Neuro Mechanical Networks and Design Information Entropy

Petter Krus

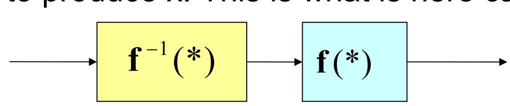
Division of Machine Design

Department of Management and Engineering

Linköping University

The Nature of Design

The design problem is then reduced to finding the inverse in order to produce x. This is what is here called *ideal design*



In the general case, however, the inverse can not be found analytically, since is nonlinear in the general case.

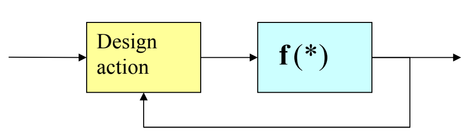
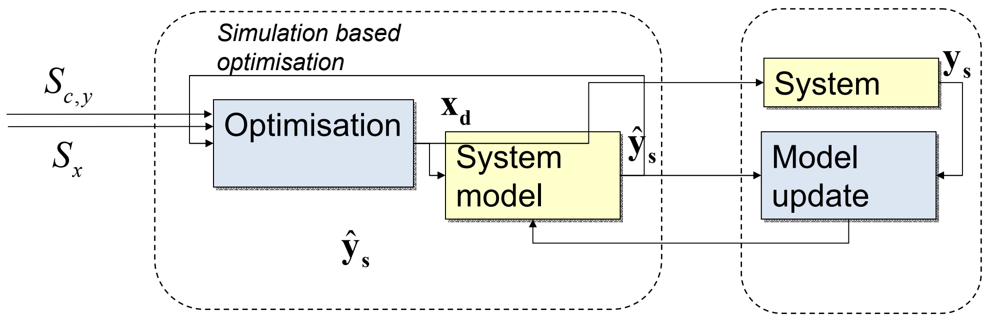


Figure 2. Iterative design

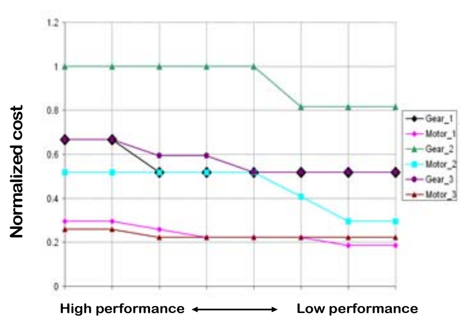
Cognitive Dynamic System?

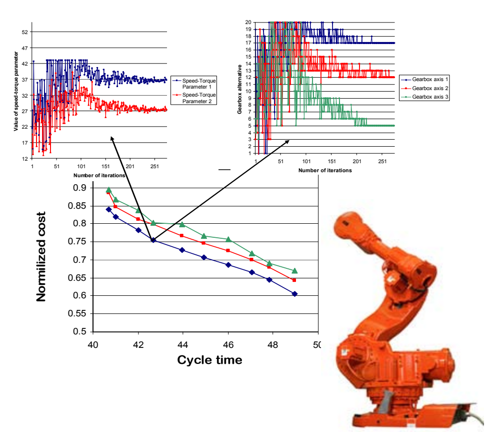


$$\mathbf{y}_\mathbf{s} = \mathbf{f}(\mathbf{x}_\mathbf{d})$$
$$\mathbf{x}_\mathbf{d} \in S_x$$
$$\mathbf{y}_\mathbf{s} \in S_{c,y}$$
$$\min f_{obj} = g(\mathbf{y}_\mathbf{s})$$

Cognitive design automation?

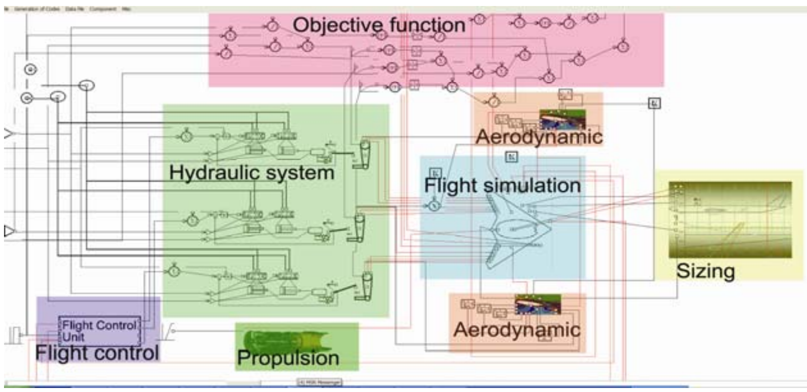
Computational engineering design. *Design Optimization In the Design of Industrial Robots at ABB*





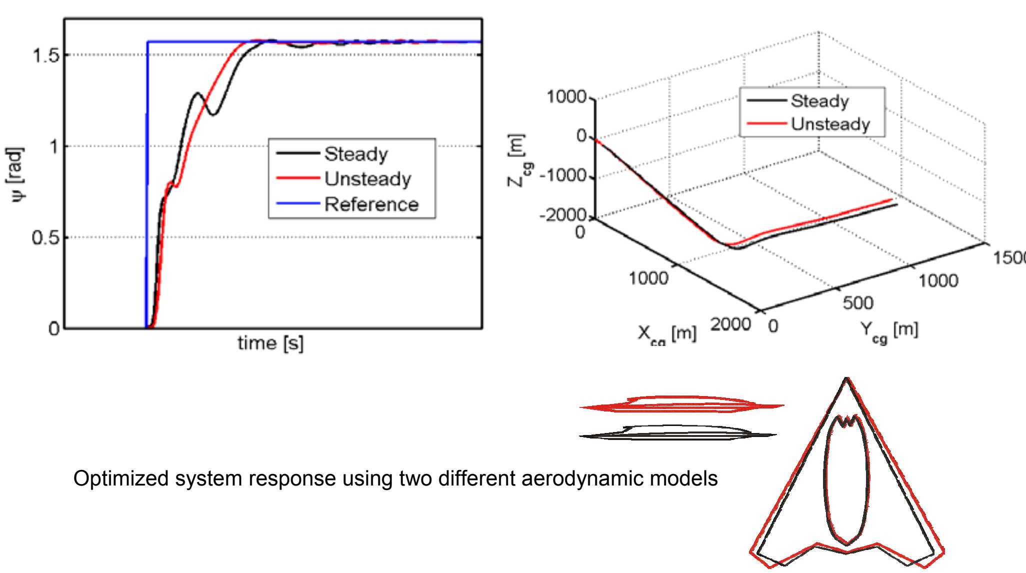
Computational engineering design *Aircraft Design Optimization*

Full system simulation and optimization



Computational engineering design *Aircraft Design Optimization*

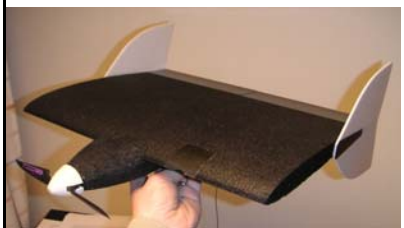
Full system simulation and optimization



Optimized system response using two different aerodynamic models

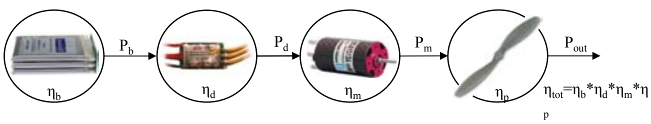
Computational engineering design. *Design Automation.*

Design on demand, Micro Aerial Vehicles, MAV

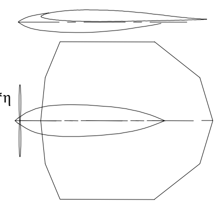


MAV baseline

	Baseline	**Optimized**
motor	Astroflight 010	Hacker B20 31S
propeller	Graupner 6x3 folding	APC 5x5
motor controller	Astroflight 10	YGE 4
Battery	Etec 1200 2s1p	Tanic 1100 3s1p
S	0.12 m2	0.12
AR	1.33	1.33
Endurance at cruise	**22 min**	**42 min**
Max speed	75 km/h	
T/W ratio	0.95	1.2



MAV propulsion system.



