

Modelling and Information Entropy of Design Spaces

Petter Krus

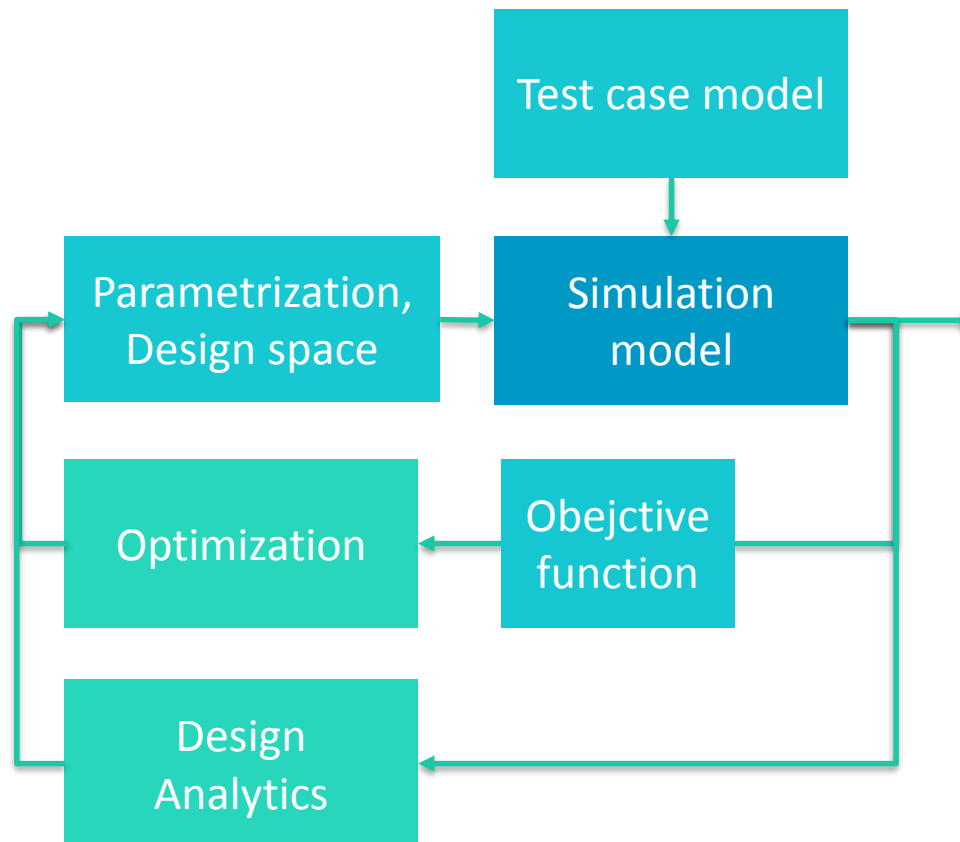
Department of Management and Engineering

Linköping University

Extended System Simulation

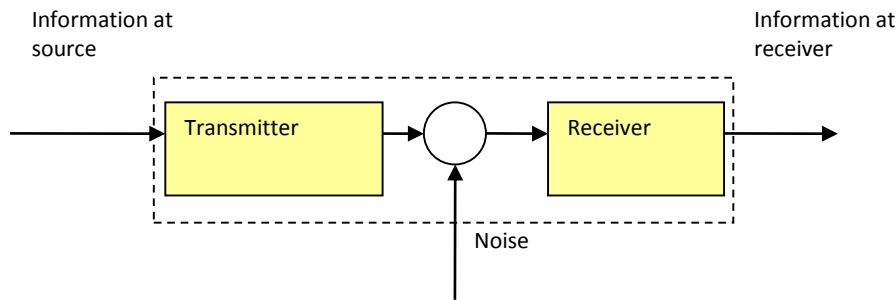
- Connectivity, co-simulation, multi-core, FMU etc.
- Simulation based optimization
- Design analytics
 - I.e. sensitivity analysis, correlation analysis, robustness, **complexity metrics**, etc.
 - Methods for experimental validation
- Parametrization for design.
 - **Analytic parametrization**, and reduction
- Test case modelling and management

Extended System Simulation



Information Theory

- Claude Shannons ground breaking work.



<https://youtu.be/R4OIXb9aTvQ>

Reprinted with corrections from *The Bell System Technical Journal*,
Vol. 27, pp. 379-623, 623-650, July, October, 1948.

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist¹ and Hartley² on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.
2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measure entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.
3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits*, a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\begin{aligned}\log_2 M &= \log_{10} M / \log_{10} 2 \\ &= 3.321928 \log_{10} M\end{aligned}$$

¹Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A.I.E.E. Trans.*, v. 47, April 1928, p. 617.
²Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal*, July 1928, p. 535.

Applications

- Product Platforms
- Complexity of Computer Programs
- Logic Hardware Design
- Human factors
- Search Theory
- Axiomatic Design
- System Design
- Optimizatoin

Approaches to Product Variety Management Assuming Configuration Conflict Problem

V. Modrak
Professor
Technical University of
Czechia
Faculty of Manufacturing Technologies

problems in mass customized manufacturing are frequent and analyzed in order to find suitable methods to optimize customer varieties. Moreover, product designers have to consider also satisfaction problem, since some serious conflicts may occur. Improvement of the customer are specified based on a wide portfolio of components and given configurations cannot be found. The aim of this paper is to explore the possibilities of using axiomatic and entropy based measures for quantifying product variety. For this purpose, two approaches will be applied on



Available online at www.sciencedirect.com
ScienceDirect

Procedia CIRP 34 (2015) 87–92

9th International Conference on Axiomatic Design

Using Axiomatic Design and Entropy to Measure Complexity

Vladimir Modrak^{a,*}, Slavomir B.

An Entropy-Based Complexity Measure for Object-Oriented Designs

Jagdish Bansaly
Department of Mathematics and Computer Science, California State University, Hayward, CA
E-mail: jbansly@cs.uah.edu

Carl Davis, Letha Eitzorn
Computer Science Department, University of Alabama, Huntsville, AL 35899, USA
E-mail: cdavis@cs.uah.edu, letzorn@cs.uah.edu

The use of entropy as a measure of information content has led to its use in measuring the code complexity of functionally-oriented software products; however, no similar capability exists for evaluating complexities of object-oriented systems using entropy. In this paper, a new metric based on entropy as a measure of information content is proposed for object-oriented programming. In development, it is extremely important to understand the complexity of object-oriented products and the impact of this complexity on the development process.

acceptance of the object-oriented development process can be optimized and to compare different object-oriented products and the impact of this complexity on the development process. In development, it is extremely important to understand the complexity of object-oriented products and the impact of this complexity on the development process.

1. Introduction
The object-oriented software development process of the 1990s is reflected in the market in a number of ways. The growth of the software development process is reflected in the market in a number of ways. The growth of the software development process is reflected in the market in a number of ways.

Received August 1999
Received November 1999
© 1999 Elsevier B.V.

THEORY AND PRACTICE OF SOFTWARE ENGINEERING
Design, Measurement, Theory, Verification

Keywords
Complexity, Measurement, Entropy, Abstract, Model, States

Paper organization
This paper is organized as follows. Section 1 identifies a general need for a new and different measurement for complexity. The following section 2 describes the approach and the goals of the design entropy concept. Section 3 analyzes the origins for complexity and can define the formulae for the design entropy. Before section 5 presents the formulae of the concept section 4 will clarify some terminology. The final section 6 will apply the formulae on some different examples.

1. INTRODUCTION
Most engineering disciplines exert well defined methods and models to manage, control and evaluate projects. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CODING: 10.1016/j.proci.2015.08.011
CODING: 10.1016/j.proci.2015.08.011
CODING: 10.1016/j.proci.2015.08.011



An Entropy-Based Approach to Assessing Object-Oriented Software Maintainability and Degradation – A Method and Case Study

Hector M. Olague¹, Letha H. Eitzorn², Glenn Cox²
Computer Science Department
University of Alabama in Huntsville, Huntsville, AL U.S.A.

The term "software entropy" has been analogically defined to mean that software declines in quality, utility and understandability through its lifetime. While there are numerous software metrics that assess "of software maintainability, few assess software degradation at multiple discrete points in the life cycle. This paper introduces a new software degradation metric (SDM) that is based on the number of classes, methods, and attributes in the code. The SDM is a measure of software degradation that is based on the number of classes, methods, and attributes in the code. The SDM is a measure of software degradation that is based on the number of classes, methods, and attributes in the code.

