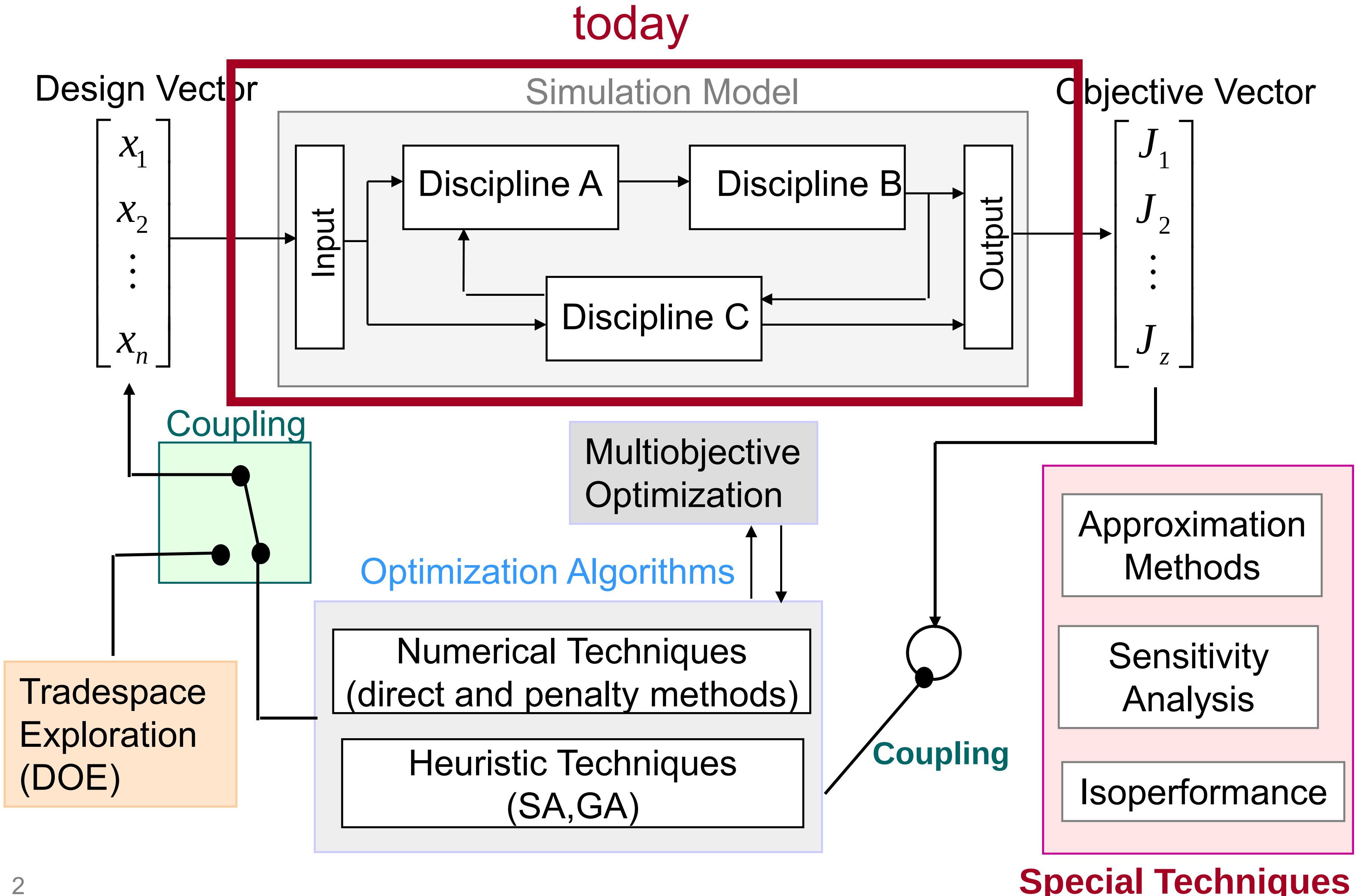
16.888
ESD.77

Multidisciplinary System Design Optimization (MSDO)

Lecture 3: Modeling and Simulation

Prof. Olivier de Weck

MSDO Framework



Today's Topics

Definitions of Modeling and Simulation
- physics-based modeling
- empirical modeling

Model/Simulation Development Process
- module identification
- module ordering: DSM's and N^2 diagrams
- module coding: fidelity and benchmarking
- model execution = simulation

Computational Issues
- coupling disparate CAE/CAD tools

Definitions

MIT esd

Definitions

16.888 ESD.77

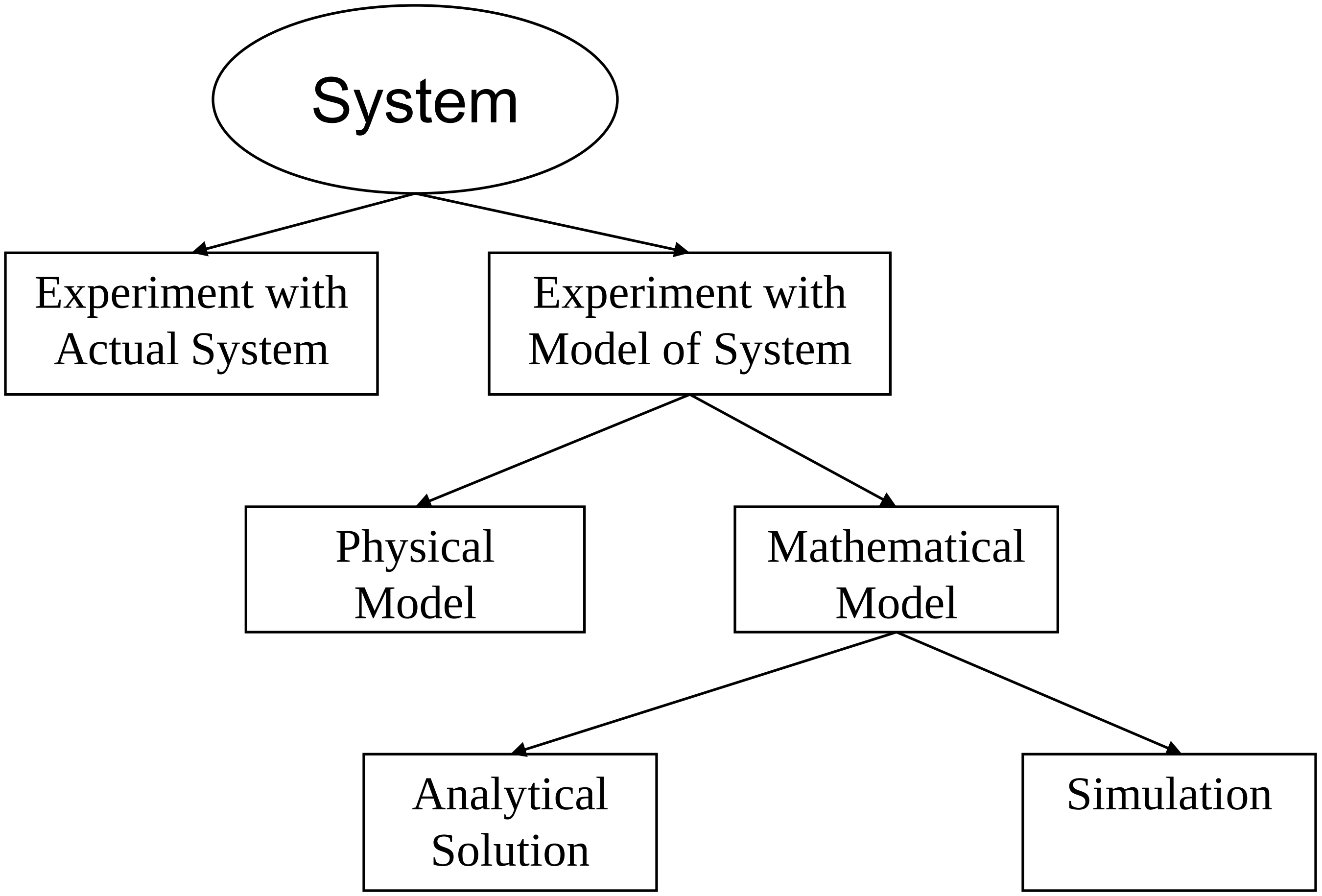
Definition: ***Model*** *(as used in this class)*

A model is a mathematical object that has the ability to predict the behavior of a real system under a set of defined operating conditions and simplifying assumptions.

Definition: ***Simulation*** *(as used in this class)*

Simulation is the process of exercising a model for a particular instantiation of the system and specific set of inputs in order to predict the system response.

From the Reference



Law & Kelton (2000), Simulation Modeling and Analysis 3rd ed., McGraw-Hill, Inc.

Additional Detail – Next Chart

The following chart includes additional detail to emphasize the factors that differentiate a model and a simulation

Simulation/Model Factors:

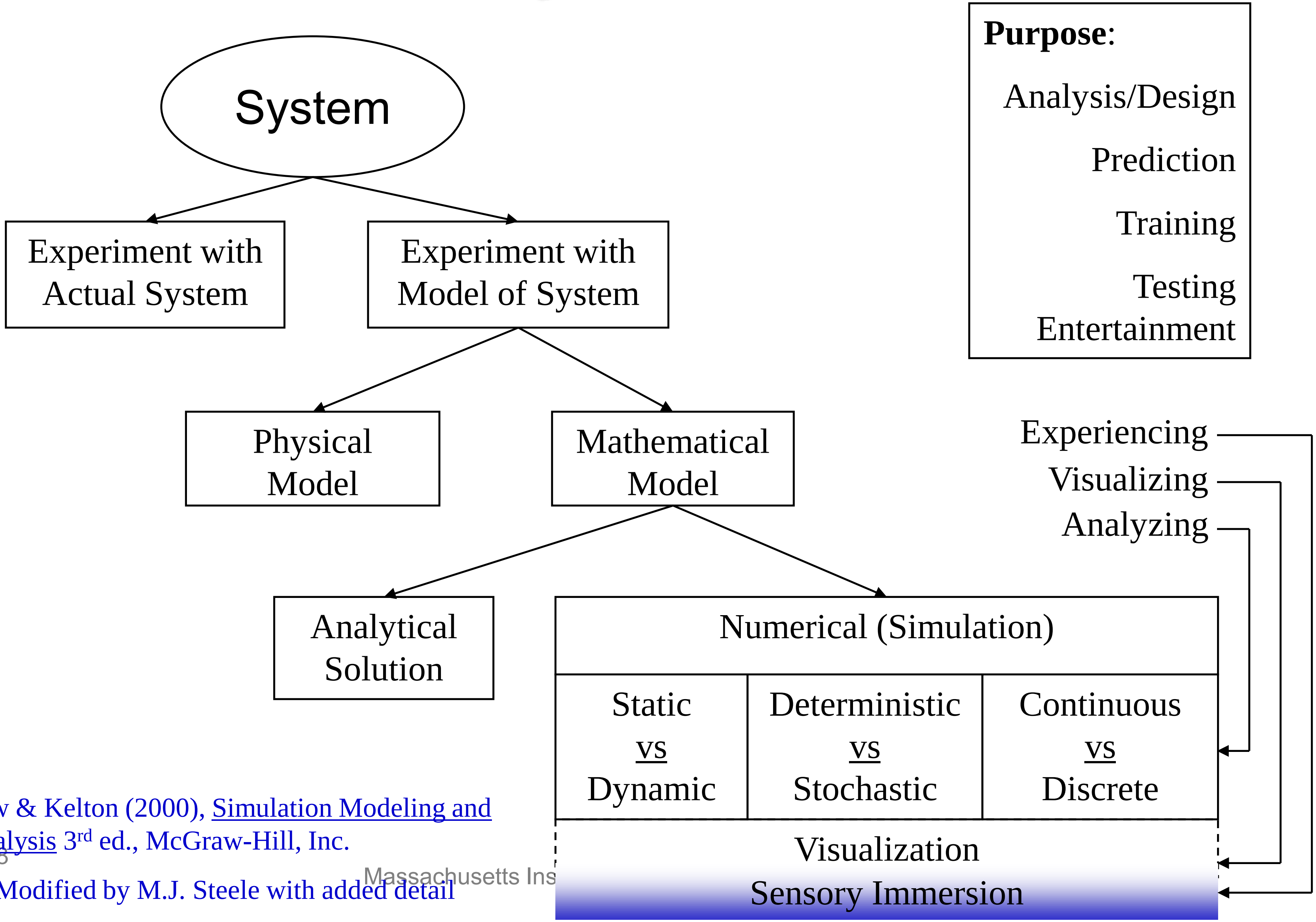
- Real World Variability
- Reaction to Events