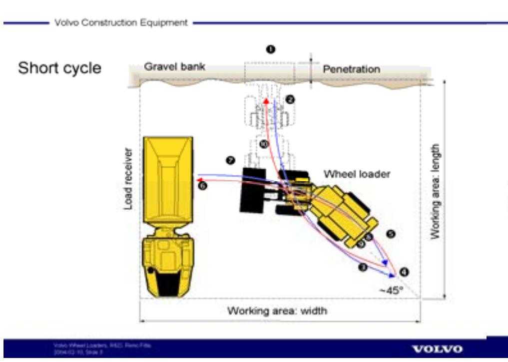
MAV optimized geometry

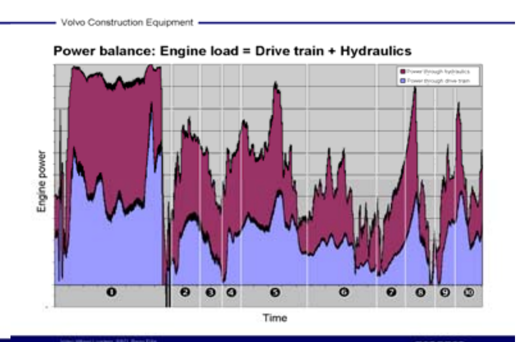
Result: *Shopping list and geometrical dimensions*

David Lundström, Petter Krus

Computational engineering design, *Harmonic Machine, VCE*

Computational design tools	Modelling and Simulation
Design Optimization Matlab	System modelling in Adams and Simulink

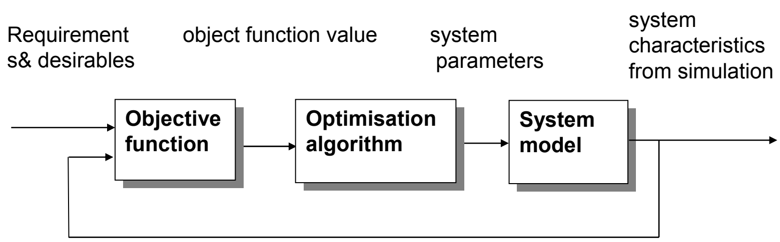




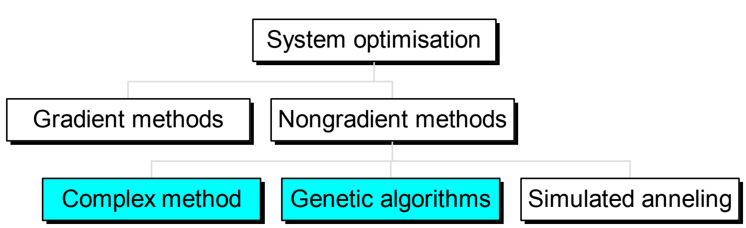
Includes human operator model

Petter Krus SECD

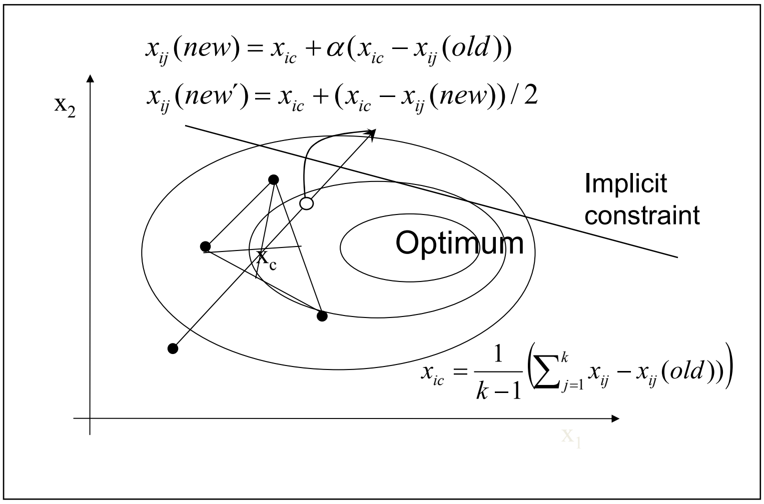
Simulation based optimisation for design



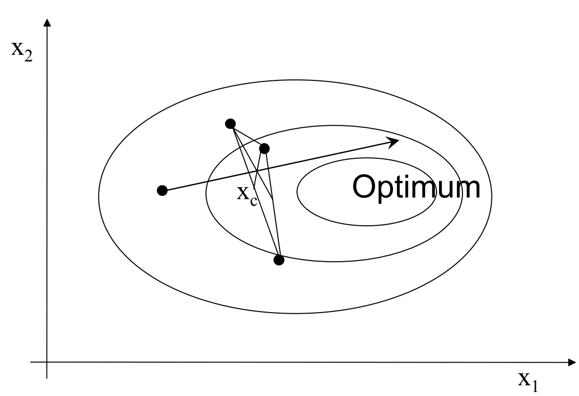
Optimisation methods



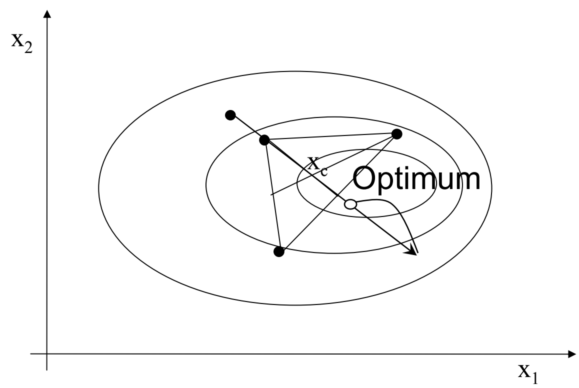
The Complex method (Constrained simplex)