Design Entropy Concept

A Measurement for Complexity

Benjamin Menhorn and Frank Slomka
Institute for Embedded Systems/Real-Time Systems
University of Ulm, Germany
benjamin.menhorn@frank.slomka@uni-ulm.de

Design Entropy Concept

A Measurement for Complexity

Benjamin Menhorn and Frank Slomka
Institute for Embedded Systems/Real-Time Systems
University of Ulm, Germany
benjamin.menhorn@frank.slomka@uni-ulm.de

ABSTRACT
In general, this work will deal with measuring complexity. The focus question is towards addressing complexity in an adequate way. This work concentrates on digital circuits and digital hardware. For this field of computer science the complexity for circuits will be calculated.

Therefore, a new complexity measure will be introduced, called design entropy. It allows a mathematical calculation of complexity resulting in figures. These allow a direct evaluation and comparison between different systems and realizations. The application and important capabilities of this measurement will be demonstrated on different examples.

Categories and Subject Descriptors
H.6.m [Hardware]: LOGIC DESIGN—Miscellaneous

General Terms
Design, Measurement, Theory, Verification

Keywords
Complexity, Measurement, Entropy, Abstract, Model, States

Paper organization
This paper is organized as follows. Section 1 identifies a general need for a new and different measurement for complexity. The following section 2 describes the approach and the goals of the design entropy concept. Section 3 analyzes the origins for complexity and can define the formulae for the design entropy. Before section 5 presents the formulae of the concept section 4 will clarify some terminology. The final section 6 will apply the formulae on some different examples.

1. INTRODUCTION
Most engineering disciplines exert well defined methods and models to manage, control and evaluate projects. The

2. DESIGN ENTROPY CONCEPT
It is important to have a model which can deal with different key aspects. For some projects key aspects are timing constraints, for others design costs and even others concern optimization. But it would not be possible to have an applicable measurement for all concerns right away. Therefore, this projects concentrates on digital circuits. Always in mind, that the developed formulae and applications methods should be also applicable within different

determination of project size makes it possible to give statements such as needed resources for development, how development processes can be optimized and to compare different projects and implementations [3] [6]. For determining sizes of projects the main challenge is to get complexity under control.

Computer science especially digital circuit design is a very young science with only a few decades of experience. Additionally computer science is subjected to fast changes and developments in technology. Therefore empirical data from recent projects is hard to transfer to new projects and brings high inaccuracies [9].

Today, most methods for estimating project size use empirical data, by analyzing previous projects [8]. They try to find key figures with which project sizes can be estimated and compared. All those approaches are basically trying to get complexity under control. But most of these approaches only work for one certain technology, programming language or description language. This indicates a need for an abstract measurement, which can be used even with changes in technology and new developments.

For digital circuits, simply counting transistors was sufficient for an adequate estimation of project size in the beginning of microelectronics. Today, with rising complexities and sizes it is not enough to count transistors anymore [5]. In hardware design it is still possible to count transistors and maybe connections. But it doesn't address complexity in an adequate way. With hardware description languages additional abstraction layers are introduced. This makes transistor counts very less significant as a complexity measurement. Hardware design today gets more and more equivalent to methods used in software engineering. If it would be possible to give funded complexity estimations, as at least represented by figures, project management models from other engineering disciplines could be used.

One way to achieve this goal, is to become independent from design methods and abstraction layers. Changes in technologies and new inventions would then still allow to use this model. And even more, a comparison between new and previous projects would still be possible. This calls for an abstract measurement.

DESIGN ENTROPY CONCEPT
Important to have a model which can deal with different key aspects. For some projects key aspects are timing constraints, for others design costs and even others concern optimization. But it would not be possible to have an applicable measurement for all concerns right away. Therefore, this projects concentrates on digital circuits. Always in mind, that the developed formulae and applications methods should be also applicable within different

It is important to have a model which can deal with different key aspects. For some projects key aspects are timing constraints, for others design costs and even others concern optimization. But it would not be possible to have an applicable measurement for all concerns right away. Therefore, this projects concentrates on digital circuits. Always in mind, that the developed formulae and applications methods should be also applicable within different

It is important to have a model which can deal with different key aspects. For some projects key aspects are timing constraints, for others design costs and even others concern optimization. But it would not be possible to have an applicable measurement for all concerns right away. Therefore, this projects concentrates on digital circuits. Always in mind, that the developed formulae and applications methods should be also applicable within different

It is important to have a model which can deal with different key aspects. For some projects key aspects are timing constraints, for others design costs and even others concern optimization. But it would not be possible to have an applicable measurement for all concerns right away. Therefore, this projects concentrates on digital circuits. Always in mind, that the developed formulae and applications methods should be also applicable within different

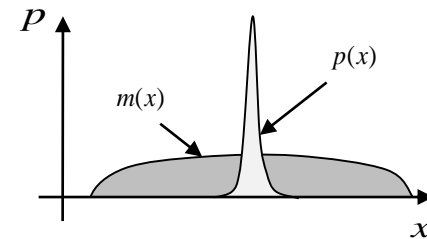
Amount of Information content (Information Entropy)

The differential information entropy for continuous signals, defined by Shannon [1] as:

$$H = - \int_{-\infty}^{\infty} p(x) \log_2(p(x)) dx$$

Kullback-Leibler divergence

$$H_{rel} = - \int_{-\infty}^{\infty} p(x) \log_2\left(\frac{p(x)}{m(x)}\right) dx$$



Generalized

$$H_{rel} = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x_1 \text{ K } x_n) \log_2\left(\frac{p(x_1 \text{ K } x_n)}{m(x_1 \text{ K } x_n)}\right) dx_1 \text{ L } dx_n$$

Amount of Information content (Information Entropy)

If the distribution $m(x)$ is a rectangular distribution in the bounded interval.

$$I_x = H_{rel}(x) = - \int_{x_{\min}}^{x_{\max}} p(x) \log_2(p(x)x_R) dx$$

Generalized

$$I_x = - \int_{x_{1,\min}}^{x_{1,\max}} \dots \int_{x_{n,\min}}^{x_{n,\max}} p(x_1, \dots, x_n) \log_2(p(x_1, \dots, x_n)x_{R1} \dots x_{Rn}) dx_1 \dots dx_n$$

More compact

$$I_x = \int_D p(\mathbf{x}) \log_2(p(\mathbf{x})D) dD$$

Amount of Information content (Information Entropy)

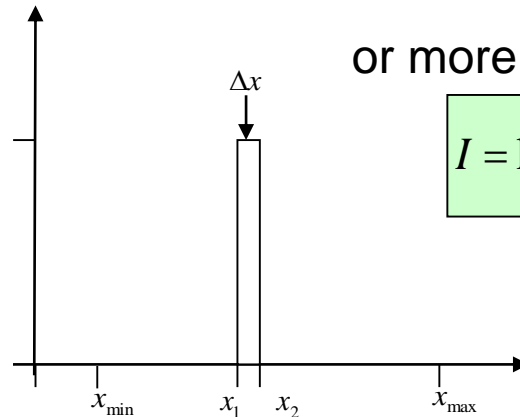
The information content I of a variable (in bits).

$$I = - \int_{x_{\min}}^{x_{\max}} p(x) \log_2 (p(x) x_R) dx \quad x_R = x_{\max} - x_{\min}$$

If the range x_r is divided in equal parts Δx the amount of information is:

$$I = \log_2 \frac{x_r}{\Delta x} = \log_2 \frac{1}{\delta_x}$$

Here Δx is the tolerance in x .



or more general:

$$I = \log_2 \frac{S_0}{s}$$

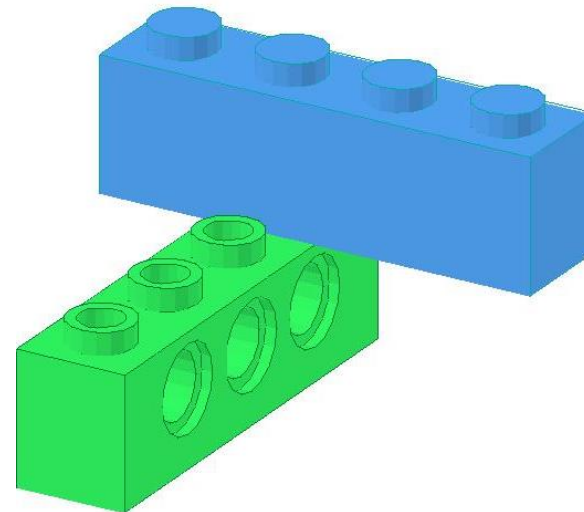
The choice of logarithm as a base (Shannon)

