the design space.

Sometimes parameters *p* are former design variables that were fixed at some value because they were found not to affect any of the objectives J_i or because their optimal level was predetermined.

Constraints

Constraints act as boundaries of the design space **x** and typically occur due to finiteness of resources or technological limitations of some design variables.

Often, but not always, optimal designs lie at the intersection of several active constraints

Inequality constraints: $g_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \ldots, m_1$

Equality constraints: $h_k(\mathbf{x}) = 0 \quad k = 1, 2, \ldots, m_2$

Bounds: $x_{i,LB} \leq x_i \leq x_{i,UB} \quad i = 1, 2, \ldots, n$

Objectives are what we are trying to achieve
Constraints are what we cannot violate
Design variables are what we can change

MIT esd

Constraints versus Objectives

It can be difficult to choose whether a condition is a constraint or an objective.

For example: should we try to minimize cost, or should we set a constraint stating that cost should not exceed a given level.

The two approaches can lead to different designs.

Sometimes, the initial formulation will need to be revised in order to fully understand the design space.

In some formulations, all constraints are treated as objectives (physical programming).

Example Problem Statement

design variables

objective function

Minimize the **take-off weight of the aircraft** by changing **wing geometric parameters** while satisfying the given **range and payload requirements** at the given **cruise speed**.

constraints

parameter

Group Exercise... (10 mins)

For your group's system:

1. Consider the preliminary design phase.
   Identify:
   - -important disciplines
   - -potential objective functions
   - -potential design variables
   - -system parameters
   - -constraints and bounds

2. Report out

MDO in the Design Process

What MDO really does

MDO mathematically traces a path in the design space from some initial design $\mathbf{x}_o$ towards improved designs (with respect to the objective **J**).

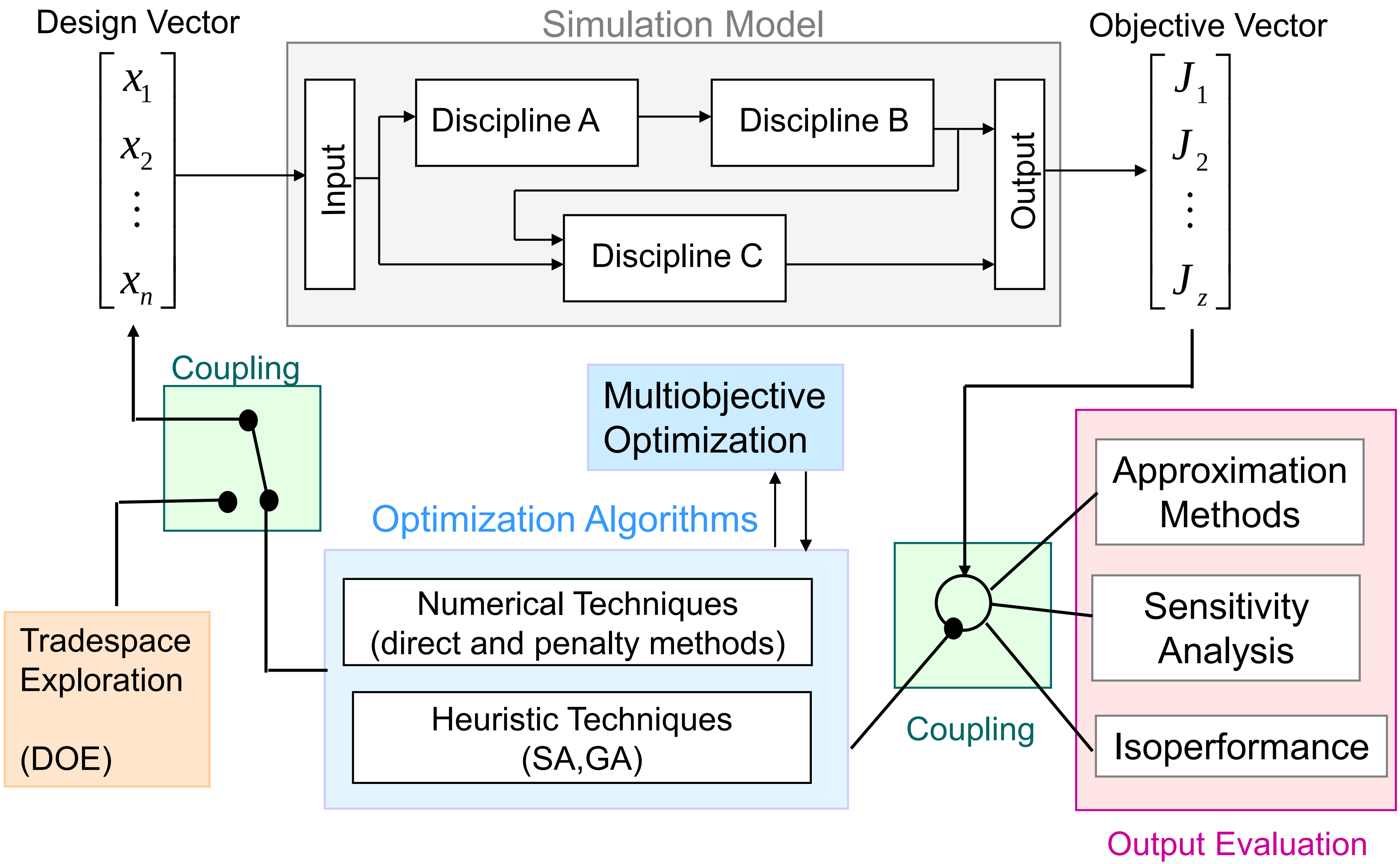
It does this by operating on a large number of variables and functions simultaneously - a feat beyond the power of the human mind.