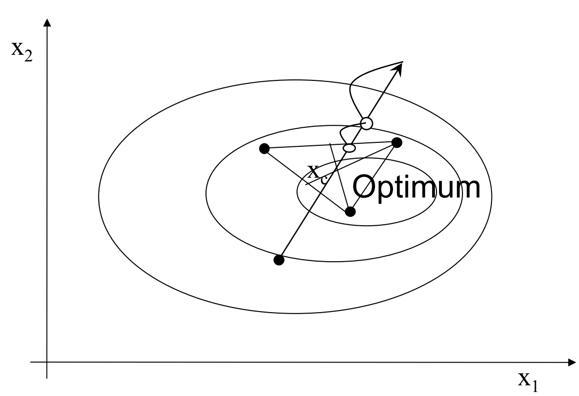
The Complex method



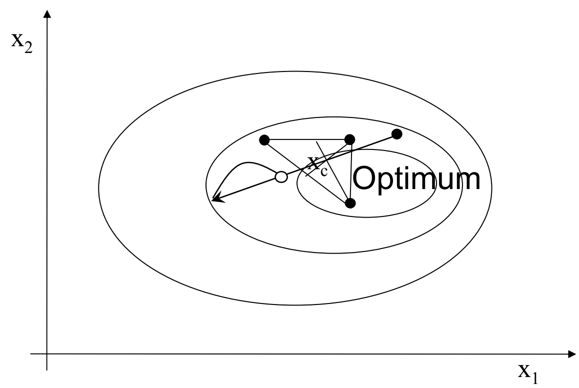
The Complex method



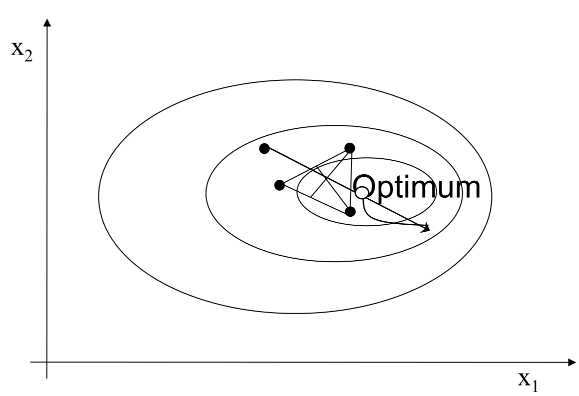
The Complex method



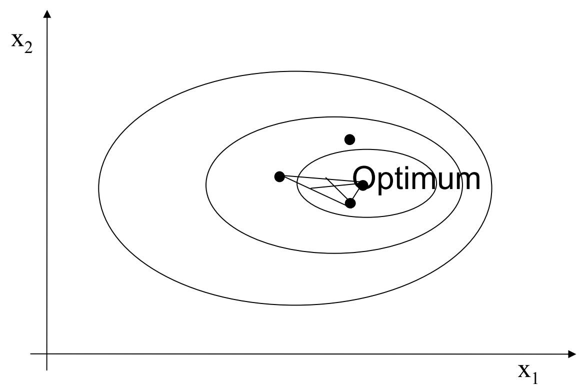
The Complex method



The Complex method

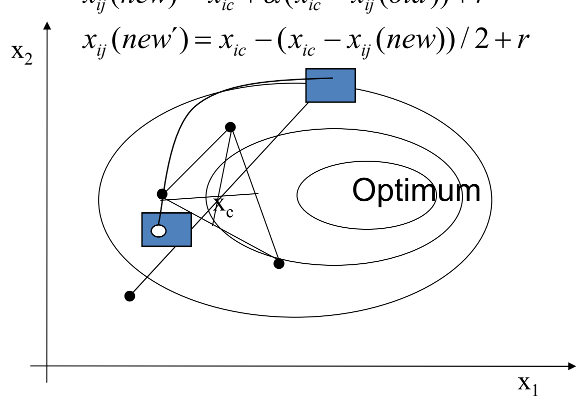


The Complex method



The Complex-R method

$$x_{ij}(new) = x_{ic} + \alpha(x_{ic} - x_{ij}(old)) + r$$
$$x_{ij}(new') = x_{ic} - (x_{ic} - x_{ij}(new))/2 + r$$



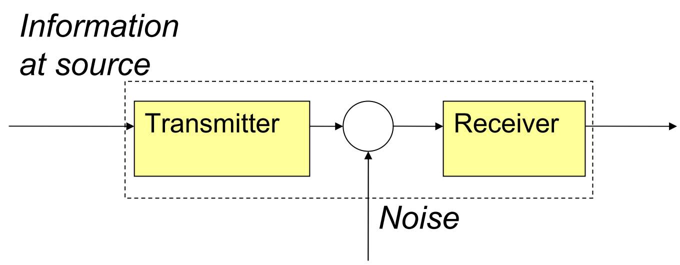
The Complex-RF

- In order to track dynamic object functions it is necessary to introduce a forgetting factor
- This is done by decreasing an object function value each time it is not re-evaluated
- This is also an advantage in “tricky” non-dynamic object functions such as noisy and discrete functions

Information theory and design

- Both design and design optimization is about increasing information about the product/system
- Information theory was introduced by Shannon in 1947
- Information entropy is a measure of level of uncertainty. High entropy->Low uncertainty

A general communication system



Amount of Information content (Information Entropy)

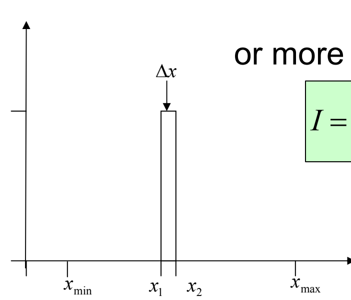
The information content *I* of a variable (in bits).

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

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

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

Here Δx is the tolerance in *x*.

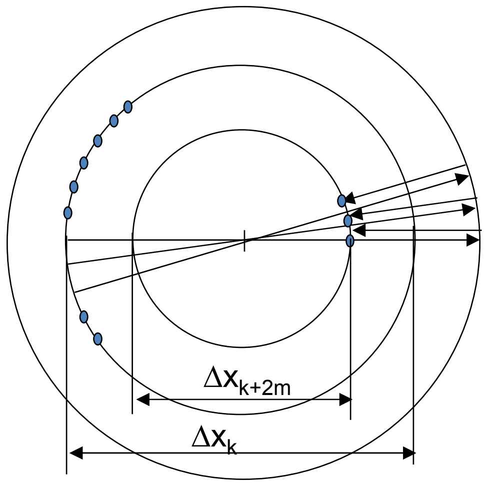


or more general:

$$\boxed{I = \log_2 \frac{S}{s}}$$

Complex convergence rate

The average degree of contraction in each step



$$\frac{\Delta x_{k+1}}{\Delta x_k} = \left(\frac{\alpha}{2}\right)^{\frac{1}{2m}}$$

$$\frac{\Delta x_{k+1}}{\Delta x_k} = \left(\frac{\alpha}{2}\right)^{\frac{1}{2\kappa n}}$$

$$m = \kappa n$$

Complex contraction rate

The average degree of contraction in each step of optimization is:

$$\frac{\Delta x_{k+1}}{\Delta x_k} = \left(\frac{\alpha}{2}\right)^{\frac{1}{2\kappa n}} \qquad m = \kappa n$$

The increase in information in each step is: (Times n because there are n parameters that are gaining information)

$$\Delta I = -n\log_2 \frac{\Delta x_{k+1}}{\Delta x_k} = -n\log_2 \left(\frac{\alpha}{2}\right)^{\frac{1}{2\kappa n}} = -\log_2 \left(\frac{\alpha}{2}\right)^{\frac{1}{2\kappa}}$$

Complex contraction

The increase in information *I* in each step

$$\Delta I = -\log_2 \left(\frac{\alpha}{2}\right)^{\frac{1}{2\kappa}}$$

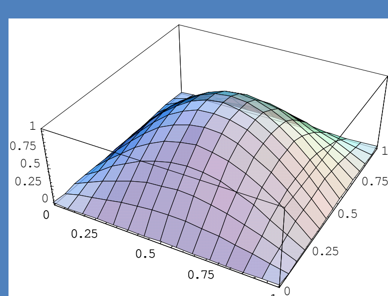
Example: α=1.3, κ=2 yields

$$\Delta I = -\log_2 \left(\frac{1.3}{2}\right)^{\frac{1}{2*2}} = 0.155$$

The amount of information gained in each step is
ΔI = 0.155 bits/evaluation

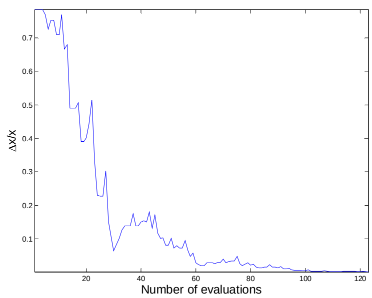
Test function: Unimodal hump

$$f(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2)$$

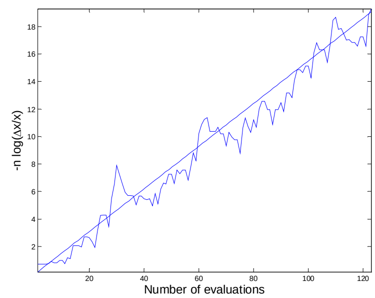


Convergence in parameters

Unimodal hump



Convergence of optimization variables. This shows the max relative spread dx of all (both) variables



log2 max(dx), and a straight line corresponding to the theoretical convergence rate.