- It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.
- It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measure entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.
- It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

Design Information Entropy is a Measure of the Size of the Design Space

Lego example

- The design space of a set of Lego bricks represents all combinations of arranging these bricks.
- With a set of only two bricks with four knobs on each there are 51 discrete possible arrangements
- Two of these represents picking only one brick. And one state is to pick no one.
- The 51 different configuration (states) means that the amount of information needed to specify a particular design is:



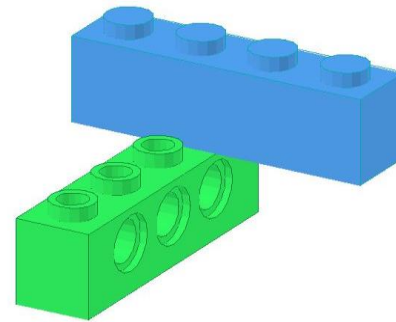
$$I_x = \log_2 n_{Dstate} = \log_2 51 = 5.7\text{bits}$$

Design Information Entropy is a Measure of the Size of the Design Space

Lego example

- The design space of a set of Lego bricks represents all combinations of arranging these bricks.
- With a set of only two bricks with four knobs on each there are 51 discrete possible arrangements
- Two of these represents picking only one brick. And one state is to pick no one.
- The 51 different configuration (states) means that the amount of information needed to specify a particular design is:

$$I_x = \log_2 n_{Dstate} = \log_2 51 = 5.7\text{bits}$$



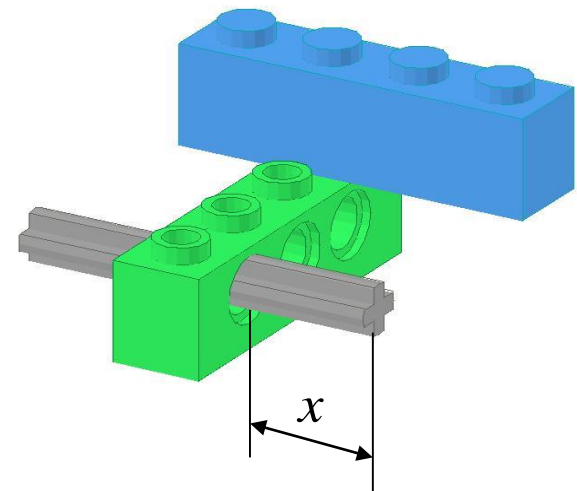
Design Space with Both Continuous and Discrete Variables

- The position of the inserted axis represents a continuous variables
- The information entropy associated with that is dependent on the accuracy with which it is specified.

$$I_x = \log_2 n_{Dstates} + \log n_{Cstates} + \log_2 \frac{x_R}{\Delta x}$$

- The axis can be in three position and If the position of the axis within one hole is specified within 10% the total information entropy is:

$$I_x = \log_2 (51 + 3) + \log_2 \frac{1}{0.1} = 8.2\text{bits}$$



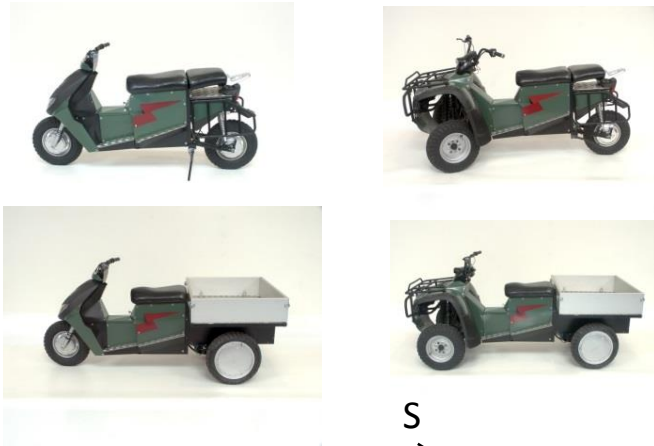
Information entropy in modular design



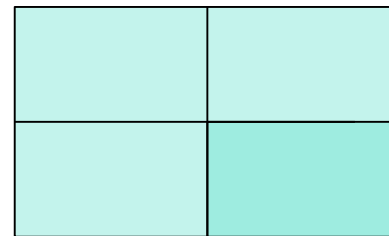
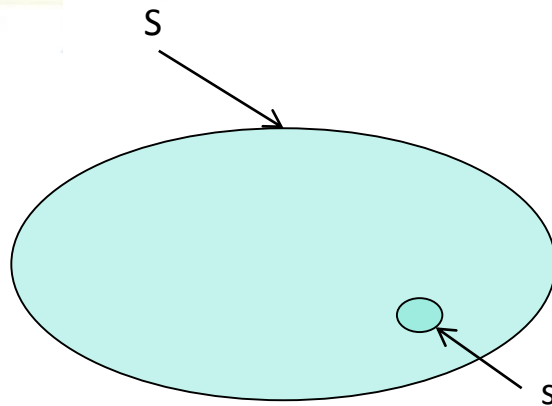
Valid design	Invalid design
Valid design	Chosen design

Design information entropy can be used as a measure of quality of product platforms for modular design. A good product platform should have little “waste” of design space.

Design Information



- Design information entropy
- The amount of bits needed to specify a design within a design space.
- To specify one designs of the four takes:



$$H_x = \log_2 \frac{S}{s} = \log_2 n_s = \log_2 (2 \times 1 \times 2) = 2 \text{ bits}$$

Design Information

- To specify one design of the four takes:

$$H_x = \log_2 \frac{S}{s} = \log_2 n_s = \log_2 (2 \times 1 \times 2) = 2 \text{ bits}$$

- The entropy of the constraint design space is

$$H_c = \log_2 n_v = \log_2 3 = 1.58$$

Valid design	Invalid design
Valid design	Chosen design

← waste

- Wasted design information entropy can be formulated as:

$$H_w = H_x - H_c = -\log_2 \frac{S_x / s}{S_c / s} = -\log_2 \frac{S_x}{S_c} = 0.42$$

Morphological Matrix for concept selection

$$N_s = \prod_{i=1}^{n_f} n_{m,i}$$

$$N_s = \prod_{i=1}^4 3 = 3^4 = 81$$

$$H_x = \log_2 N_s = \log_2 81 = 6.34 \text{ bits}$$

Increasing the number of rows by one:

$$N_s = \prod_{i=1}^5 3 = 3^5 = 243$$

$$H_x = \log_2 N_s = \log_2 243 = 7.92 \text{ bits}$$

The entropy increases linearly
with number of rows.

	m ¹	m ²	m ³
f ₁	m ₁ ¹	m ₁ ²	m ₁ ³
f ₂	m ₂ ¹	m ₂ ²	m ₂ ³
f ₃	m ₃ ¹	m ₃ ²	m ₃ ³
f ₄	m ₄ ¹	m ₄ ²	m ₄ ³

	m ¹	m ²	m ³
f ₁	m ₁ ¹	m ₁ ²	m ₁ ³
f ₂	m ₂ ¹	m ₂ ²	m ₂ ³
f ₃	m ₃ ¹	m ₃ ²	m ₃ ³
f ₄	m ₄ ¹	m ₄ ²	m ₄ ³
f ₅	m ₅ ¹	m ₅ ²	m ₅ ³

Information Entropy gives a measure of complexity more
consistent with experience!