These relate back to the purpose of the sim/model

Models should not include all the details for all purposes

- They quickly become unwieldy & expensive

Analysis Methods

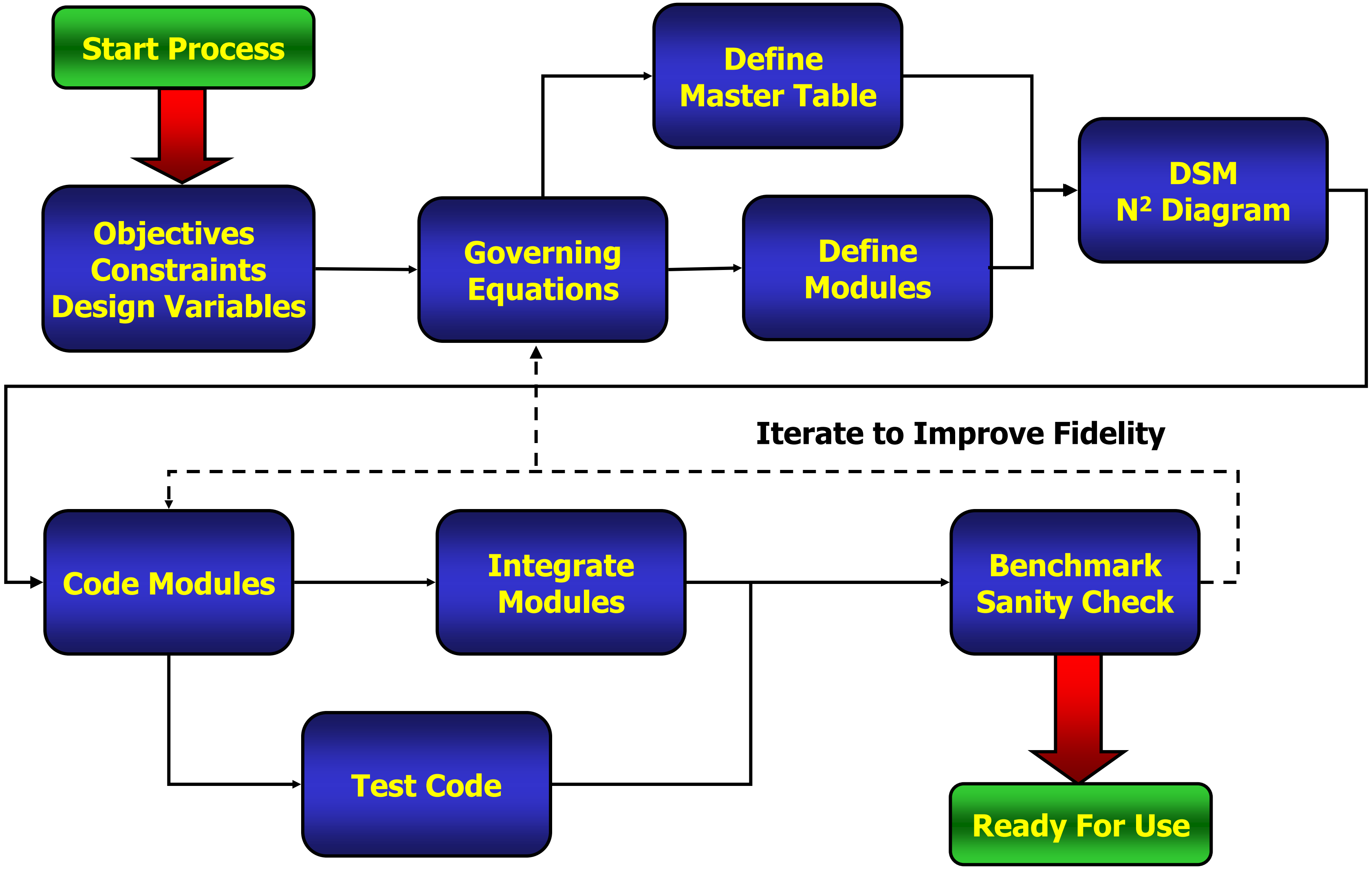


Law & Kelton (2000), Simulation Modeling and Analysis 3rd ed., McGraw-Hill, Inc.

→ Modified by M.J. Steele with added detail

Model and Simulation Development Process

Model Development Process



Objectives, Constraints, Design Variables

- Define Objectives **J** — This is how we want system to behave
- Define Design Variables **x** — Things about system we can change
- Define Constraints and Bounds **g, h** — Must satisfy this
- Determine important fixed parameters **p** — Fixed, outside our control yet important

Influence Matrix

	x_1	…	x_n
J_i	+	+	o
g_j	+	o	+

model relationships

+ influence

o no influence

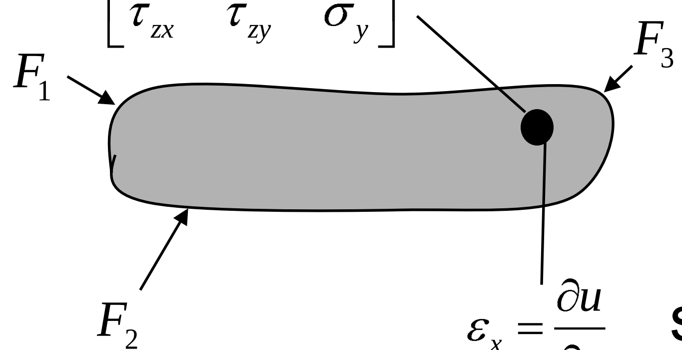
Physics Based Modeling

- Start with governing equations
- Continuum Mechanics for physical systems
- Introduce Boundary Conditions
- Introduce Initial Conditions
- External forcing functions
- Discretize system

Governing Equations

Continuum (Structural) Mechanics

$$S = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_y \end{bmatrix}$$ stress tensor



$$\varepsilon_x = \frac{\partial u}{\partial x}$$ strain

-Equilibrium Equations $\Sigma F_i = 0$

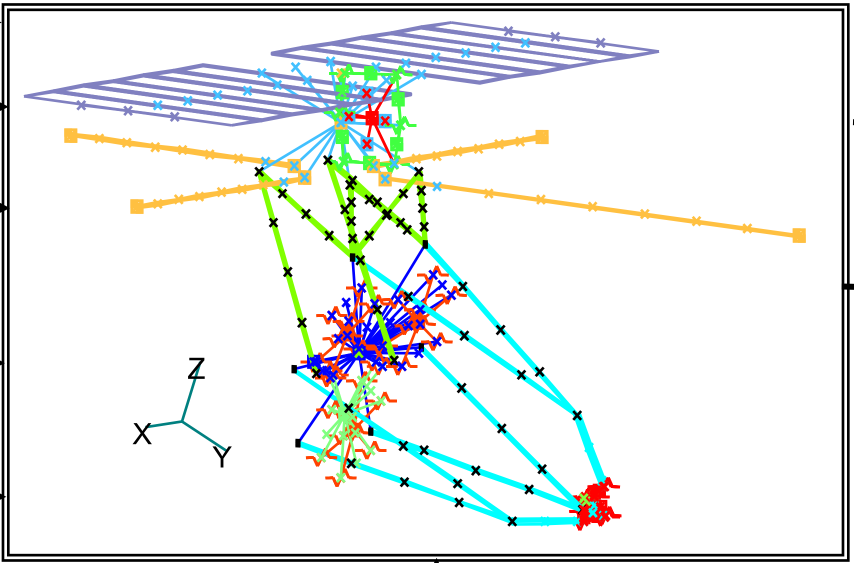
-Constitutive equations $\sigma_x = E\varepsilon_x$

-Compatibility equations $\varepsilon_x = \frac{dx' - dx}{dx}$

Example: Finite Element Model

$$\mathbf{M}\ddot{x}+\mathbf{C}\dot{x}+\mathbf{K}x=F$$

Geometry →
Connectivity →
Material Properties →
Boundary Conditions →
Loads →



→ Mass and Inertia Matrix
→ Deflections, Stress, Strain
→ Natural Frequencies Mode Shapes