Efficiency

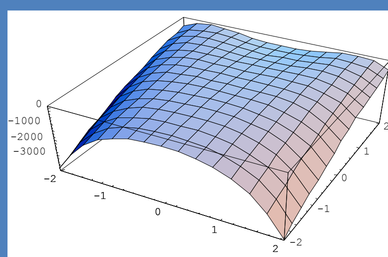
Unimodal hump

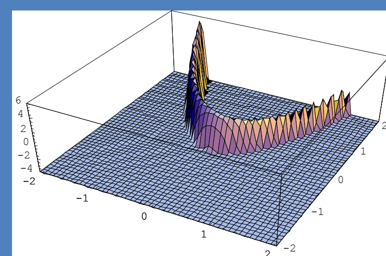
$$\phi = n\frac{\log_2 \varepsilon}{k_m} = -2\frac{\log_2 0.001}{99} = 0.2$$

bits/evaluation

Test function: Rosenbrock´s banana

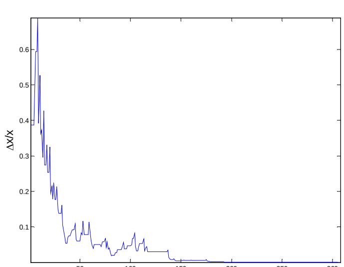
$$f(x_1, x_2) = 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_1\right)^2$$



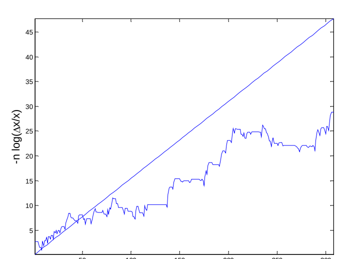


Convergence in parameters

Banana function



Convergence of optimization variables. This shows the max relative spread dx of all (both) variables



The convergence expressed as –n log2 max(dx), and a straight line corresponding to the theoretical convergence rate.

Meta object function.

Minimize the amount of evaluation needed to reach an optimum with a certain probability

$$(1-P) = (1-P_{opt})^m$$

$$m = \frac{Log_2(1-P)}{Log_2(1-P_{opt})}$$

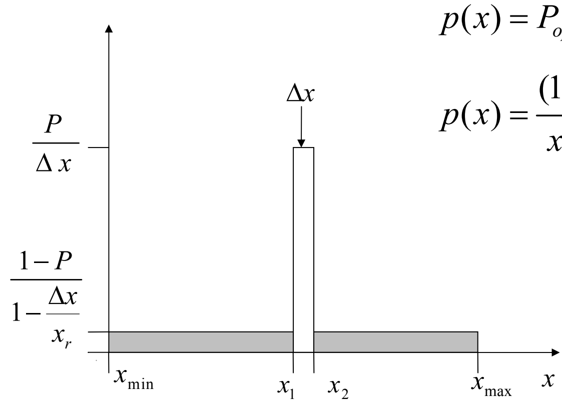
The number of optimization runs needed to reach optimum with a certain probability

$$\phi^{(1)} = \frac{I}{mk_m} = -n\frac{\log_2 \varepsilon}{Log_2(1-P)}\frac{Log_2(1-P_{opt})}{k_m}$$

The adjusted meta object function

$$\widehat{P}_{opt} = hitrate = \frac{N_{succes}}{N}$$

Probability distribution after optimization



$$p(x) = P_{opt}\frac{1}{\Delta x} : x \in [x_1, x_2]$$

$$p(x) = \frac{(1-P_{opt})}{x_r - \Delta x} : x \notin [x_1, x_2]$$

Relative Entropy of the Found Optimum

$$I_x = \int_{x_{\min}}^{x_{\max}} p(x)\log_2(p(x))x_r dx =$$

$$+\int_{x_{\min}}^{x_1} \frac{(1-P_{opt})}{1-\Delta x / x_r}\log_2\left(\frac{1-P_{opt}}{1-\Delta x / x_r}\right)dx +$$

$$+\int_{x_1}^{x_2} P_{opt}\frac{x_r}{\Delta x}x\log_2\left(P\frac{x_r}{\Delta x}\right)dx +$$

$$+\int_{x_2}^{x_{\max}} \frac{(1-P_{opt})}{1-\Delta x / x_r}\log_2\left(\frac{(1-P_{opt})}{1-\Delta x / x_r}\right)dx =$$

$$= (1-P_{opt})\log_2\left(\frac{1-P_{opt}}{1-\Delta x / x_r}\right) + P_{opt}\log_2\left(\frac{P_{opt}}{\Delta x / x_r}\right)$$

Meta object function

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

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

$$(1-P_{opt}) >> \varepsilon^n \qquad \phi^{(2)} \approx \frac{1}{N_m}P_{opt}\log_2\left(\frac{P_{opt}}{\varepsilon_x^n}\right)$$

I expresses the total uncertainty, representing the sum of

uncertainty in location

and

uncertainty of success

Example: The banana function

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

$P_{opt} = 0.85$
$\varepsilon = 0.001$
$N_m = 272$

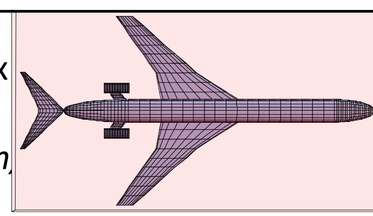
Calculate the efficiency in bits/eval

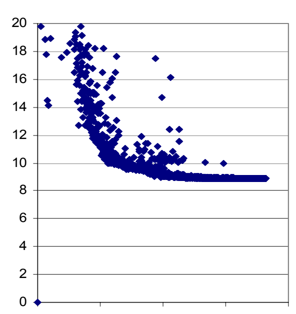
$$\phi^{(2)} = \frac{I_x}{N_m} = \frac{1}{N_m}\left((1-P_{opt})\log_2\left(\frac{1-P_{opt}}{1-\varepsilon_x^n}\right) + P_{opt}\log_2\left(\frac{P_{opt}}{\varepsilon_x^n}\right)\right)$$

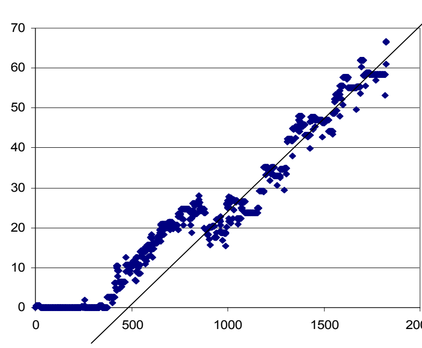
$$-\frac{1}{272}\left((1-0.85)\log_2\left(\frac{1-0.85}{1-0.001^2}\right) + 0.85\log_2\left(\frac{0.85}{0.001^2}\right)\right) = 0.044$$

bits/evaluation

Optimization Trace Using the Complex Optimization Method (*Aircraft Design Example, size planform, propulsion*)