Information Entropy of Morphological Matrix

In the general case there can be variable number of elements in each row.

$$N_s = \prod_{i=1}^4 4 \times 2 \times 3 \times 2 \times 3 = 144$$

$$H_x = \log_2 N_s = \log_2 144 = 7.16 \text{ bits}$$

	m ¹	m ²	m ³	m ⁴
f ₁	m ₁ ¹	m ₁ ²	m ₁ ³	m ₁ ³
f ₂	m ₂ ¹	m ₂ ²		
f ₃	m ₃ ¹	m ₃ ²	m ₃ ³	
f ₄	m ₄ ¹	m ₄ ²		
f ₅	m ₅ ¹	m ₅ ²	m ₅ ³	

Information Entropy and Complexity

According to Axiomatic Design the best designs are uncoupled

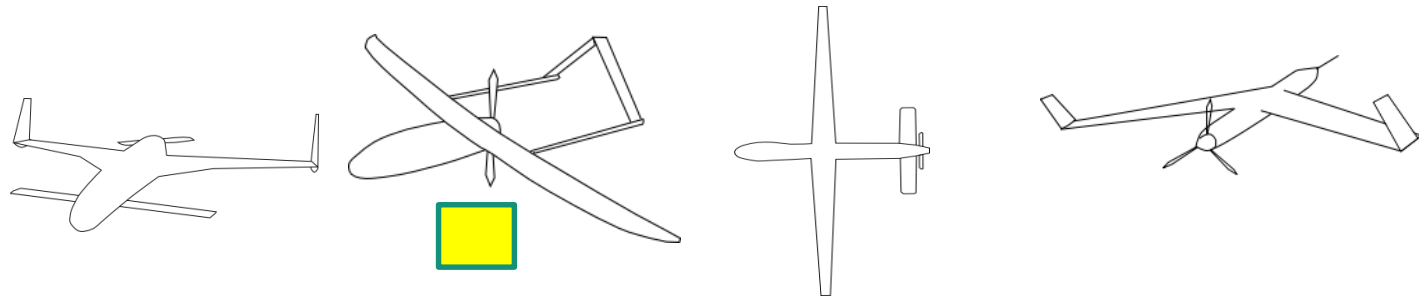
$$\begin{pmatrix} FR_1 \\ FR_2 \end{pmatrix} = \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix} \begin{pmatrix} DP_1 \\ DP_2 \end{pmatrix}$$

Functional
requirements
(Anatomy)

Design
parameters
(Architecture)

If this is true. Design decision becomes independent of each other

Example: UAV Aircraft Concept Generation



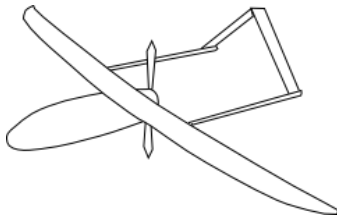
Design elements	Alternative solutions				
Horizontal stabilization	Front (canard)	Aft	Aft tail integrated	Wing integrated	
Vertical stabilization	Central	Wing tip	Aft tail integrated	Upper	Lower
Tail mount	Single fuselage	Twin boom			
Propulsion	Tractor	Pusher			

$$N_s = \prod_{i=1}^{n_f} n_{m,i} \quad n_s = 4 \times 5 \times 2 \times 2 = 80$$

$$I_x = -\log_2 \frac{1}{N_s} = -\log_2 \frac{1}{80} = 6.32 \text{ bit}$$

Aircraft Optimization

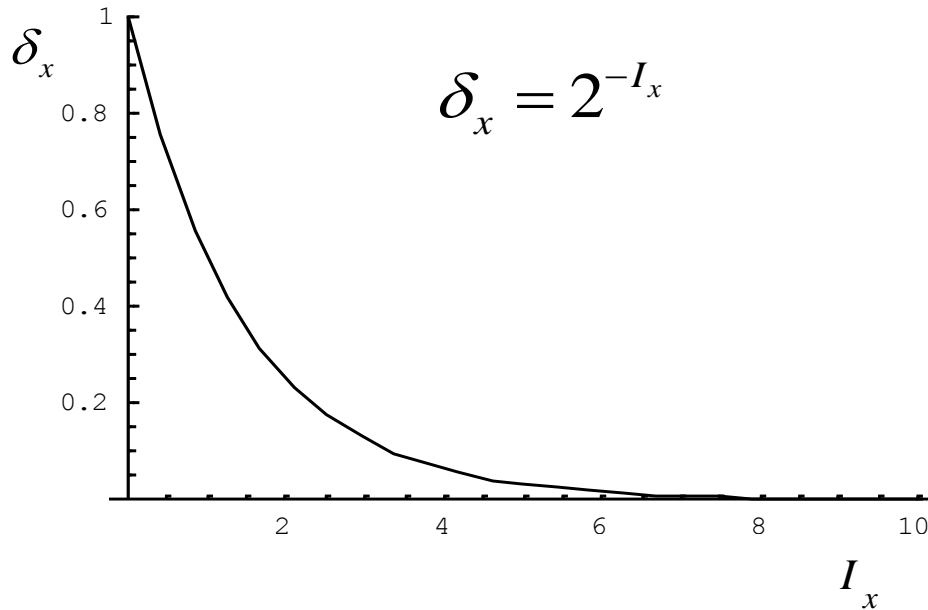
- For the aircraft example typical design parameters would be;
 - wing span, root cord, tapering, thickness, and sweep, structural weight, fuel weight, engine size, wing position, span of horizontal tail, cruise speed.



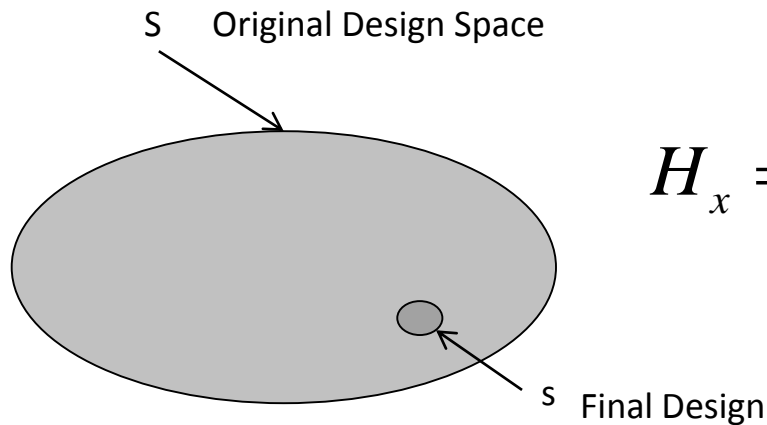
Design uncertainty as a function of design information

$$I_x = \log_2 \frac{x_R}{\Delta x} = \log_2 \delta_x$$

$$\delta_x = 2^{-I_x}$$

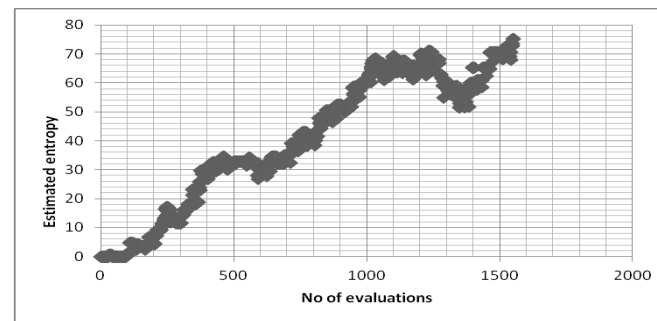


Information entropy of design s relative to design space S



$$H_x = \log_2 \frac{S}{s}$$

Information entropy to measure convergence in optimization



Information increase in Optimization

Information entropy is estimated as

$$\hat{H}_x = -n \log_2 \left(\max \left(\delta_{x,i} \right) \right) \quad \delta_{x,i} = \frac{x_{i,\max} - x_{i,\min}}{x_{0,i,\max} - x_{0,i,\min}}$$

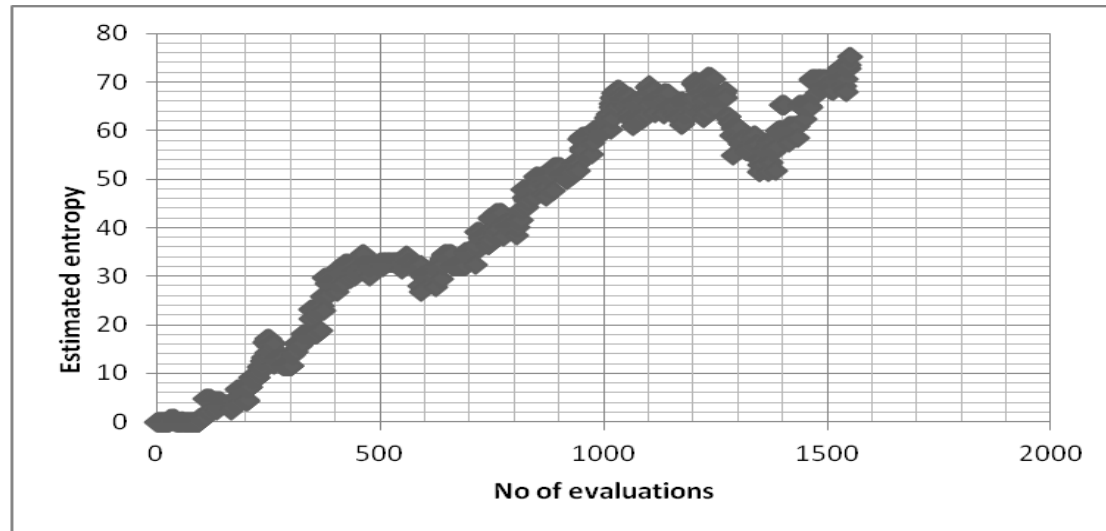


Figure 5. Accumulation of information as a function of number of objective function evaluations