↑
Assumptions
Discretization

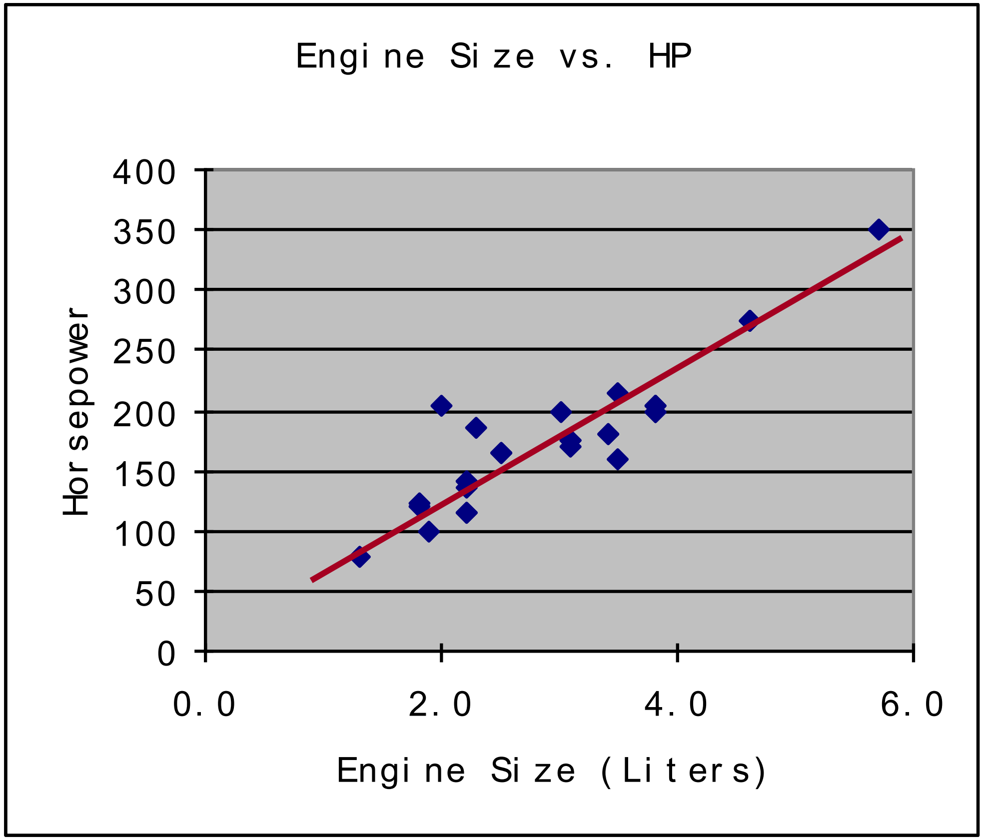
Time as Variable:

Static → Steady State → Transient

Empirical Modeling

- Derive a model, not from physics and first principles, but from observation, i.e. data
- Usually leads to low order models
- Only valid under similar operating conditions
- Many cost models are of this nature

Example: Empirical Modeling



HP = 51.48*ED + 23.12

In-line engine image removed due to copyright restrictions. Animation can be found at HowStuffWorks.com.

… could do physics-Based modeling of this in-line 4 engine, but instead do …

Linear Regression

$$HP = \alpha \cdot ED + \beta$$

How to decompose a system

- What to do when system is new - no experience ?
- First define "black boxes" or modules based on: disciplinary tradition, degree of coupling of governing equations or availability of analysis software
- Crisply define inputs and outputs of each module

Ref: Rogers, J.L.: "A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem", NASA TP2903, 1989

Partitioning of Equations

E1 $x_1 x_2 - 2x_3 + 2 = 0$

E2 $x_2 + 3x_5 - 9 = 0$

E3 $x_1 - x_4 x_5 - x_3 + 10 = 0$

E4 $9x_5 - 3x_2 + 7 = 0$

E5 $x_2 x_5 - x_2 x_4 + x_2 - 9 = 0$

-5 variables
-5 independent equations
-no degrees of freedom

1. **Solve x_2,x_5 from E_2 and E_4**
2. **Solve x_4 from E_5**
3. **Solve x_1 and x_3 from E_3 and E_1**

	X_1	X_2	X_3	X_4	X_5
E_1	1	1	1		
E_2		1			1
E_3	1		1	1	1
E_4		1			1
E_5		1		1	1

Occurrence matrix for system of equations

	X_2	X_5	X_4	X_1	X_3
E_2	1	1			
E_4	1	1			
E_5	1	1	1		
E_3		1	1	1	1
E_1	1			1	1

Module 1 (E_2, E_4; X_2, X_5)
Module 2 (E_5; X_4)
Module 3 (E_3, E_1; X_1, X_3)

Occurrence matrix showing the system of equations partitioned into three subsets

Module Definition

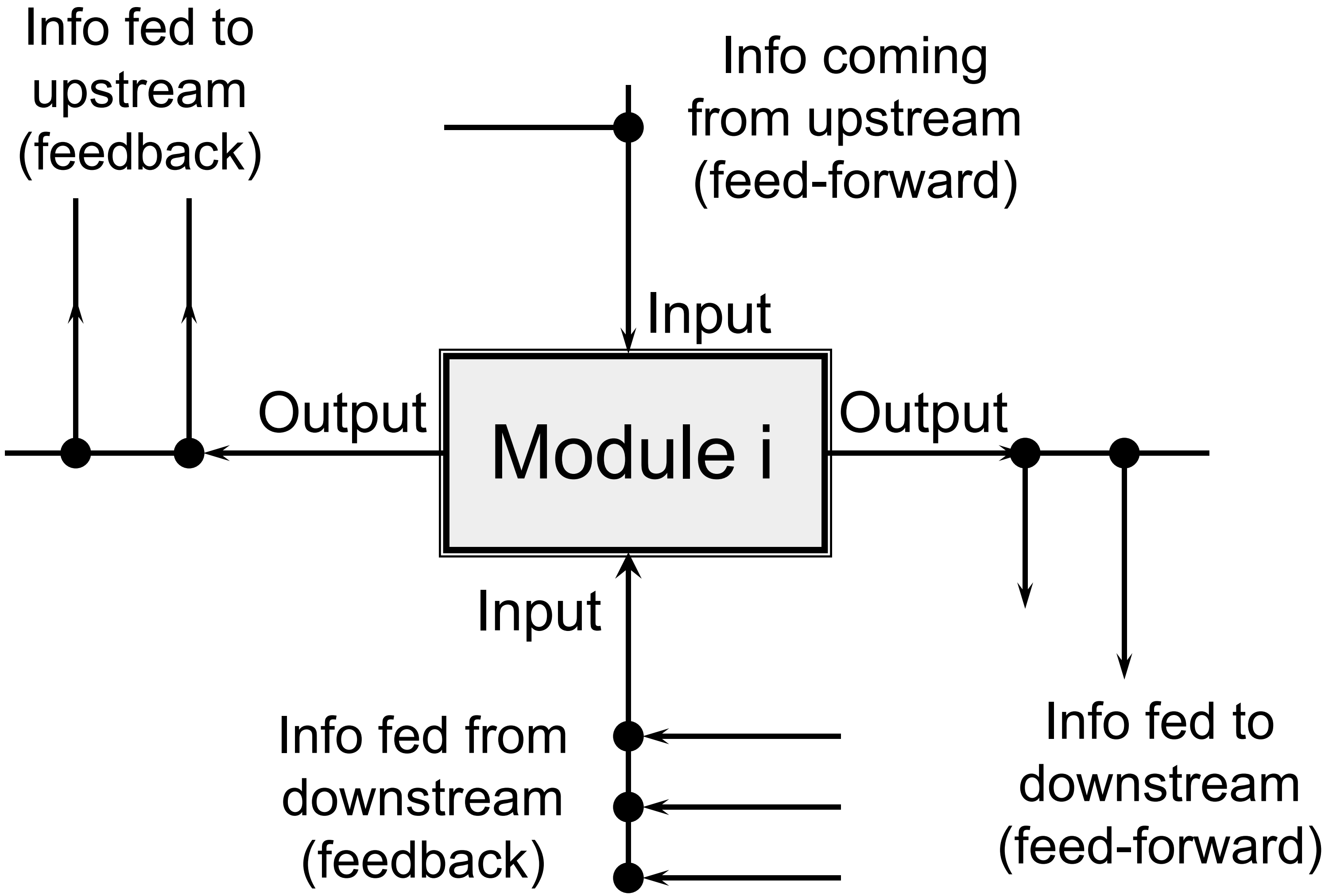
What is a module in MSDO ?

A module in multidisciplinary system design optimization is a finite group of tightly coupled mathematical relation-ships who are under the responsibility of a particular individual or organization, and where some variables represent independent inputs while others are dependent outputs. The module frequently appears as a "black box" to other individuals or organizations .

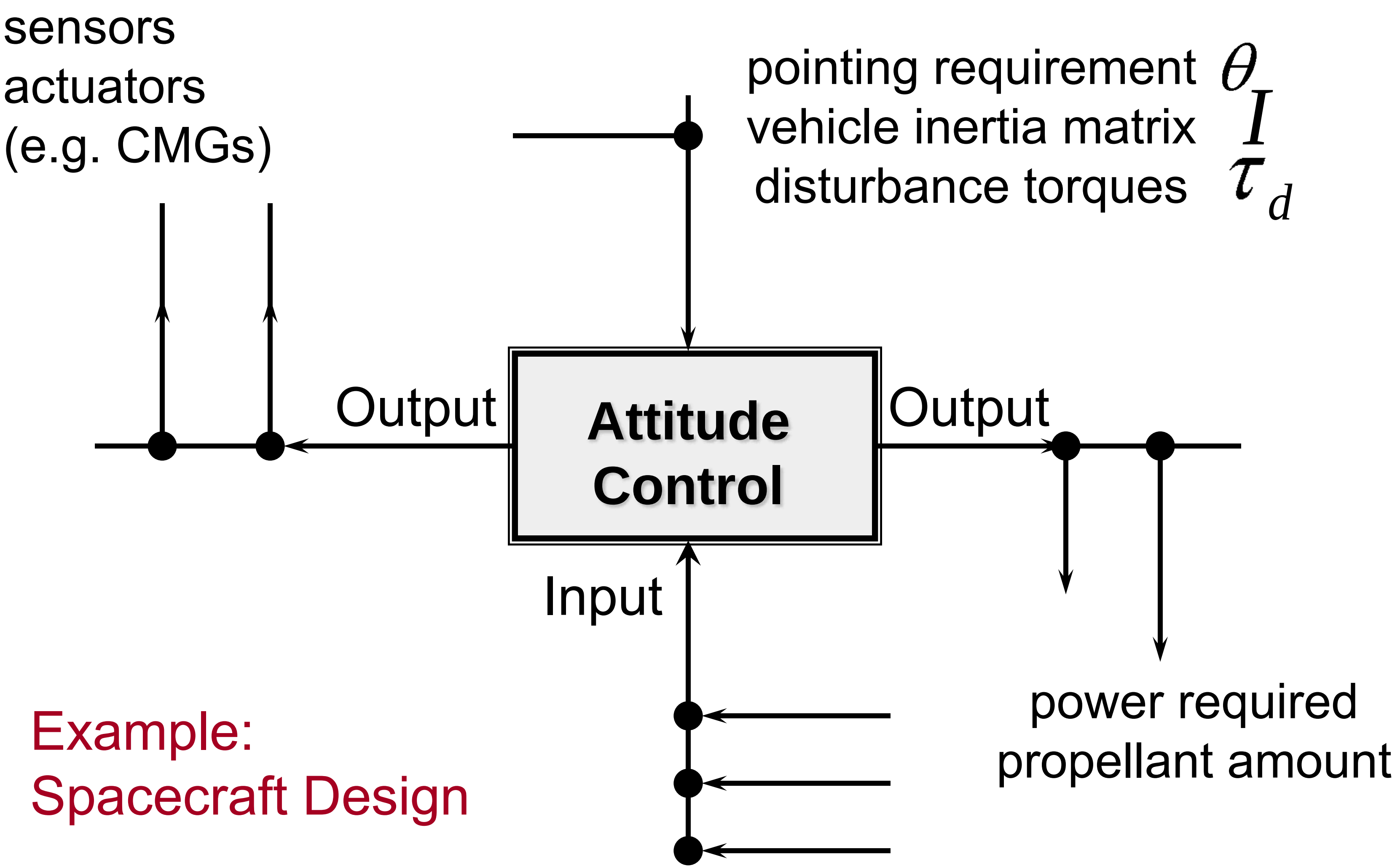
Modules for Simulation

- A module within a simulation architecture may be defined as a piece of computer code which:
  - Performs a compact set of calculations.
  - Contains a single entry point and exit point.
  - May be tested in isolation.
- Attributes of a good modular unit within a simulation architecture include:
  - **High internal coupling within the module**
    - All sub-functions within the module contribute to form a single primary function.
  - **Low coupling between modules**
    - Minimize the number of variables that flow between modules.
  - **Minimization of feedback loops**
    - Data flow is processed sequentially from input to output.