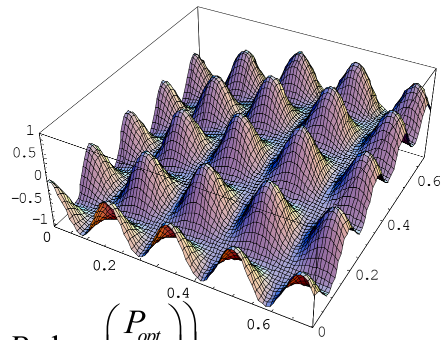


0.05 bits/ev al

Objectiv function (minimization)

Increase in information entropy in optimization variables

Example: Multiomodal function



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

$$-\frac{2}{133}\left((1-0.78)\log_2\left(\frac{1-0.78}{1-0.001^2}\right) + 0.78\log_2\left(\frac{0.78}{0.001^2}\right)\right) = 0.0087$$

compare to 0.2 bit/eval for unimodal. 22 times smaller!

bits/evaluation

Multiomodal function

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

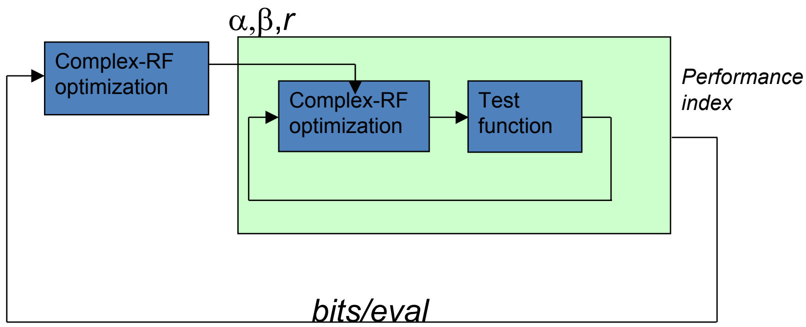
$$-\frac{2}{133}\left((1-0.78)\log_2\left(\frac{1-0.78}{1-0.001^2}\right) + 0.78\log_2\left(\frac{0.78}{0.001^2}\right)\right) = 0.0087 \text{ bits/evaluation}$$

compare to 0.2 bit/eval for unimodal. 22 times smaller,
for 25 times higher modality

- The ”curse of dimensionality ” is not in the number of parameters. It is in the modality of the output.

Meta optimization scheme

- In each evaluation point optimization of two test functions (Rosenbrock´s banana and the integer valued function) where done and the sum of the performance indices where taken as the measure of merit.



Variations in *r* and γ

using Rosenbrock´s banana as test function

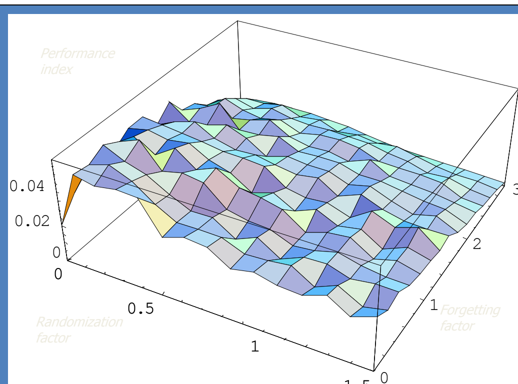


Figure 11. The performance index of the method, as a function of the randomization factor and the forgetting factor for Rosenbrock´s banana.

Information Entropy

- Design information entropy is a measure of the contraction of design space during design. It is roughly proportional to the number of objects in the design space and thus a more convenient unit than the number of possibilities excluded.
- Relative Information entropy is a very useful state in evolutionary learning processes. It is consistent with the view that the design process is a learning process.
- A general model is presented that links information to optimization, refinement, performance and cost.
- The results indicate that it sometimes may be more efficient to increase the number of design parameters, rather than refining a few parameters to a high degree.

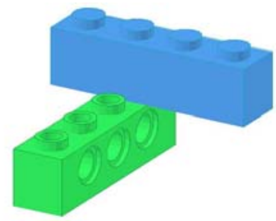
Information Entropy

- Using the performance criterion the COMPLEX-RF method has been compared with GA and Fmincon(Matlab). The result is surprising in that all the method are remarkably similar in performance index over a wide range of problems (within a factor of three).
- This is in support of the ”no free lunch theorem”, Wolpert and Macready.

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 arrangir 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:



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

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

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

Generalized

$$H_{rel} = -\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p(x_1\dots x_n)\log_2(\frac{p(x_1\dots x_n)}{m(x_1\dots x_n)})dx_1\cdots dx_n$$

Amount of Information content (Information Entropy)

The distribution m(x) should be a rectangular distribution in the bounded interval.

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

Generalized

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

More compact

$$I_x = -\int_S p(\mathbf{x})\log_2(p(\mathbf{x})S)dS$$

Amount of Information content (Information Entropy)

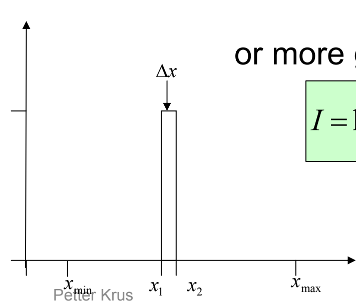
The information content *I* of a variable (in bits).

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

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

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

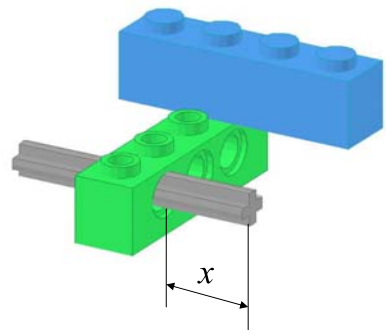
Here Δx is the tolerance in *x*.



or more general:

$$\boxed{I = \log_2\frac{S}{s}}$$

Design Space with Both Continuous and Discrete Variables



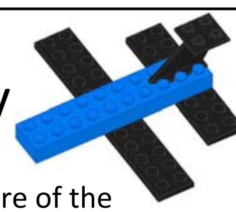
- The position of the inserted axis represents a continuous variables
- The information entropy associated with that is dependent on the accuracy with which it is specified.

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

- 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:

$$I_x = \log_2 51 + \log_2 3 + \log_2 \frac{1}{0.1} = 10.6\text{bits}$$

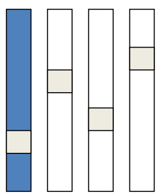
Design Information entropy



- The design information entropy represents a measure of the precision by which a design is defined relative to the design space in consideration.
- It is a measure of the dimensionality of the design problem
- It can also be seen as a measure of a probability that a random design in the design space is found, that is within the tolerance region of the design parameters.
- It is hence not possible to know the design information entropy of a design just by looking at it, since it is dependent on the design space from which the design has been picked.
- Design information entropy should not be taken as a direct measure of complexity. A very simple design can represent a large information content if it has been picked from a large design space.