The path is not biased by intuition or experience.

This path instead of being invisible inside a "black box" becomes more visible by various MDO techniques such as sensitivity analysis and visualization

⇨ Optimization does not remove the designer from the loop, but it helps conduct trade studies

MSDO Framework



Simulation versus Optimization

There are two distinct components of the MSDO process:

The **optimization algorithm** decides how to move through the design space.

The **simulation model** evaluates designs chosen by the optimizer. Both objective functions and constraints must be evaluated.

Sometimes, disciplinary simulation models can be used in an optimization framework, but often they are not appropriate.

There are several different approaches to couple the optimizer and the simulation models (Lecture 4).

Typical Process in MDO

(1) Define overall system requirements
(2) Define design vector **x**, objective **J** and constraints
(3) System decomposition into modules
(4) Modeling of physics via governing equations at the module level - module execution in isolation
(5) Model integration into an overall system simulation
(6) Benchmarking of model with respect to a known system from past experience, if available
(7) Design space exploration (DoE) to find sensitive and important design variables x_i
(8) Formal optimization to find min J(x)
(9) Post-optimality analysis to explore sensitivity and tradeoffs: sensitivity analysis, approximation methods, isoperformance, include uncertainty

In Practice...

(i) Step through (1)-(8)

(ii) The optimizer will use an error in the problem setup to determine a mathematically valid but physically unreasonable solution