Module Inputs and Outputs



Module Example



Example:
Spacecraft Design

The N^2 Diagram

- An NxN matrix used to develop and **organize interface information**.
- Similar to a Design Structure Matrix (DSM)
- Each module within the simulation architecture is placed along the **diagonal**.
- Provides a visual representation of the **flow of information** through the simulation architecture.
- Helps to **identify critical modules** that have many inputs and outputs. The fidelity of critical modules should be thoroughly tested and verified.
- Explicitly defines all **inputs** and **outputs** for macro-modules and modules.
- Allows for “plug and play”
  - Independent testing
  - Alternative modules easily analyzed
  - Can increase overall model fidelity incrementally

Initial random sequencing

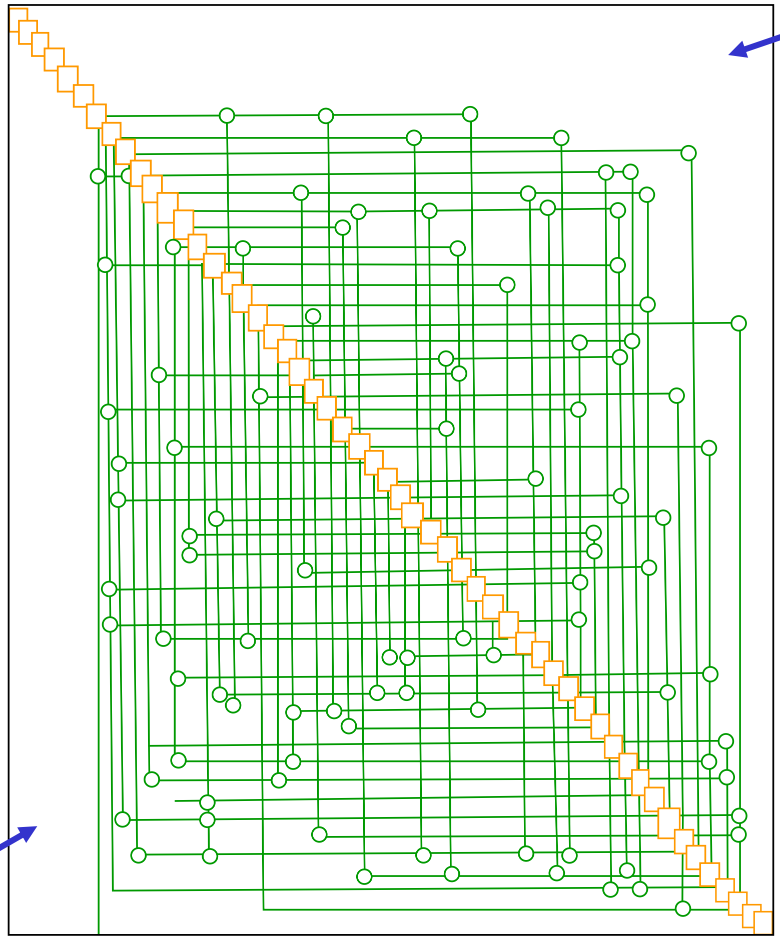

Initial, random arrangement of modules on the N-square matrix diagonal.

Image by MIT OpenCourseWare.

feedforward

feedback

Execution sequence goes from upper left corner to lower right corner

Problem: Each instance of feedback requires an iteration

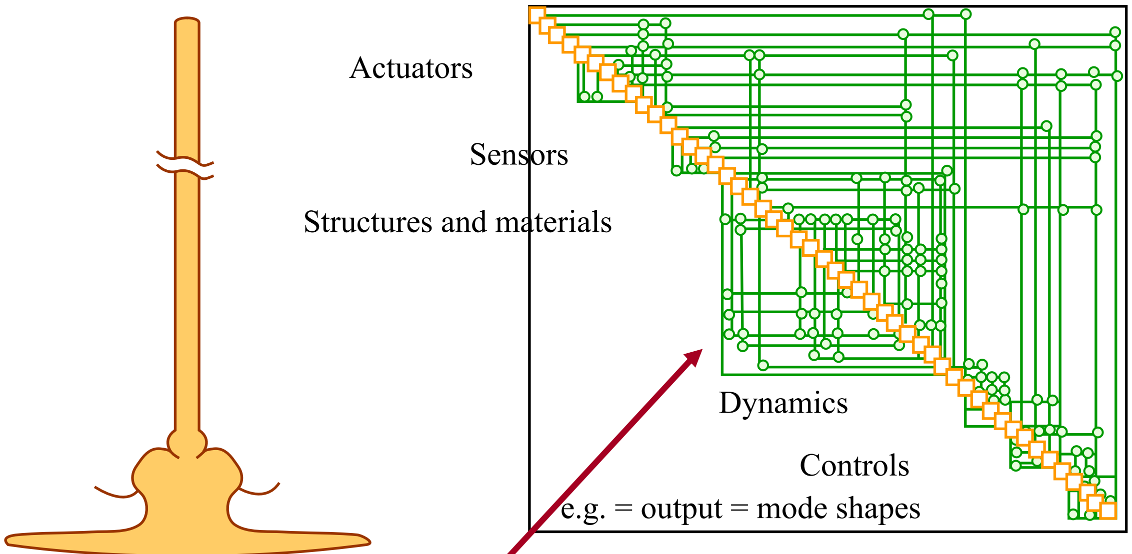
Ordered sequence

The Need

Identify coupling between discipline or subsystems in a complex system.

The Solution

Automatic identification of subsystems based on interdependence of design variable and constraints.



Modules in the N-square matrix resequenced to reduce feedback.

Image by MIT OpenCourseWare.

Systematic permutation groups and reduces number of feedback loops

TPF Example : Overview N^2 Diagram

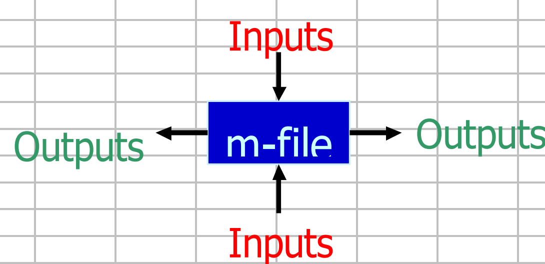
16.888
ESD.77

TMAS N-Squared Diagram

Modules: Design Vector (Design Vector); Const. Vector (Architecture Constants Vector); Env. Module (Environment); Apert. Conf. Module (Aperture Configuration); Spacecraft Payload and Bus Module (Payload, Power, Thermal, Propulsion, Communications, Structure); Dynamics, Control, & Stability Module (Truss Design, State-Space Plant Model, Attitude Determination and Control, Model Integration, Performance Assesment); Deployment & Operations Module (Orbit Transit, Launch, Operations); GINA Systems Analysis Module (Capability, Performance, Cost, Cost Per Function, Adaptability)

	Design Vector	Architecture Constants Vector	Environment	Aperture Configuration	Payload	Power	Thermal	Propulsion	Communications	Structure	Truss Design	State-Space Plant Model	Attitude Determination and Control	Model Integration	Performance Assesment	Orbit Transit	Launch	Operations	Capability	Performance	Cost	Cost Per Function	Adaptability
Design Vector	■		1	2,3,4	2,4	1,2,3	2,3,4	2,3	1	2,3,4		2,3,4	1,2,3,4	3		1		1,2,3	1,2	2	2,3,4		1,2,3,4
Architecture Constants Vector		■		11	2	2,3	3	1,2	4	13,14	14,31	15,16	11,32-40					3	5 to 11		2,12		all
Environment			■				1						1,2,3										
Aperture Configuration				■						1	1		1						1				
Payload					■	3,4		1,2		1,2													
Power						■	1,6	6,7	6,7	2,3													
Thermal						2,3,5,6	■			1,4													
Propulsion						4		■		3													
Communications						6			■	1 to 5													
Structure							8,10	6		■			6,10		3,4,5		6,9				6		
Truss Design											■	1,2		1									
State-Space Plant Model												■	1	1,2									
Attitude Determination and Control													■										
Model Integration														■	3								
Performance Assesment															■				1				
Orbit Transit																■	1,2,3	4					
Launch																	■				2		
Operations																		■	2		1		
Capability																			■	7,8,9			
Performance																				■		4	
Cost																					■	6	
Cost Per Function																						■	1
Adaptability																							■