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

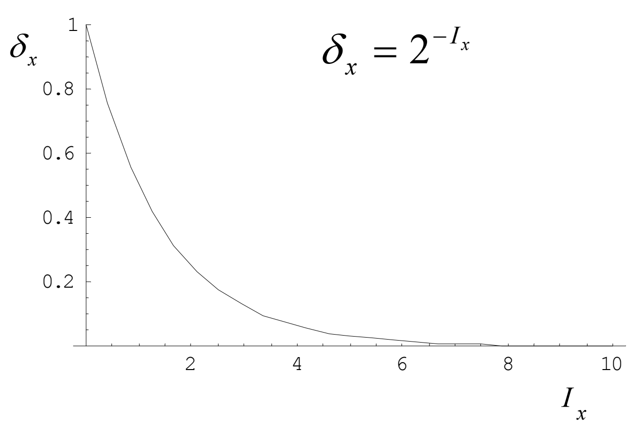


$$I_x = \log_2 \frac{x'_R}{\Delta x} = \log_2 \frac{x_R}{\Delta x}\frac{x'_R}{x_R} = \log_2 \frac{x_R}{\Delta x}\frac{x''_R}{x_R} = n\log_2 \frac{x_R}{\Delta x}$$
$$= nI_p$$

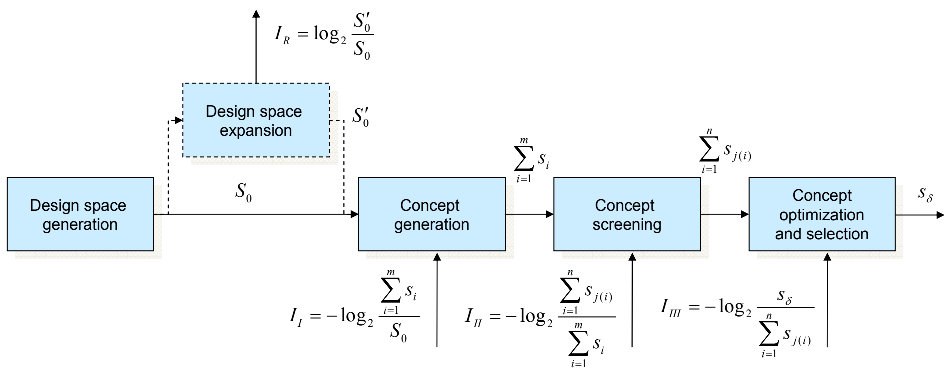
Design uncertainty as a function of design information

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

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



Growth of design information entropy during the design process



Influence factor

$$y = \mathrm{f}_x(x)$$
$$z = \mathrm{g}(\mathrm{y})$$
$$z = \mathrm{z}(x)$$

$$\psi(i) = \mathrm{z}_i\left(x_{R,i}, x_{p,opt,i}\right) - \mathrm{z}_{i-1}\left(x_{R,i-1}, x_{p,opt,i}\right)$$

An aproximate modell of this is:

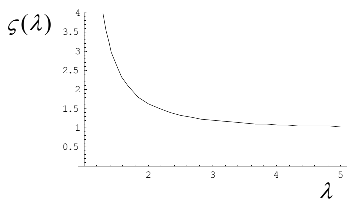
$$\psi(i) = c_p i^{-\lambda}$$

Asymptotic behaviour

$$z_{opt}(n) = z\left(n \middle| x_{p,opt}\right) = \Psi(n) - z_0 =$$

$$= c_p \sum_{i=1}^{n} i^{-k} - z_0 = c_p H(n,\lambda) - z_0$$

$$\lim_{n\to\infty} H(n,\lambda) = \varsigma(\lambda) = \begin{cases} \infty \text{ when } \lambda \le 1 \\ \text{finite value} > 1 \end{cases}$$



Sorted spectrum of parameter influence

$$\psi(i) = c_p i^{-\lambda}$$

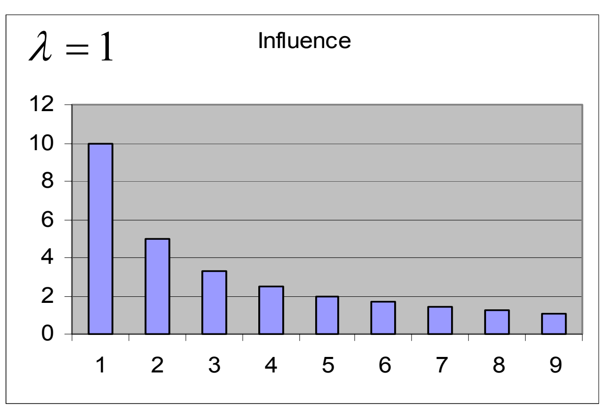


Figure 3. Sorted parameter influences

Total influence up to the n:th parameter:

$$\Psi(n) = \sum_{i=1}^{n} \psi(i) = c_p \sum_{i=1}^{n} i^{-\lambda} = c_p H(n,\lambda)$$

$H(n,\lambda)$ is the harmonic function

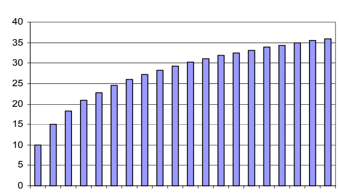
$$z_{opt}(n) = z\left(n \middle| x_{p,opt}\right) = \Psi(n) - z_0 =$$

$$= c_p \sum_{i=1}^{n} i^{-k} - z_0 = c_p H(n,\lambda) - z_0$$

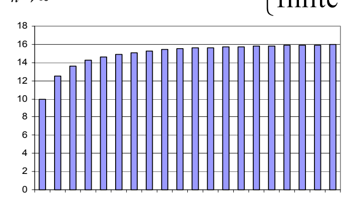
$$\approx c_p \left(c(\lambda) + \frac{1}{\lambda - 1}\left(1 - \frac{1}{n^{\lambda-1}}\right)\right) - z_0$$

Accumulated influence

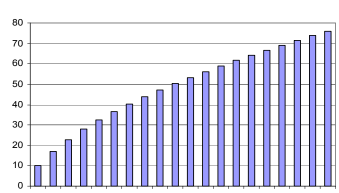
$$\lim_{n\to\infty} H(n,\lambda) = \varsigma(\lambda) = \begin{cases} \infty \text{ when } \lambda \le 1 \\ \text{finite value} > 1 \end{cases}$$



$\lambda = 1$



$\lambda = 2$



$\lambda = 0.5$

Performance model

$$p(n) = k_p H(n,\lambda) - p_0$$

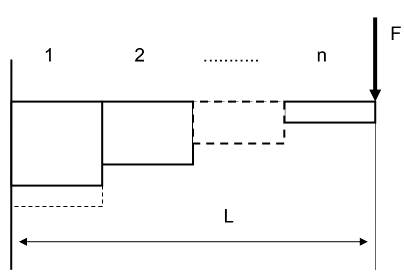
$$p_{opt}(n) - p_{opt}(n,\delta_x) = k_p H(n,k)\delta_x$$

Performance loss due to uncertainty

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

$$p_{opt}(n, I_x) = k_p H(n,k)(1 - 2^{\frac{I_x}{n}}) - p_0$$

Example: optimization of a beam



$$m = \frac{FL}{\eta k \sigma_{\max}} \qquad \eta \in]0,1]$$

Efficiency as a function of information

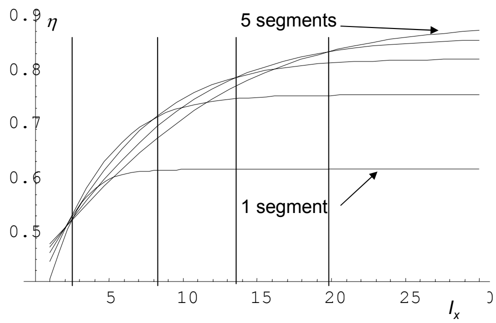
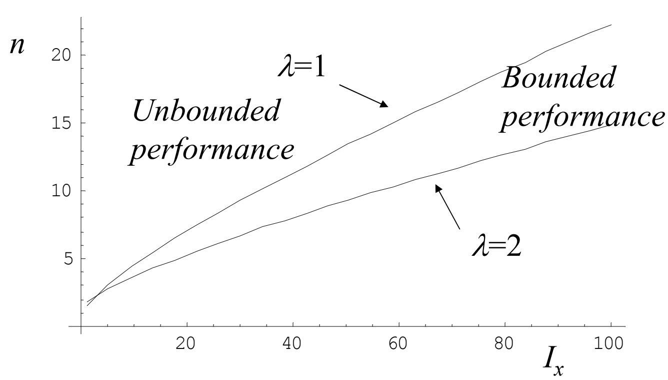