Meta object function

I expresses the total uncertainty, representing the sum of

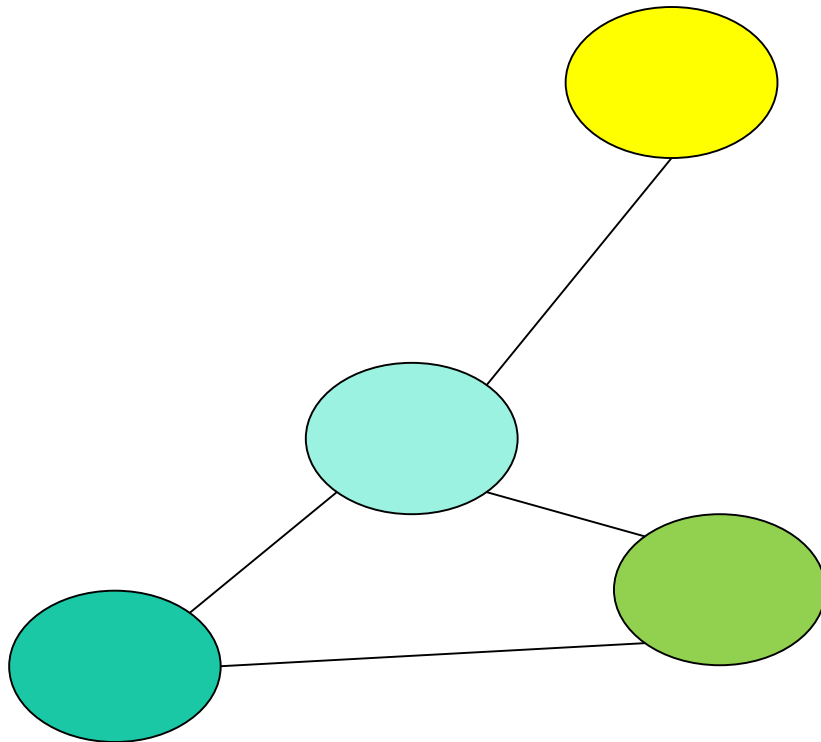
uncertainty in location and

uncertainty of success

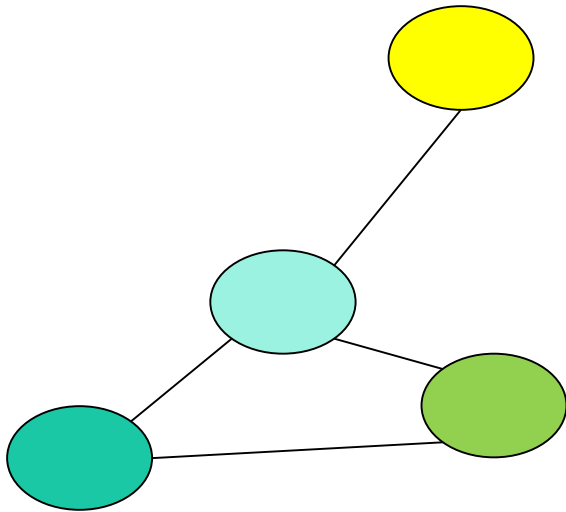
$$I_{tot} = (1 - P_{opt}) \log_2 \left(\frac{1 - P_{opt}}{1 - \epsilon_x^n} \right) + P_{opt} \log_2 \left(\frac{P_{opt}}{\epsilon_x^n} \right)$$

$$\phi^{(2)} = \frac{I_x}{N_m} = \frac{1}{N_m} \left((1 - P_{opt}) \log_2 \left(\frac{1 - P_{opt}}{1 - \epsilon_x^n} \right) + P_{opt} \log_2 \left(\frac{P_{opt}}{\epsilon_x^n} \right) \right)$$

Information Entropy in System Design



Information Entropy in System Design



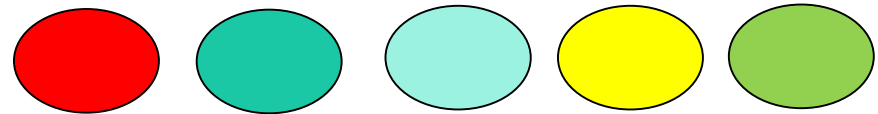
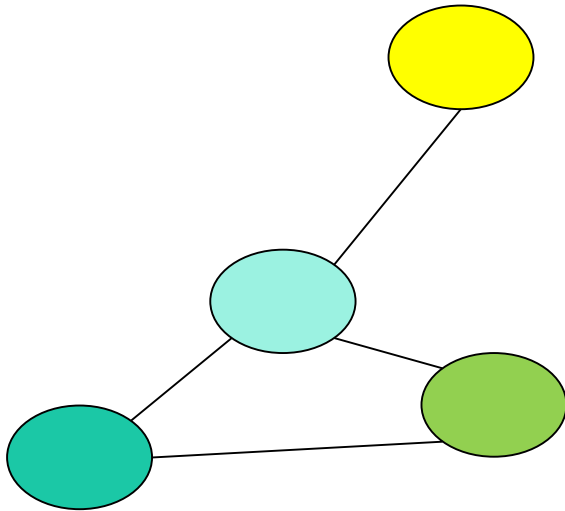
A system configuration is defined by its components and by how they are connected. The design space can thus be expressed as:

$$N_x = N_s \times N_p \quad (9.35)$$

Here N_s is the number of possibilities for component selections, and N_p the number of possible ways to connect the components. The corresponding information entropy is:

$$I_x = \log_2 N_x = \log_2 N_s + \log_2 N_p \quad (9.36)$$

Information Entropy in System Design



$$n_{s,tot} = 5$$

$$n = 4$$

$$N_s = 5^4 = 625$$

$$I_s = \log_2 N_s = 9.29\text{bit}$$

Information Entropy in System Design

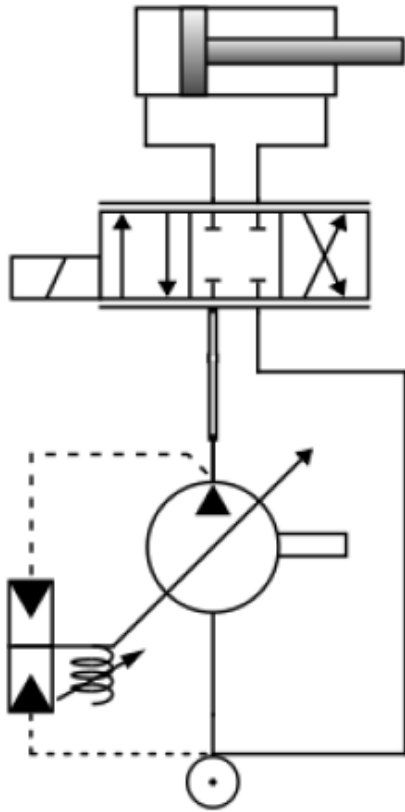


Figure 9.10: System configuration of a simple hydraulic system.