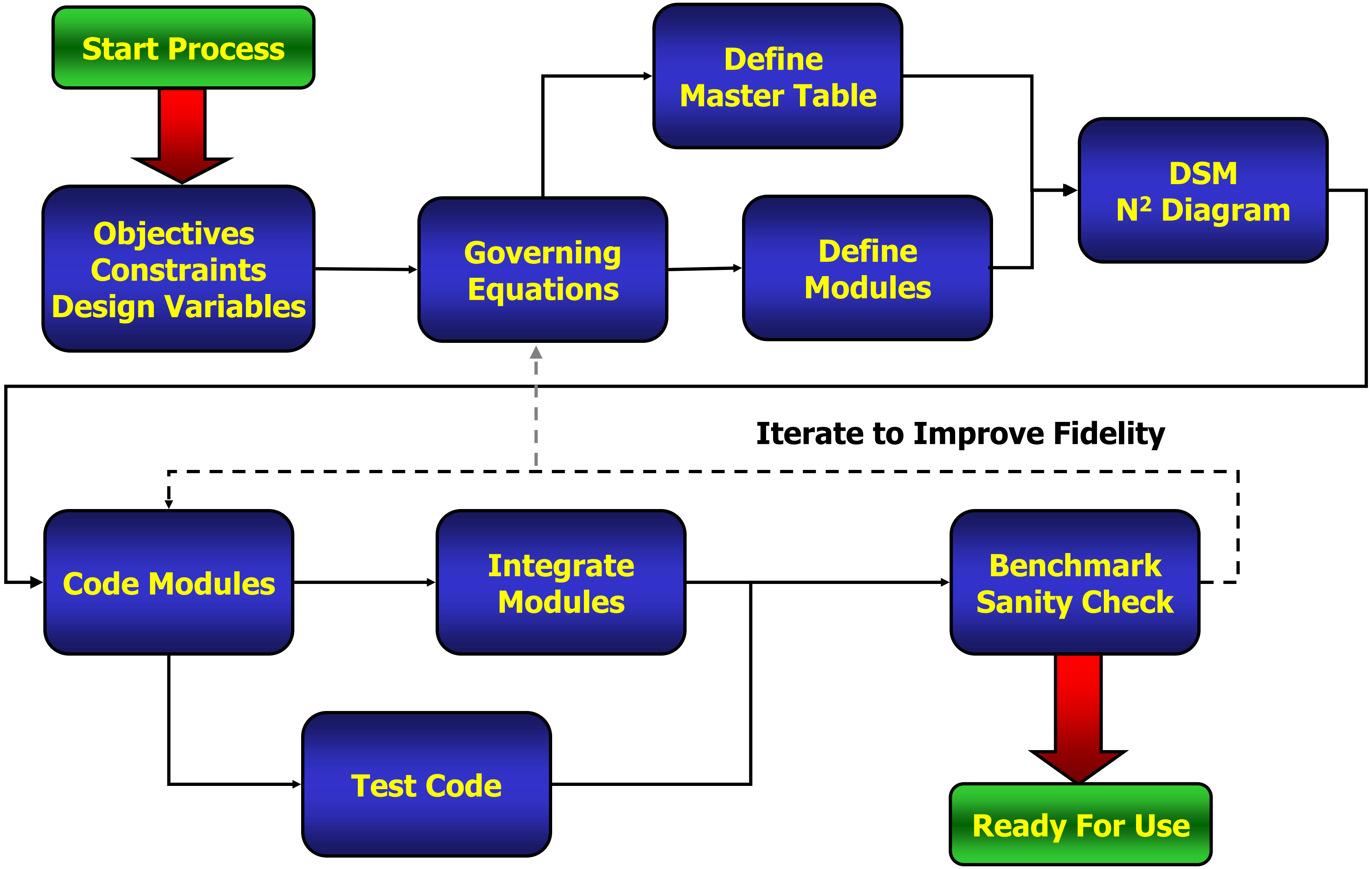
Sub-Modules
- Design Vector
- Architecture Constants Vector
- Environment
- Aperture Configuration
- Payload
- Power
- Thermal
- Propulsion
- Communications
- Structure
- Truss Design
- State-Space Plant Model
- Attitude Determination and Control
- Model Integration
- Performance Assesment
- Orbit Transit
- Launch
- Operations
- Capability
- Performance
- Cost
- Cost Per Function
- Adaptability



Coding - Benchmarking

- Coding of modules can be done in parallel, once the I/O structure has been decided
- Use "dummy" input data to exercise modules in isolation
- Integrate modules step-by-step starting from upper left corner in N^2-Diagram
- Do end-to-end simulation test before release
- Benchmark ("validate") simulation against known cases (experimental data)

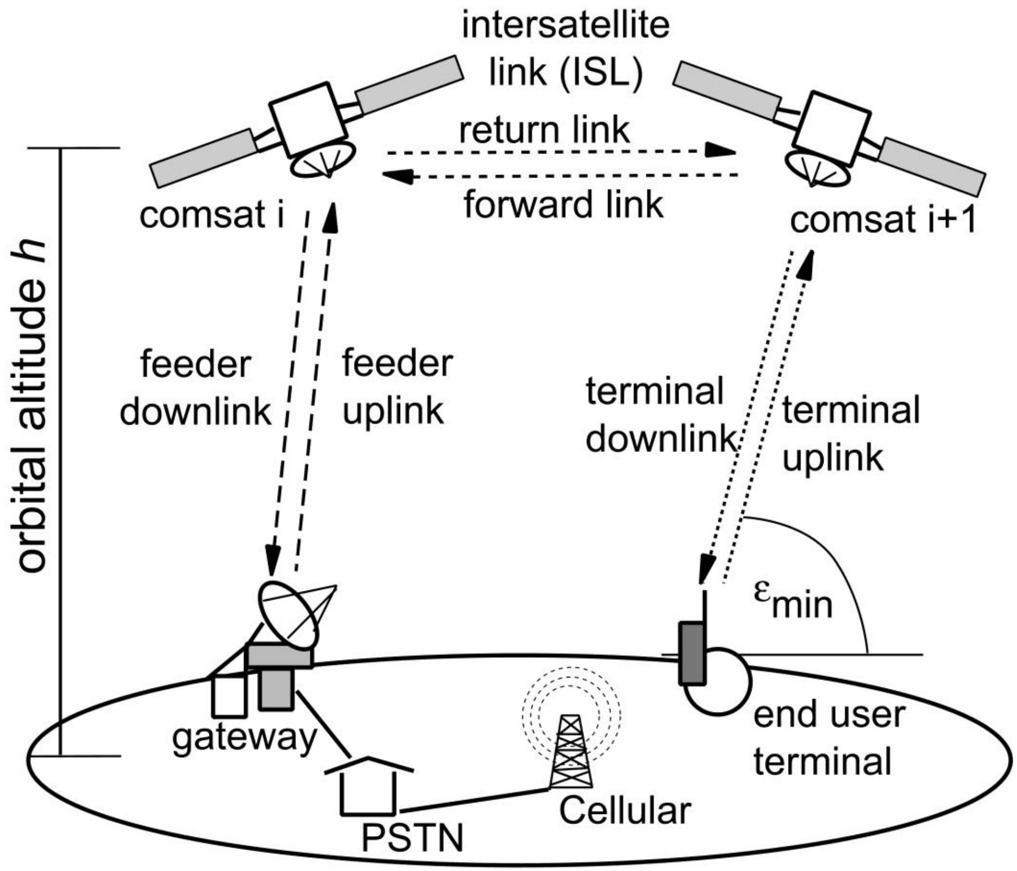
Recap

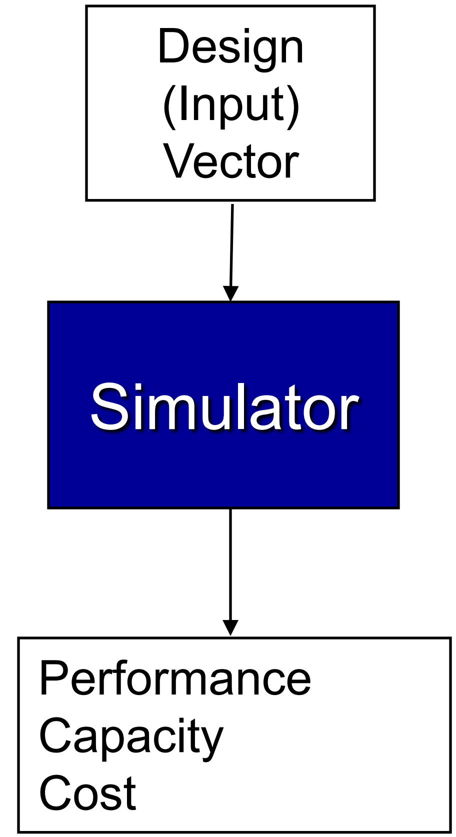

Start Process
Objectives
Constraints
Design Variables
Governing
Equations
Define
Master Table
Define
Modules
DSM
N² Diagram
Iterate to Improve Fidelity
Code Modules
Integrate
Modules
Benchmark
Sanity Check
Test Code
Ready For Use

Example

de Weck, O. L. and Chang D., ”Architecture Trade Methodology for LEO Personal Communication Systems “, *20th International Communications Satellite Systems Conference*, Paper No. AIAA-2002-1866, Montréal, Québec, Canada, May 12-15, 2002

Example: Communications Satellites





Can we quantify the conceptual system design problem using modeling and simulation?

Design (Input) Vector X

- The design variables are:

Astro-dynamics
- – Constellation Type: C
- – Orbital Altitude: h
- – Minimum Elevation Angle: ε_{min}

Satellite Design
- – Satellite Transmit Power: P_t
- – Antenna Size: D_a
- – Multiple Access Scheme MA:

Network
- – Network Architecture: ISL

Design Space

Polar, Walker	
500,1000,1500,2000	[km]
2.5,7.5,12.5	[deg]
200,400,800,1600,2400	[W]
1.0,2.0,3.0	[m]
MF-TDMA, MF-CDMA	[-]
yes, no	[-]

X_{1440}=

```
   C:  'walker'
   h:  2000
emin:  12.5000
  Pt:  2400
  DA:  3
  MA:  'MFCD'
 ISL:  0
```