Figure 6. The efficiency of the structure as a function of information I_x

The optimal number of design parameters as a function of design information entropy



Influence values

$$\psi(i) = c_p i^{-\lambda}$$

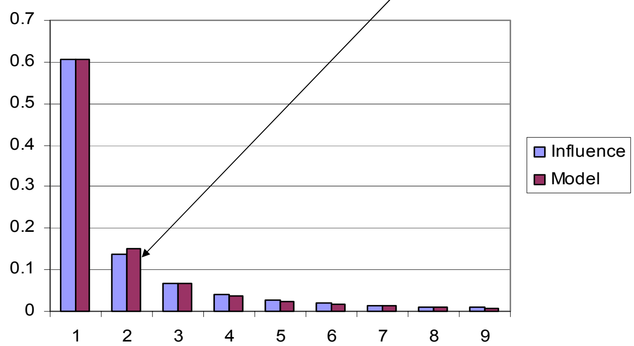


Figure 7. Influence of increasing the number of elements in the beam model. Exact solution and a model based on a negative power law relationship with λ=2.

Influence values

$$\psi(i) = c_p i^{-\lambda}$$

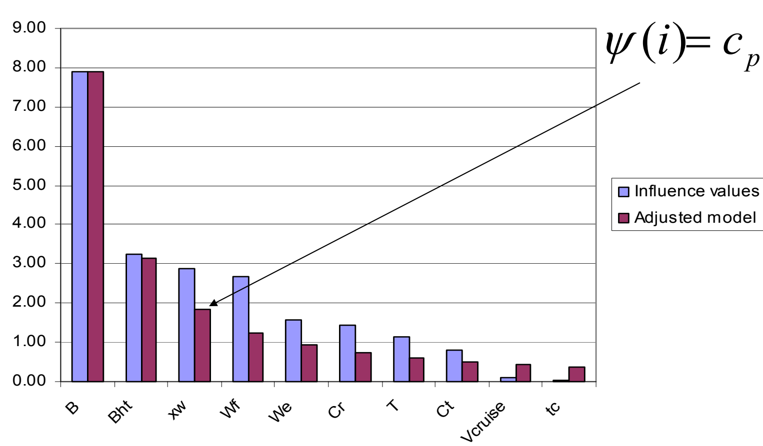


Figure 10. Sorted influences for different design parameters of a transport aircraft. Adjusted model with λ=1.4.

Performance model

$$z_{opt}(n, I_x) = c_p H(n, \lambda)(1 - 2^{\frac{I_x}{n}}) - z_0$$

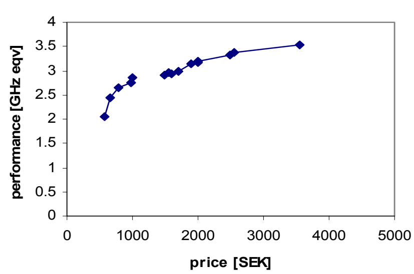
Substituting information with cost $I_x = C / k_c$

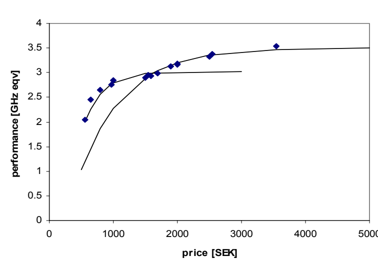
$$z_{opt}(n, C) = c_p H(n, \lambda)(1 - 2^{\frac{C/k_c}{n}}) - z_0$$

Or as a continuous function

$$z_{opt}(n, I_x) = k'_p \frac{1}{n^{\lambda - 1}}(1 - 2^{\frac{C}{k_{CI} n}}) - z'_0$$

Example



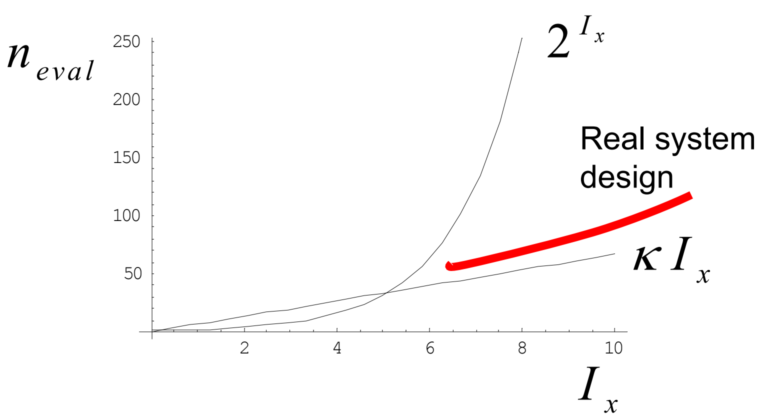


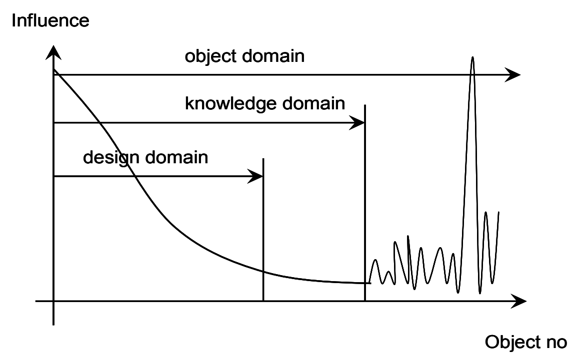
Processor performance as a function of price for Intel Pentium and Celeron processors (2004), with fitted model

Discussion

- Optimality is an illusion of ”Flatland” , we do not see the constraints that can be removed in the hidden dimmensions.
- The ”curse of dimmensionality” is in general resolved by settling for an acceptable solution.
- There is also the ”blessing of dimmensionality”, many solutions are in fact viable.
- The difference of different optima and suboptima are often shadowed by uncertainty.

Number of evaluations needed for a certain amount of information





If previously unknown design parameters enter the knowledge domain the optimal solution might shift into an entirely different optimum, a disruptive design

Why is design information entropy useful as a state?

- In the absence of other models, the simplest assumption is that cost is proportional to refinement, which is equivalent to information entropy. This applies both to design cost and manufacturing cost. and should also applicable cost of reliability etc.
- This is supported by the fact that non-gradient optimization methods converge such as the information increases approximately linearly.
- In the absence of more detailed models, the simplest assumption is that performance closes in on its maximum value as the uncertainty goes towards zero.
- It usually have a more convenient magnitude than the number of states.