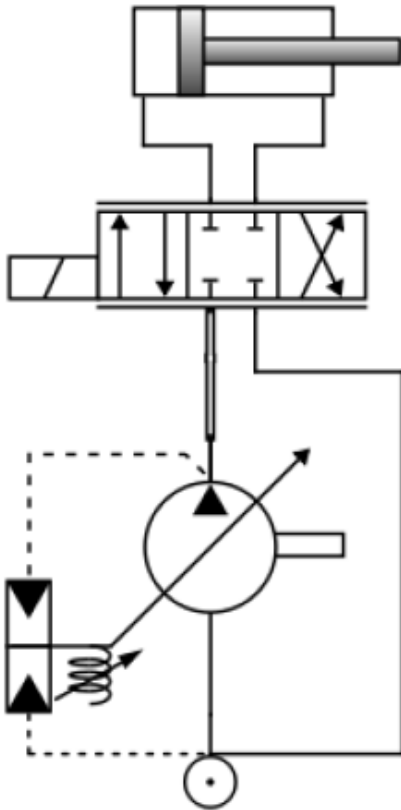
Assuming a priori information that the system should contain one cylinder one pump and one tank and a library with variants of these

$$N_s = N_{cyl} \times N_{valve} \times N_{pump} \times N_{tank} = 6 \times 27 \times 6 \times 1 = 972 \quad (9.43)$$

and hence

$$I_s = \log_2 N_s = \log_2 972 = 9.92 \quad (9.44)$$

Connectivity



System components	Connectors	A	B	A	B	P	R	P	R	R
Piston	A	■	■							
	B	■	■							
Servo valve	A	1		■	■	■	■			
	B		1	■	■	■	■			
	P			■	■	■	■			
	R			■	■	■	■			
Pump	P					1		■	■	
	R							■	■	
Tank	R						1		1	■

$$I_p = \frac{n_p \times n_p}{2} - n_p \quad (9.45)$$


Here n_p is the total number of ports in the system. For this example it is:

$$I_p = \frac{10 \times 10}{2} - 10 = 40 \quad (9.46)$$

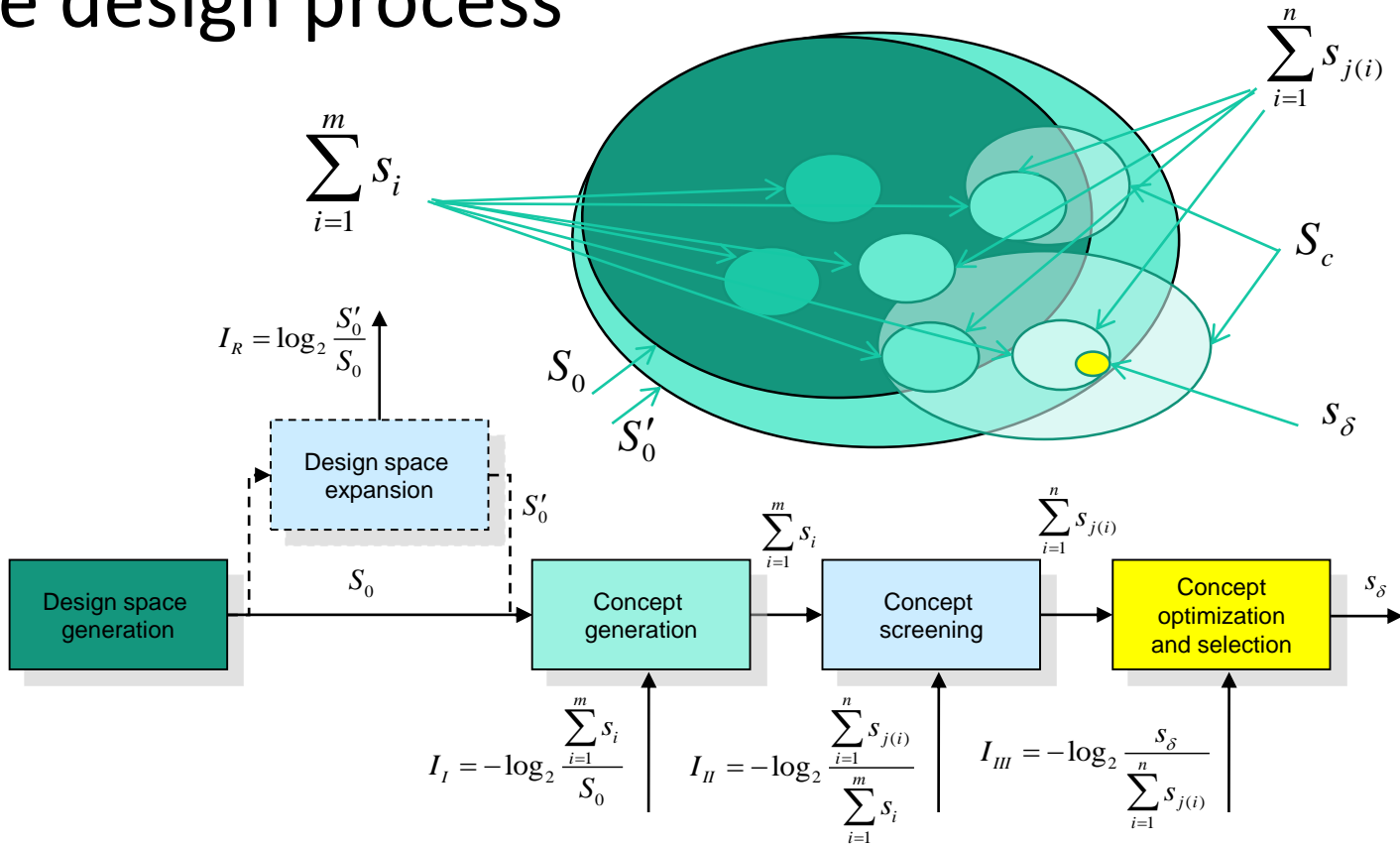
The total amount of information is:

$$I_x = I_s + I_p \quad (9.47)$$

$$I_x = 9.92 + 40 = 49.92\text{bits} \quad (9.48)$$

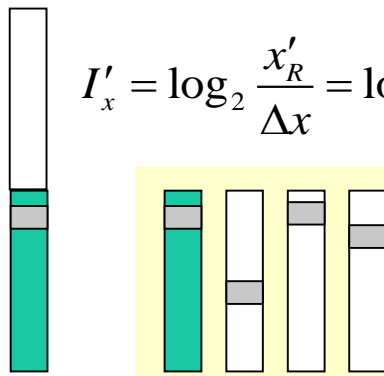
 Figure 9.11: Connectivity matrix of a hydraulic servo.

Growth of design information entropy during the design process



Design space expansion

- The design information entropy can be increased in two ways
 - Refinement
 - Design space expansion
- Design space can be increased in several ways like:
 - Adding more bricks
 - Adding other types of bricks
 - Releasing more design parameters in a design



The diagram shows a single tall bar on the left, representing a design space with a large range Δx . To its right, a yellow box contains four shorter bars of equal height, representing a design space with a smaller range $\Delta x'$ but a larger number of options n . The equations show that the total information entropy I'_x is the same in both cases.

$$I'_x = \log_2 \frac{x'_R}{\Delta x} = \log_2 \left(\frac{n \times x_R}{\Delta x} \right) = \log_2 n + \log_2 \frac{x_R}{\Delta x} = \log_2 n + I_x$$

$$I'_x = \log_2 \frac{x'_R}{\Delta x'} = \log_2 \left(\frac{x_R}{\Delta x} \right)^n = n \log_2 \frac{x_R}{\Delta x} = n I_x$$

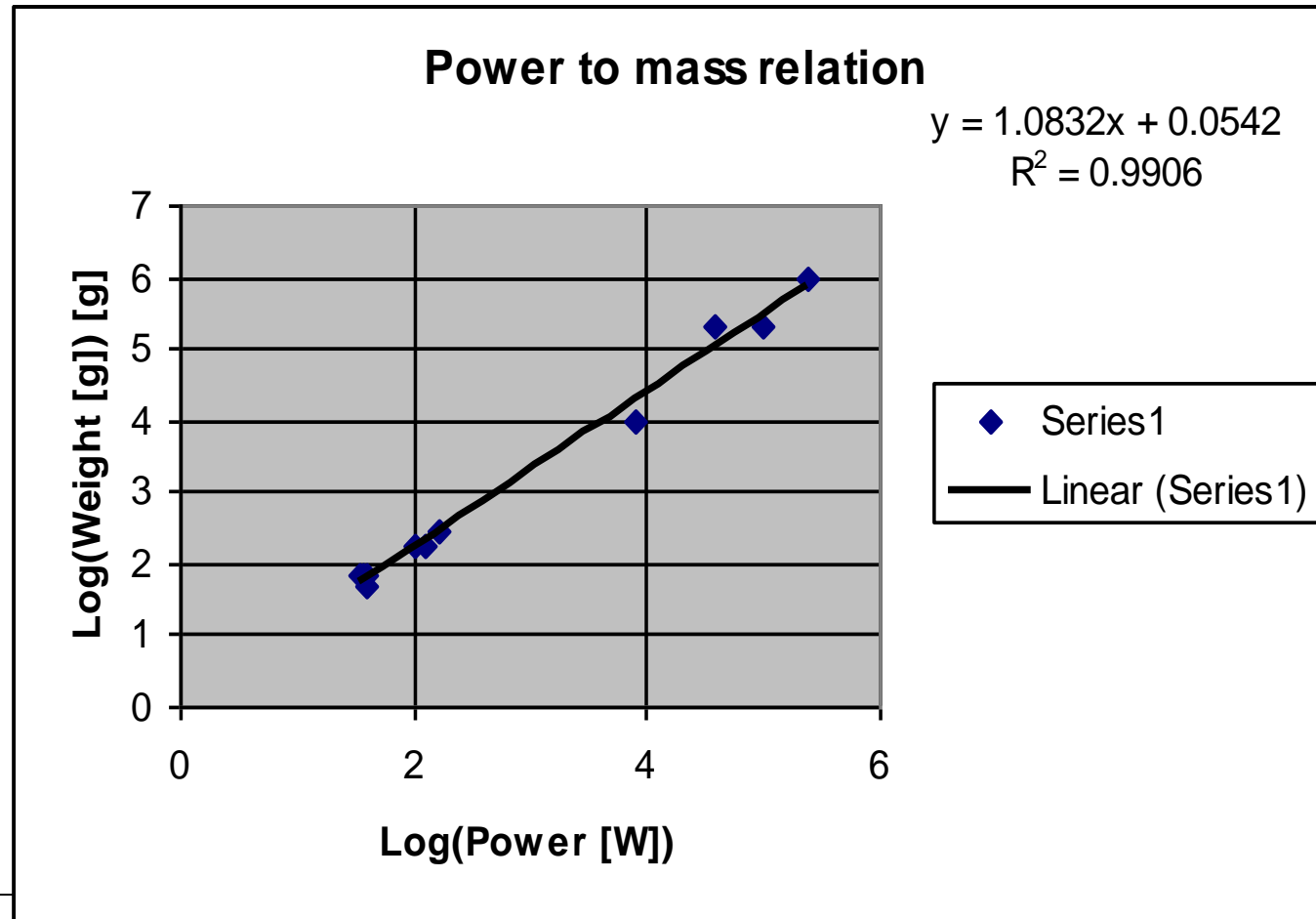
Design Space Generation and Parameter Reduction