OR

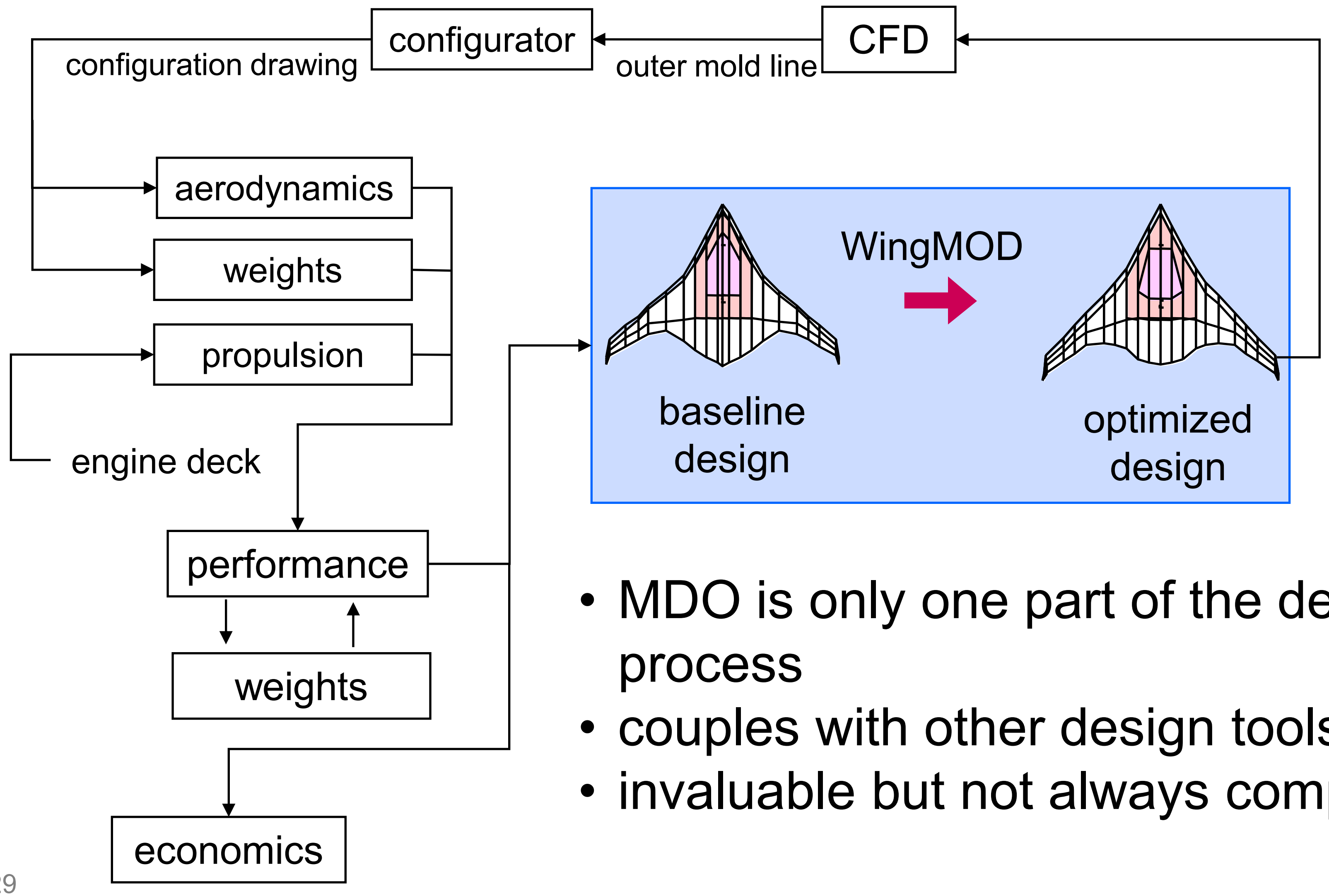
The optimizer will be unable to find a feasible solution (satisfies all constraints)

(iii) Add, remove or modify constraints and/or design variables

(iv) Iterate until an appropriate model is obtained

Although MDO is an automated formalization of the design process, it is a highly interactive procedure...