This results in a 1440 full factorial, combinatorial conceptual design space

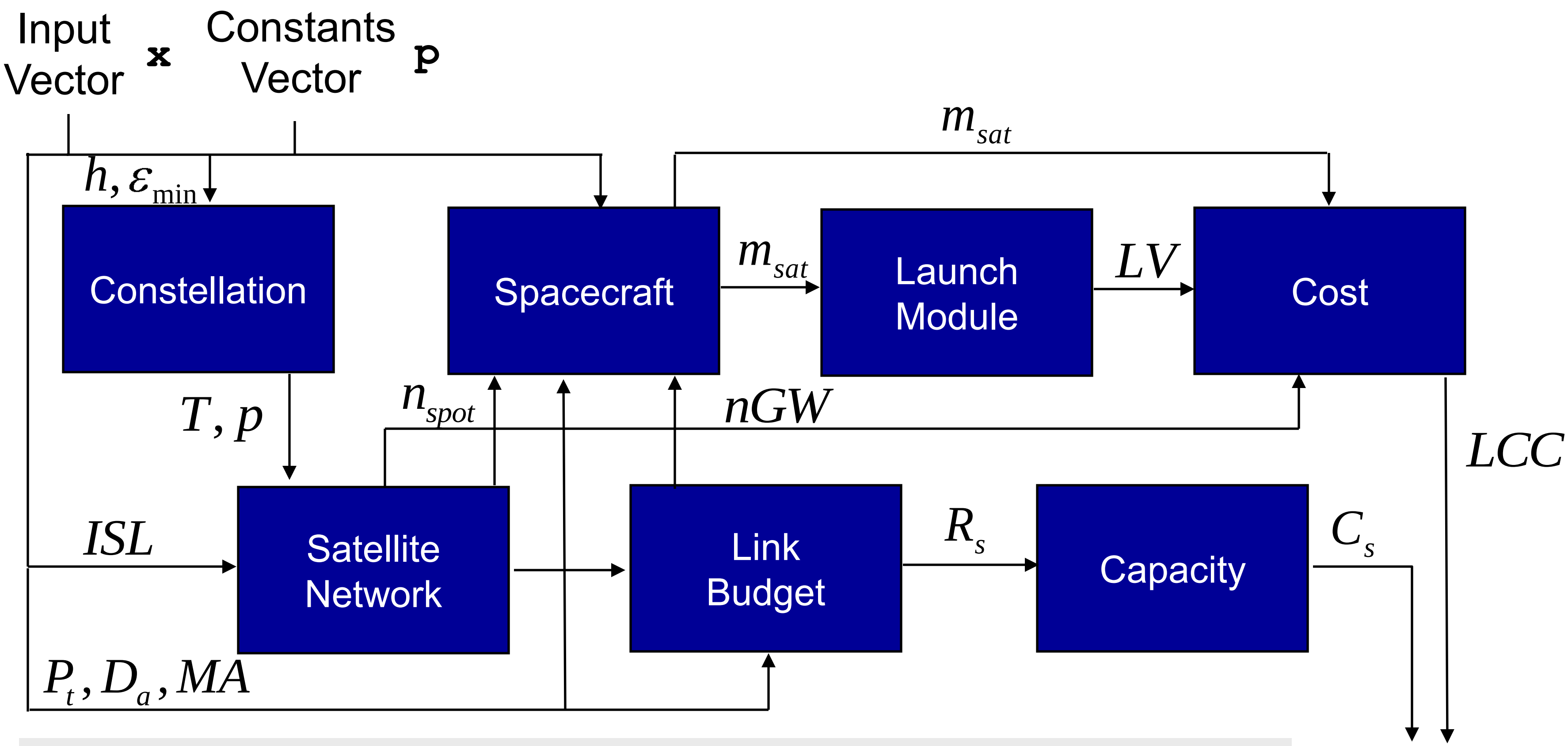
Objective Vector (Output) J

- Performance (fixed)
  - Data Rate per Channel: R=4.8 [kbps]
  - Bit-Error Rate: $p_b=10^{-3}$
  - Link Fading Margin: 16 [dB]
- Capacity
  - C_s: Number of simultaneous duplex channels
  - C_{life}: Total throughput over life time [min]
- Cost
  - Lifecycle cost of the system (LCC [$]), includes:
    - Research, Development, Test and Evaluation (RDT&E)
    - Satellite Construction and Test
    - Launch and Orbital Insertion
    - Operations and Replenishment
  - Cost per Function, CPF [$/min]

Consider

$$J_{1440} = \begin{pmatrix} \text{Cs: } 1.4885e{+}005 \\ \text{Clife: } 1.0170e{+}011 \\ \text{LCC: } 6.7548e{+}009 \\ \text{CPF: } 6.6416e{-}002 \end{pmatrix}$$

Multidisciplinary Simulator Structure



m_{sat}	Satellite Mass	n_{spot}	Number of spot beams
T	Number of Satellites	nGW	Number of gateways
p	Number of orbital planes	LV	Launch vehicle selection

Note: Only partial input-output relationships shown

Governing Equations

a) Physics-Based Models

Energy per bit over noise ratio:

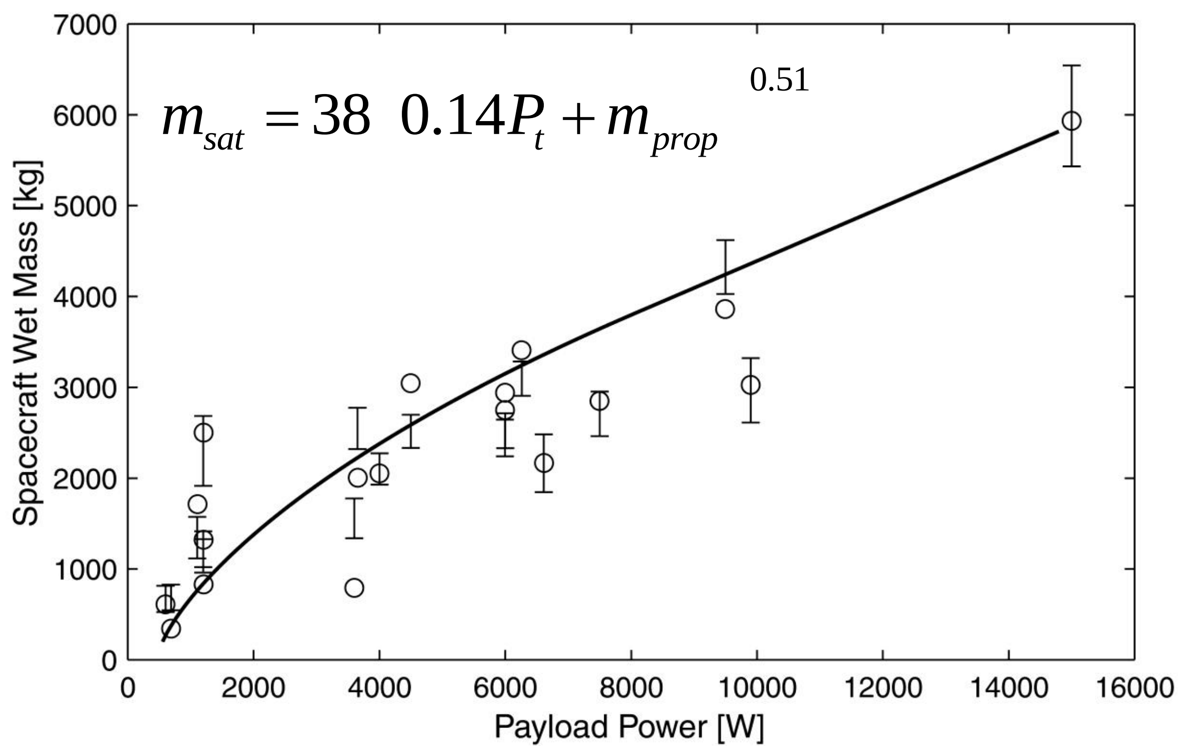
$$\frac{E_b}{N_0} = \frac{P G_r G_t}{k L_{space} L_{add.} T_{sys.} R}$$

(Link Budget)

b) Empirical Models

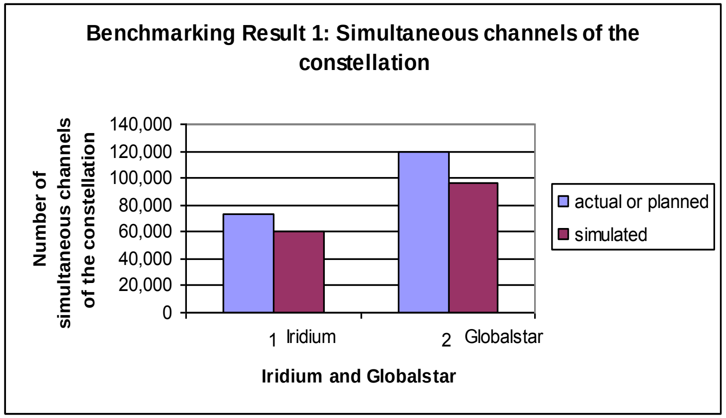
(Spacecraft)

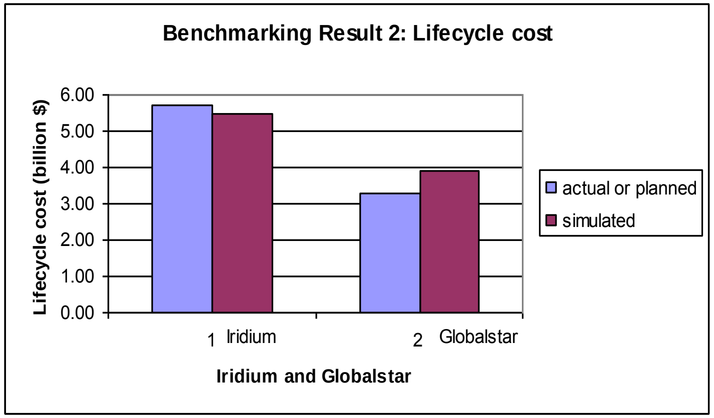
Scaling models derived from FCC database

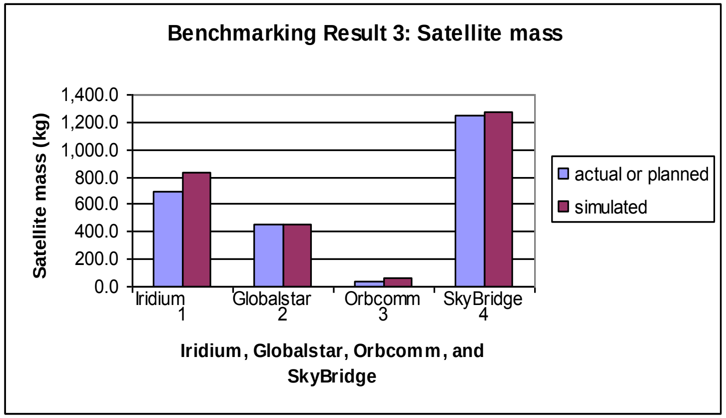


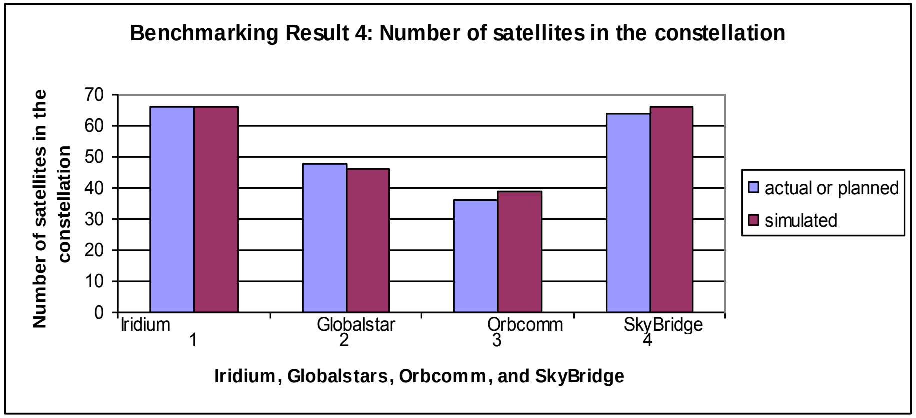
Benchmarking

Benchmarking is the process of validating a model or simulation by comparing the predicted response against reality.