Example: Electric Motor Data

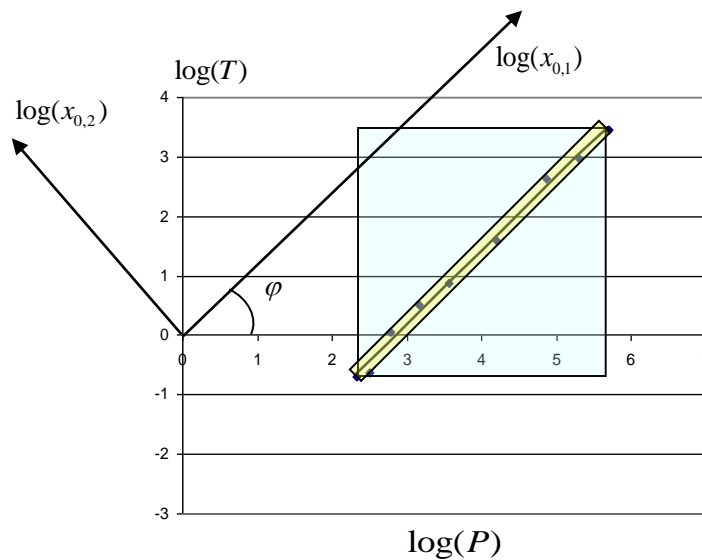
Voltage	max power	speed at load	max torque	volume	mass	power intensity k_p	mean pressure (torque density) p_m	power density ρ_p	torque intensity k_T
[V]	[W]	[rad/s]	[Nm]	[cm ³]	[kg]	[kW/kg]	[bar]	[W/cm ³]	[Nm/kg]
8.4	210	1068	0.20	55	0.1716	1.22	0.04	3.84	1.15
8	320	1378	0.23	54	0.29	1.10	0.04	5.95	0.80
24	609	523	1.164	343	1.10	0.55	0.03	1.78	1.06
24	1440	450	3.2	729	2.4	0.60	0.04	1.98	1.33
24	3580	471	7.6	729	3.9	0.92	0.10	4.91	1.95
50	15992	419	38.20	4539	9.36	1.71	0.08	3.52	4.08
460	73763	175	420.74	23487	215.00	0.34	0.18	3.14	1.96
460	198499	215	922.87	86524	215.00	0.92	0.11	2.29	4.29
460	491751	175	2813.33	165518	907.00	0.54	0.17	2.97	3.10
					Average values	0.88	0.09	3.38	2.19

Power to weight relation (electric motor)



Prinicipal Component Analysis

to minimize waste of design space (using Singular Value Decomposition, SVD)



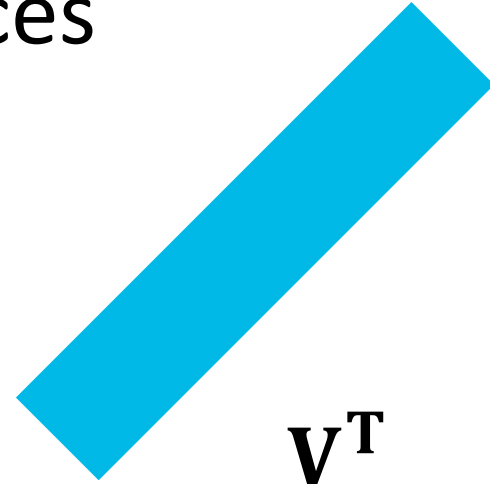
The meaning of the matrices



U



W



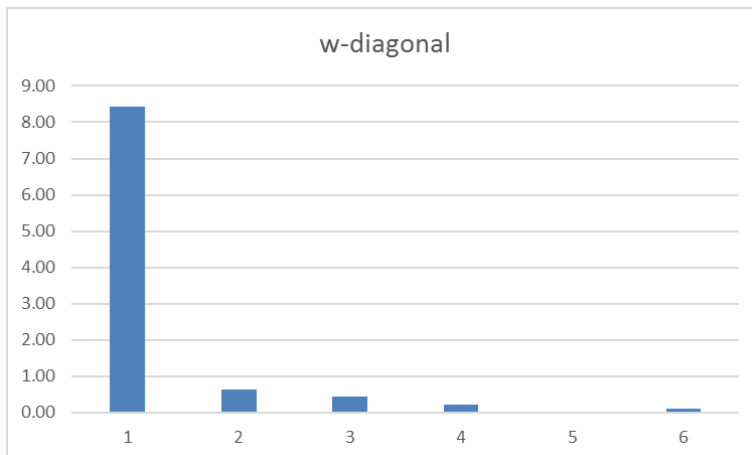
V^T

$$\mathbf{X} = \mathbf{U} \times \mathbf{W} \times \mathbf{V}^T$$

Adding logarithmic scaling and removal of offset (set mean to zero) means that limits in U on set to -1 and 1 covers the data set within one standard deviation, and thus results points likely to be feasible.

SVD model of Electric motors

	Reliance K22M13031	Estimate	Adjusted	Result	Average								SVD variables	w-diagonal	residual
Corner power [W]	491751	491744	5.69	1.99	3.70	-1.163	0.130	-0.022	0.007	0.000	-0.001	-1.66	8.43	1.86	
Speed at load [rad/s]	175	175	2.24	-0.46	2.70	0.331	0.122	-0.021	-0.031	0.000	0.005	0.39	0.62	0.20	
Max Torque [Nm]	2813	2813	3.45	2.44	1.01	-1.494	0.008	-0.001	0.038	0.000	-0.005	-0.56	0.44	0.17	
diameter [mm]	446	446	2.65	0.63	2.02	-0.384	0.005	0.064	0.007	0.000	0.028	-0.99	0.21	0.10	
volume [cm ³]	165518	165516	5.22	2.06	3.16	-1.256	-0.018	0.084	-0.036	0.000	-0.011	-1.01	0.00	0.07	
mass [kg]	907	907	2.96	2.21	0.75	-1.298	-0.079	-0.085	-0.025	0.000	0.010	1.05	0.10	0.07	