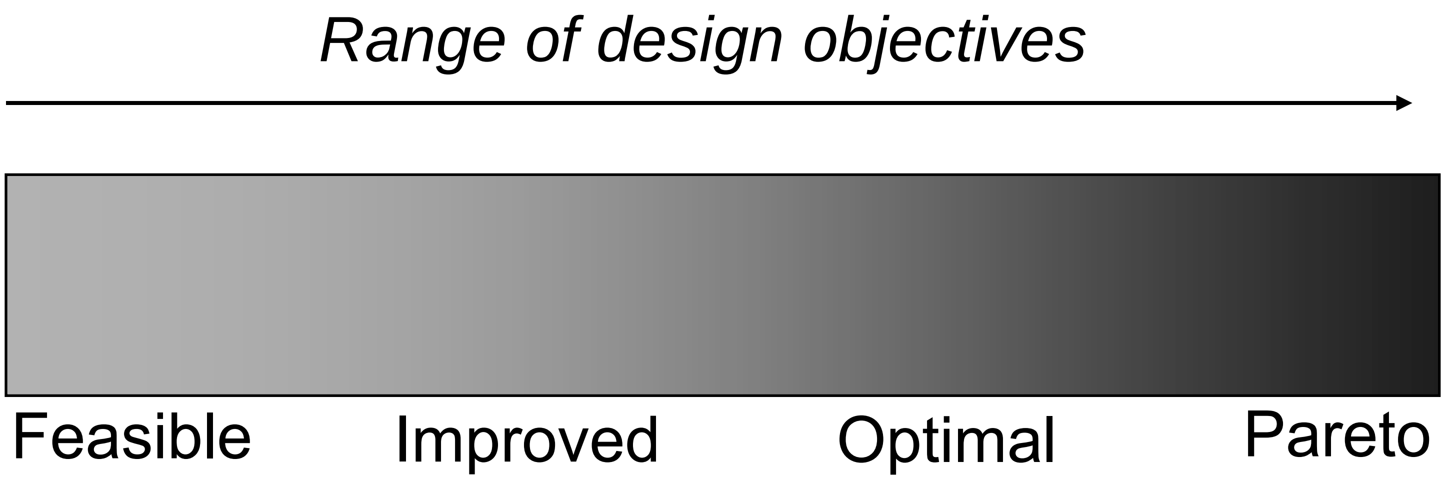
MDO in the Design Process



- MDO is only one part of the design process
- couples with other design tools
- invaluable but not always complete

MDO Uses

- The ‘MD’ portion of ‘MDO’ is important on its own
- Often MDO is used not to find the truly optimal design, but rather to find an improved design, or even a feasible design ...