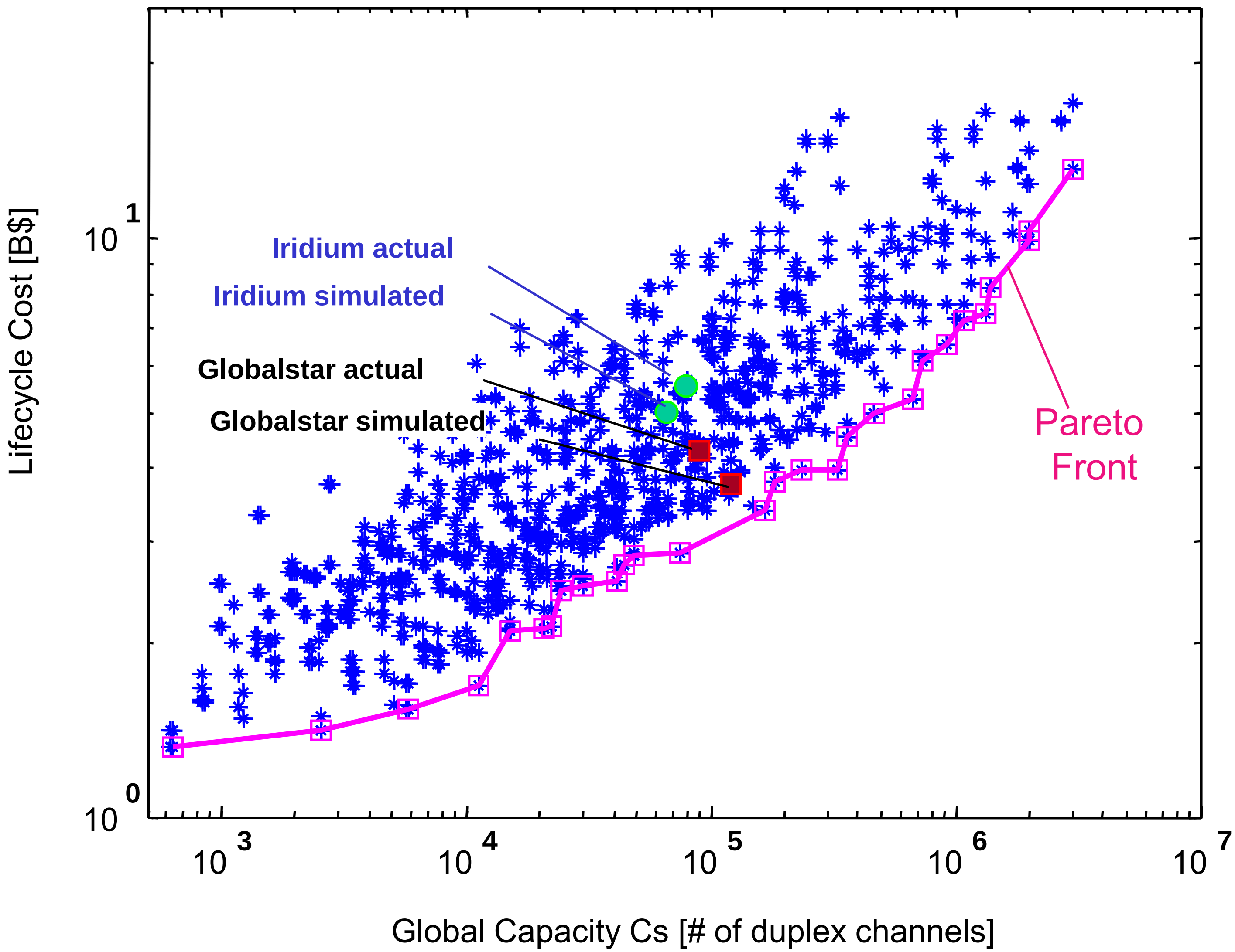








Simulation Results



Simple Example
(Prep for Homework A1)

Example: Communications Satellite

Design of a geosynchronous communications satellite

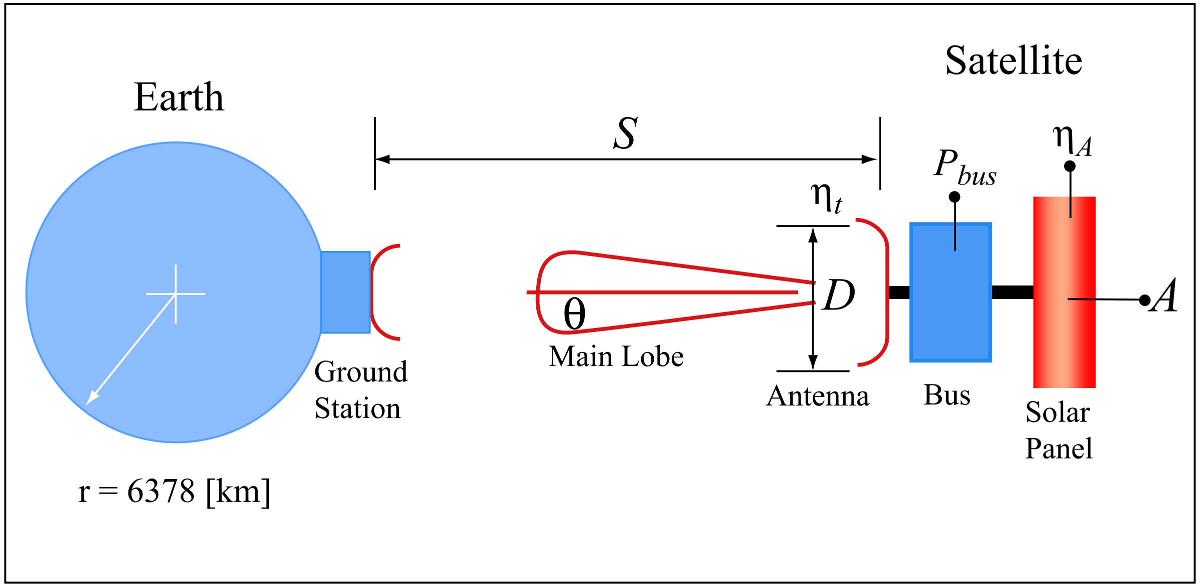


Image by MIT OpenCourseWare.

Design problem (Define D.V., objectives, constraints):
How should antenna (D) and solar array (A) be sized for a given orbital period (p) such that a data rate requirement ($R=R_{req}$) is met, while minimizing cost (C) ?

Comsat: Governing Equations

Objective: min C , *Constraint*: $R>=R_{req}$

Communications: $$R = \alpha P_t \frac{D^2 \eta_t}{16 S^2} \quad \text{[bps]}$$ (link budget)

Power: $$P_t = A\eta_A W_o \cos \; \theta_{avg} \; - P_{bus} \quad \text{[W]}$$ (power budget)

Orbits: $$p = 1.66 \cdot 10^{-4} \;\; S + r_E \;\; {}^{3/2} \quad \text{[min]}$$ (orbital period)

Cost: $$C = 2500 \cdot D^2 + 12000 \cdot \;\; A + 1 \;\; + 100 \cdot P_{bus} \quad \text{[\$]}$$ (cost budget)

Bus Engineering: $$P_{bus} = 10 \cdot \sqrt{P_t} \quad \text{[W]}$$

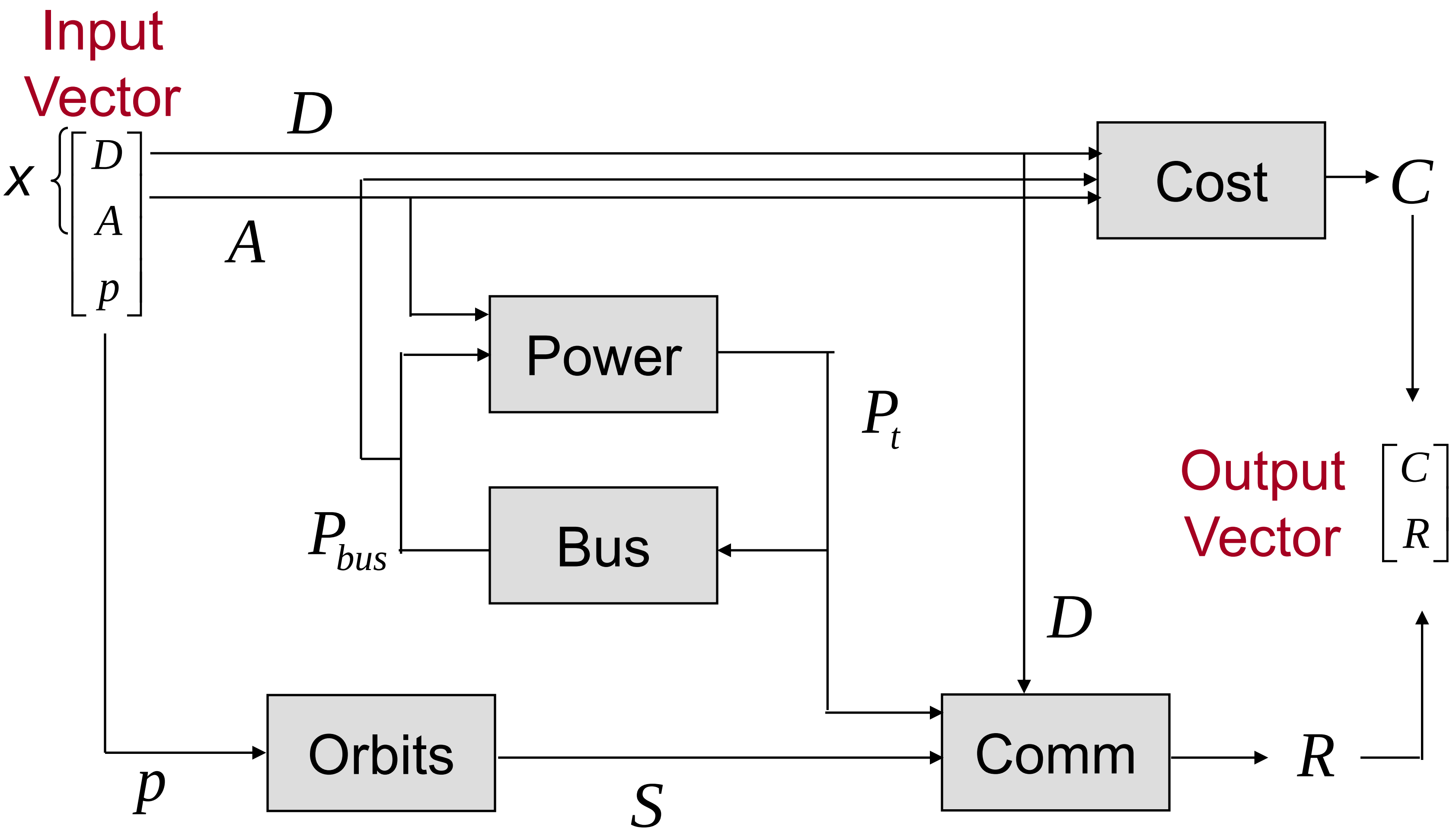
Comsat: Master Table

D	Antenna Diameter	[m]	design var.
A	Solar Panel Area	[m^2]	design var.
p	Orbital Period	[min]	design var.
R	Data Rate	[bps]	constraint
C	Cost	[$]	objective
P_t	Transmitter Power	[W]	dependent
P_{bus}	Bus Power	[W]	dependent
θ_a	Sun incidence angle	[deg]	parameter
$\eta_{a,t,}$	array/xmit efficiencies	[%]	parameter
S	Orbital altitude	[km]	dependent
α	constant	[-]	parameter
W_o	Solar constant	[W/m^2]	parameter