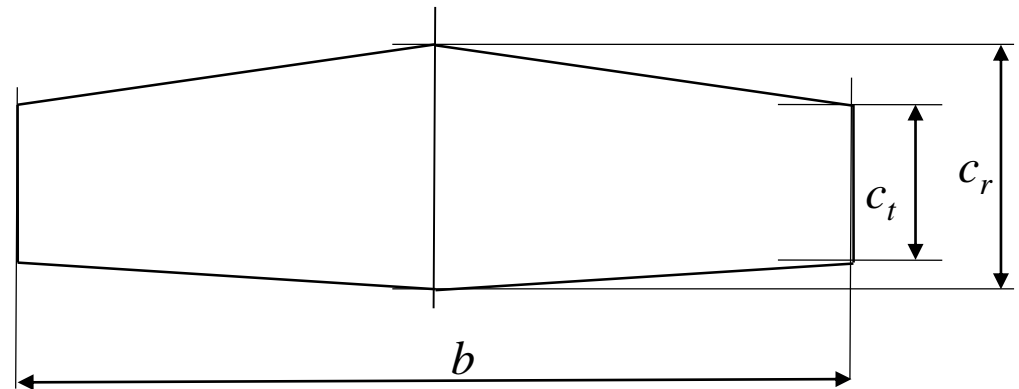


Example of Design Space for Parametrization: Aircraft Wing Planform

$$S = \frac{b}{2}(c_r + c_t)$$

$$AR = \frac{b^2}{S}$$

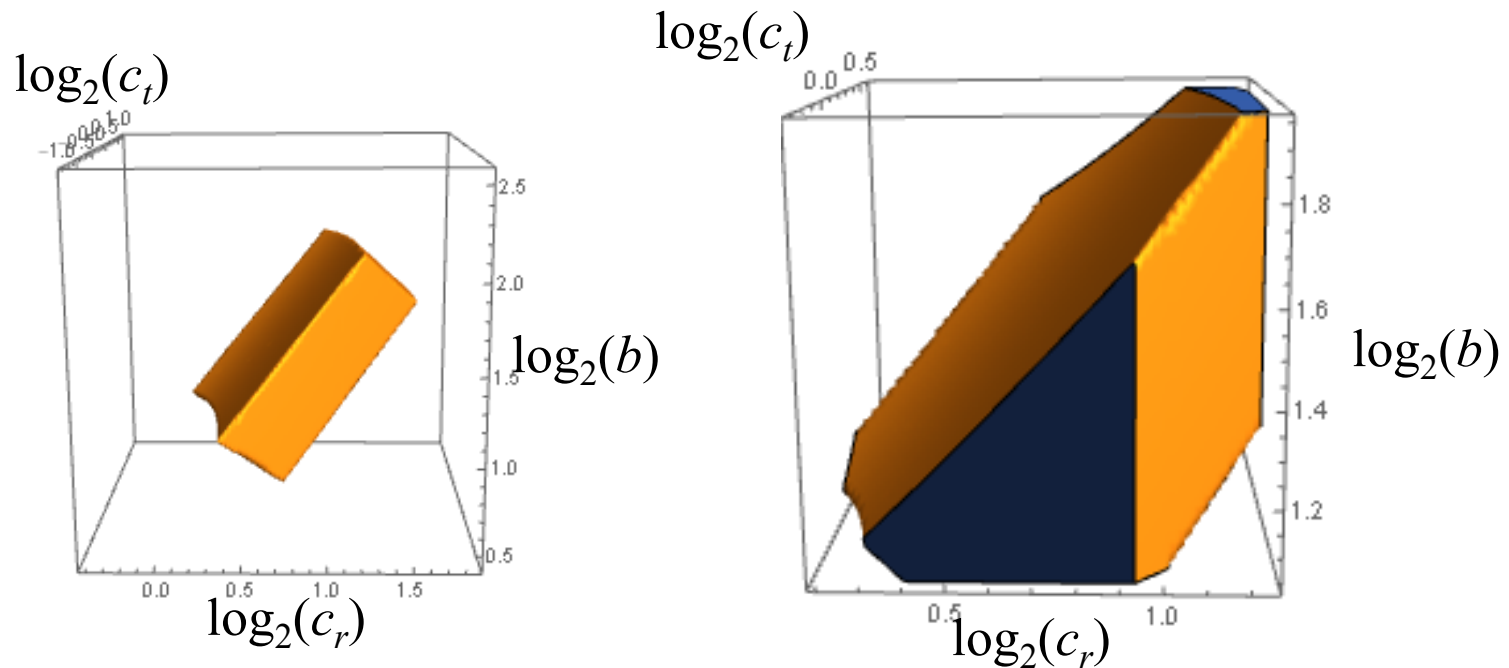
$$c_t = c_r \lambda$$



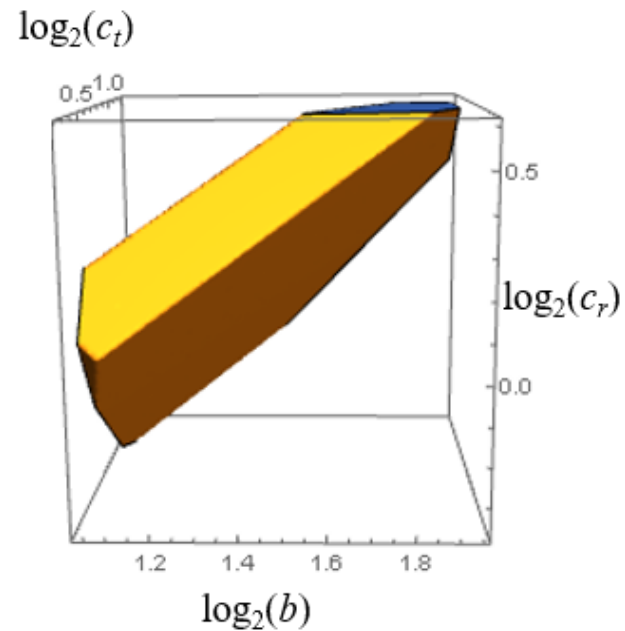
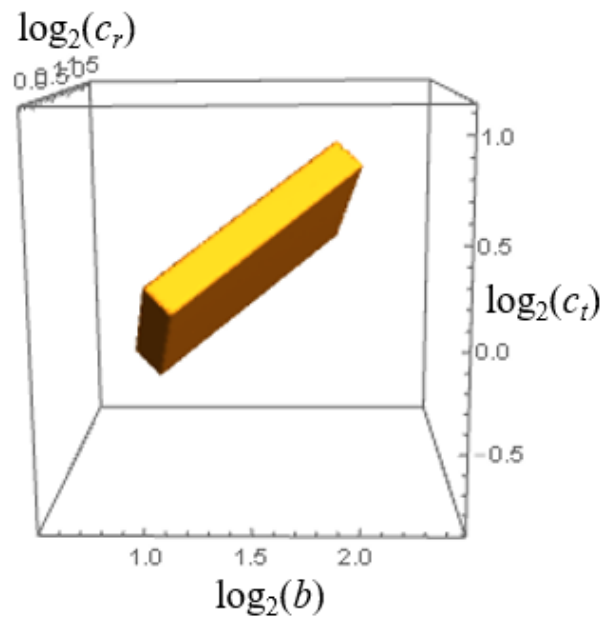
	b	c _r	c _t	S	AR	λ
AN225	88.4	16.5122	3.96293	905	8.63487	0.24
A380	79.75	17.6594	3.53187	845	7.5267	0.2
A320	34.09	5.99394	1.19879	122.6	9.47902	0.2
Gulfstream IV	23.7	5.73191	1.71957	88.3	6.36116	0.3
U2	32	4.83333	0.966667	92.8	11.0345	0.2
F-16	9.96	4.47711	1.11928	27.87	3.55944	0.25
Mirage 2000	9.13	8.5537	0.427685	41	2.0331	0.05
Cessna 172	11	1.59214	1.35332	16.2	7.46914	0.85
Max value	88.4	17.6594	3.96293	905	11.0345	0.85
Min value	9.13	1.59214	0.427685	16.2	2.0331	0.05



Design space of alternative parameters, with log-axis



Design space of SVD- parameters, with log-axis



Design Space Volumes

Parameter set	Design space volume
b, cr, ct	0.914
S,AR,I	0.877
SVD	0.25

$$H_w = H_1 - H_2 = \log_2 \frac{S_1 / s}{S_2 / s} = \log_2 \frac{S_1}{S_2} = \log_2 \frac{0.914}{0.25} = 1.87 \text{bits}$$

Conclusions

- **Design information entropy** represents a measure of the precision by which *a design* is defined relative to the ***design space*** in consideration. It is also proportional to the dimensionality of the design problem.
- Design information entropy can be used as one measure of **complexity**.
- “Thinking outside the box” is the task of finding useful directions to expand the design space.
- **Analytical parametrization** through SVD of a design can be made using sample designs to span the design space. It can also be used to produce scaling models of components. In some sense it can be regarded as the ***ideal parameter set***.