from Giesing, 1998

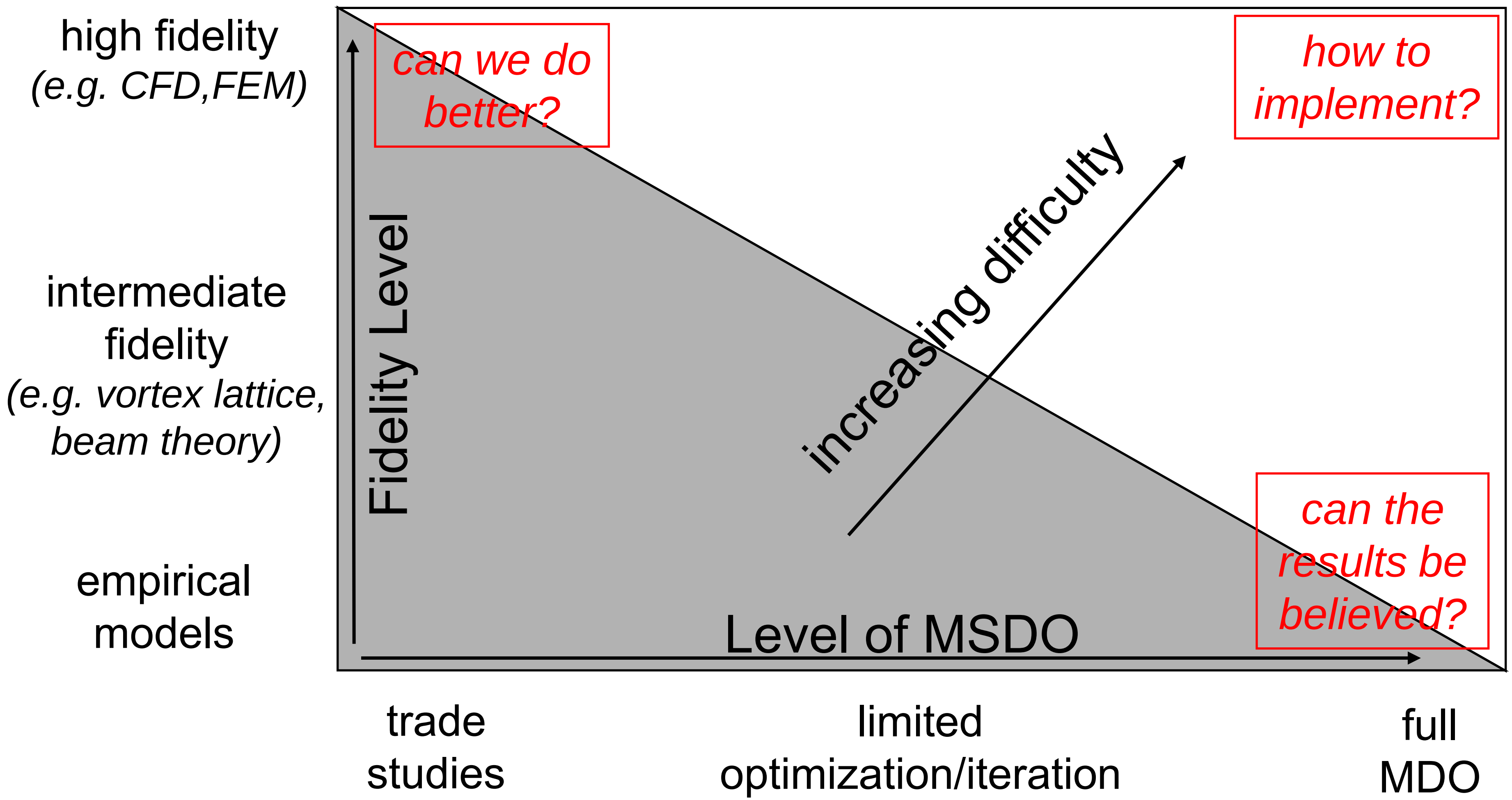
MDO Challenges

MDO Challenges

- Fidelity/expense of disciplinary models
  Fidelity is often sacrificed to obtain models with short computation times.
- Complexity
  Design variables, constraints and model interfaces must be managed carefully.
- Communication
  The user interface is often very unfriendly and it can be difficult to change problem parameters.
- Flexibility
  It is easy for an MDO tool to become very specialized and only valid for one particular problem.