ComSat Block Diagram

16.888
ESD.77

BLOCK DIAGRAM



Comsat N^2 Diagram

In	p	D	A		D,A	
	Orbits	S				
		Comm				R
		Pt	**Power**	Pt		
			Pbus	**Bus**	Pbus	
					Cost	C
						Out

iterative block

Computational Implementation

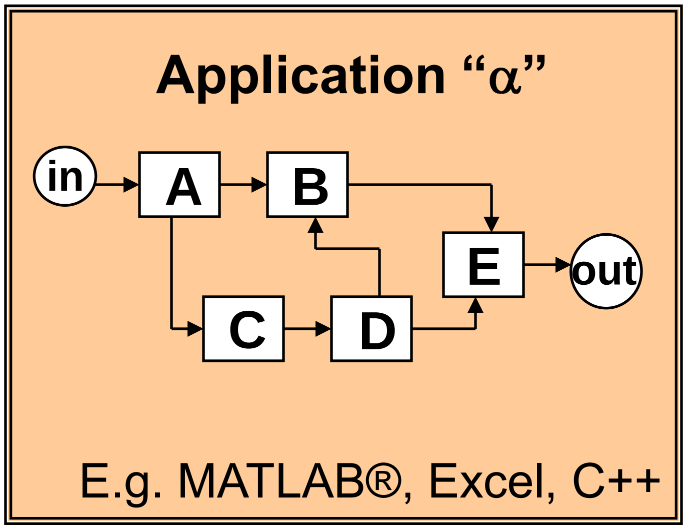
Computational Issues

- Computer technologies have been changing the environment of engineering design - enabling MDO
- Hardware: Advances in processor speed, memory and storage
- Software: Powerful disciplinary analysis and simulation programs (e.g. Nastran, Fluent …)
- This also creates new difficulties: Most activities involve stand-alone programs and many engineers spend 50-80% of their time organizing data and moving it back-and-forth between applications

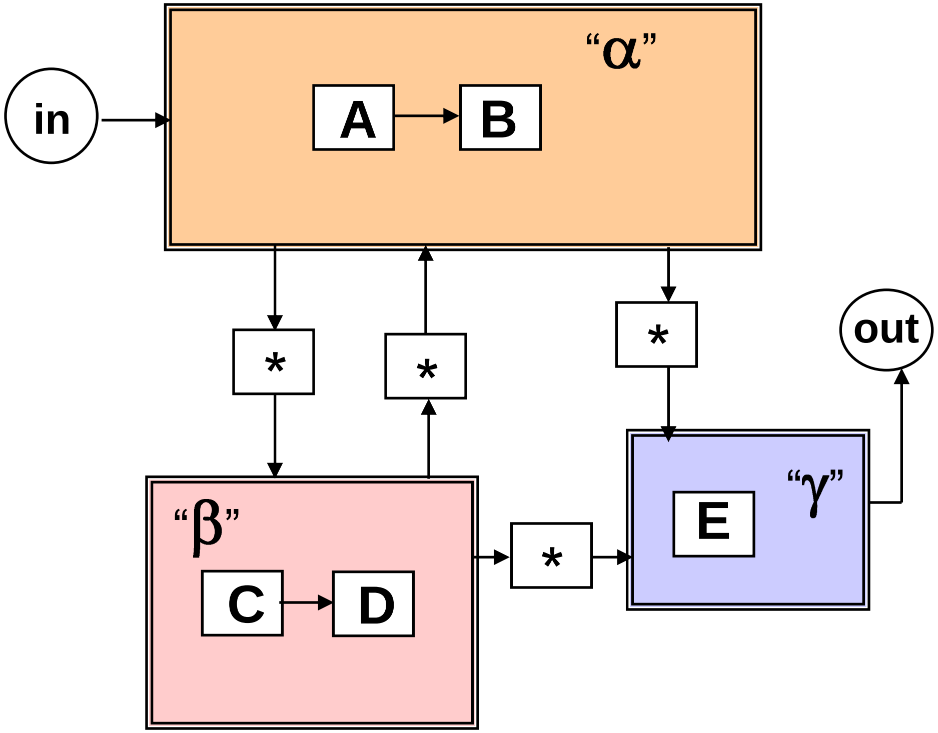
⇨ Data must be shared between disciplines more easily

Coupling Disparate Tools

Case 1:
Within one application on the same computer



Case 2:
Between different applications on the same computer

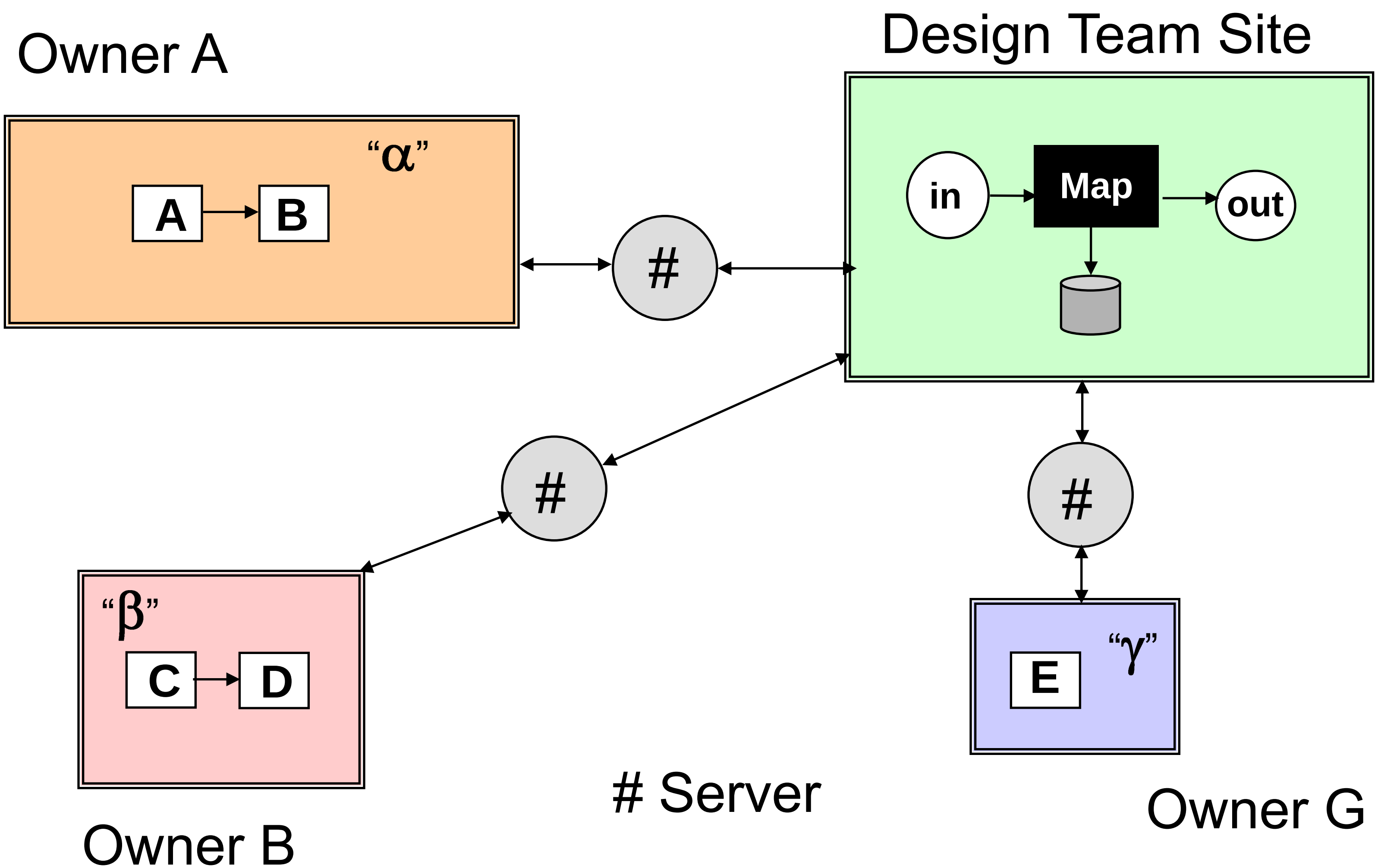


* Interface files

Implementation and Ownership (II)

16.888
ESD.77

Case 3: In a LAN or WAN environment



Modeling-Simulation Environments

- Integrated Modeling & Simulation
  - Write functions and integrate via Master script
  - MATLAB, Mathematica® are popular environments
- ICEMaker
  - Developed at Caltech/JPL
  - linked spreadsheets (client server)
- DOME (MIT) - CO (Oculus)
  - DOME based peer-peer system
  - API's into numerous Engineering applications
- FIPER (Simulia – Dassault Systems)
  - Client-server enterprise system
  - Targeted at the corporate environment
- PHX Model Center
  - Phoenix Integration – Flagship Product
  - Desktop Integration Environment

Lecture Summary

- Follow a logical model & simulation development process, don't forget benchmarking
- Decomposition is crucial in order to facilitate code integration and coupling
- N^2/DSM Matrix is useful tool to organize data
- Minimize the number of feedback loops

N^2 and DSM References

- Rogers, James L.: DeMAID/GA User's Guide - Design Manager's Aid for Intelligent Decomposition with a Genetic Algorithm, April 1996, NASA TM – 110241.
- Steward, D.V., 1981, *Systems Analysis and Management: Structure, Strategy, and Design*, New York: Petrocelli.
- *D.V. Steward. "Partitioning and Tearing Systems of Equations", SIAM Journal of Numerical Analysis. Ser.B, vol.2, no.2, 1965, pp.345-365*
- de Weck, O. L. and Chang D., "Architecture Trade Methodology for LEO Personal Communication Systems ", 20th International Communications Satellite Systems Conference, Paper No. AIAA-2002-1866, Montréal, Québec, Canada, May 12-15, 2002
- Ulrich, K.T., and S.D. Eppinger, 1995, *Product Design and Development* , McGraw-Hill.
- The Design Structure Matrix Website, http://www.dsmweb.org/

ESD.77 / 16.888 Multidisciplinary System Design Optimization
Spring 2010