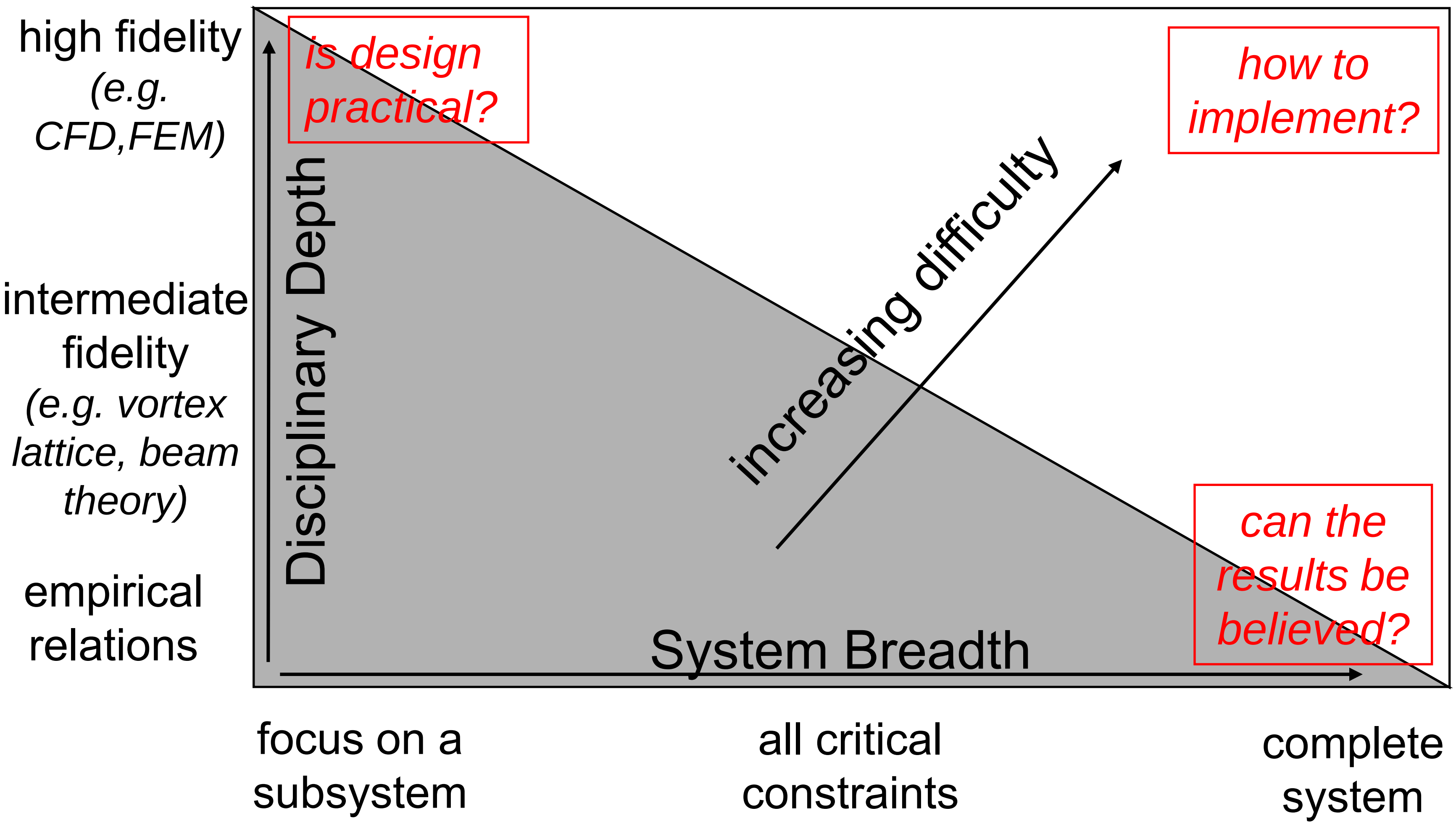
How do we prevent MDO codes from becoming complex, highly specialized tools which are used by a single person (often the developer!) for a single problem?

Fidelity vs. Expense



from Giesing, 1998

Breadth vs. Depth



MDO Pros/Cons

Advantages

- reduction in design time
- systematic, logical design procedure
- handles wide variety of design variables & constraints
- not biased by intuition or experience

Disadvantages

- computational time grows rapidly with number of dv's
- numerical problems increase with number of dv's
- limited to range of applicability of analysis programs
- will take advantage of analysis errors to provide mathematical design improvements
- difficult to deal with discontinuous functions

Lecture summary

- MDO is not a stand-alone, automated design process
- MDO is a valuable tool that requires substantial human interaction and complements other design tools
- Elements of an MDO framework
- MDO Challenges

⇨ Next two lectures will address Modeling & Simulation and Problem Decomposition

References

- Kroo, I.: "MDO applications in preliminary design: status and directions," AIAA Paper 97-1408, 1997.
- Kroo, I. and Manning, V.: "Collaborative optimization: status and directions," AIAA Paper 2000-4721, 2000.
- Sobieski, I. and Kroo, I.: "Aircraft design using collaborative optimization," AIAA Paper 96-0715, 1996.
- Balling, R. and Wilkinson, C.: "Execution of multidisciplinary design optimization approaches on common test problems," AIAA Paper 96-4033, 1996.
- Giesing, J. and Barthelemy, J.: "A summary of industry MDO applications and needs", AIAA White Paper, 1998.
- AIAA MDO Technical Committee: "Current state-of-the-art in multidisciplinary design optimization", 1991.