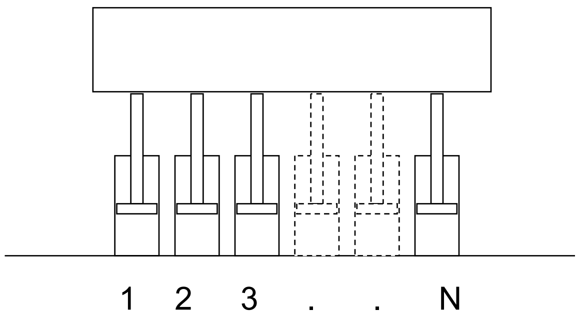
Neuro Mechanical Neworks

- There is a general trend for functional density of products to increase.
- Increasing customer requirements
- Prices of microprocessors, sensors and actuators is rapidly decreasing.

Introduction

- 3D-printing techniques
  – Layered manufacturing
- Direct Digital Manufacturing
  – Distributed manufacturing
  – The next big revolution

Performance and cost



Performance (force)

$$P = \sum_{i=1}^{n} c_{pi} s_i$$

Cost

$$C = \sum_{i=1}^{n} c_{ci} s_i^{\gamma}$$

If all s are the same

$$P = \sum_{i=1}^{n} c_p s = n c_p s$$

This yields the component size as:

$$s = \frac{P}{n c_p}$$

In the same way the cost can be calculated

$$C = \sum_{i=1}^{n} c_c s^{\gamma} = n c_c s^{\gamma} = n c_c \left(\frac{P}{n c_p}\right)^{\gamma} = n^{1-\gamma} c_c \left(\frac{P}{c_p}\right)^{\gamma}$$

γ >1 drives n to be large and component size to minimum!

More general

$$P = \sum_{i=1}^{n} c_p s_i^{\gamma_p}$$

$$C = \sum_{i=1}^{n} c_c s_i^{\gamma_c}$$

If all s are the same

$$P = \sum_{i=1}^{n} c_p s = n c_p s^{\gamma_p} \quad \text{then} \quad s = \left(\frac{P}{n c_p}\right)^{\frac{1}{\gamma^p}}$$

In the same way the cost can be calculated

$$C = \sum_{i=1}^{n} c_c s^{\gamma_c} = n c_c s^{\gamma_c} = n c_c \left(\frac{P}{n c_p}\right)^{\frac{\gamma_c}{\gamma_p}} = n^{1-\frac{\gamma_c}{\gamma_p}} c_c \left(\frac{P}{c_p}\right)^{\frac{\gamma_c}{\gamma_p}}$$

$\gamma_c > \gamma_p$ drives component size to minimum!

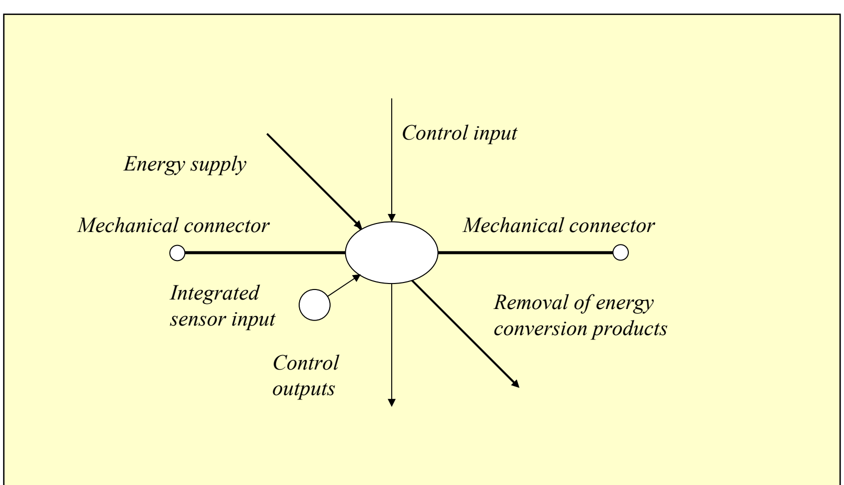
Consequences

- In electronics γ_c is already greater than γ_p since γ_p is close to zero. Performance is much more related to number than size.
- As this happens in other technologies, component size will be driven down to the minimum size.
- This has happened in biological systems
- This leads to hyper functional technical products

The Generic Integrated Actuator

- Actuation, through mechanical connections
- Energy supply and removal of energy conversion residues
- Control and/or sensory input
- Sensory output
- Simple signal processing

The Generic Integrated Actuator



Neural Network

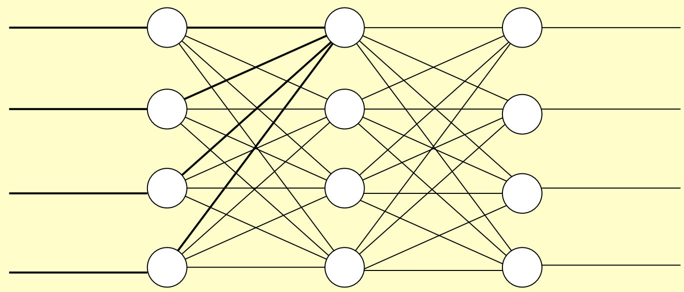


Figure 2. Neural network with one hidden layer.

Neuron

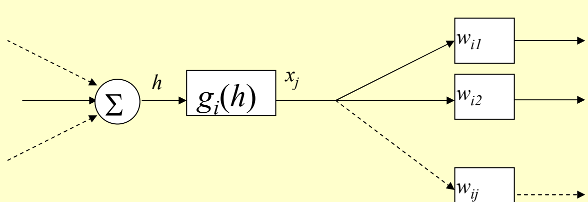
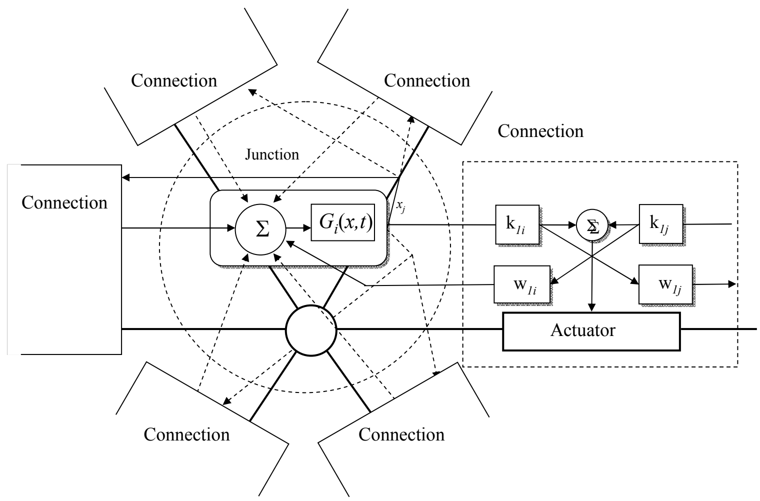
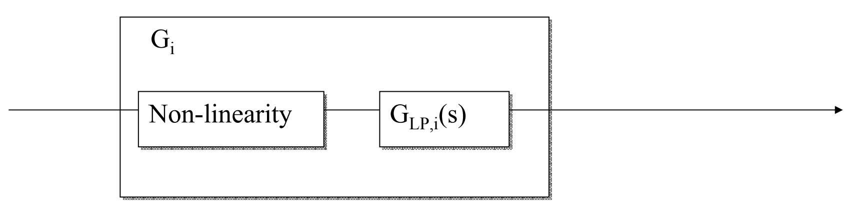


Figure 3. Neuron

Recurrent Neural Network superimposed on a Mechanical Network
Neuron and connection/actuator



The transfer function G(x,t)



The neuro-mechanical network

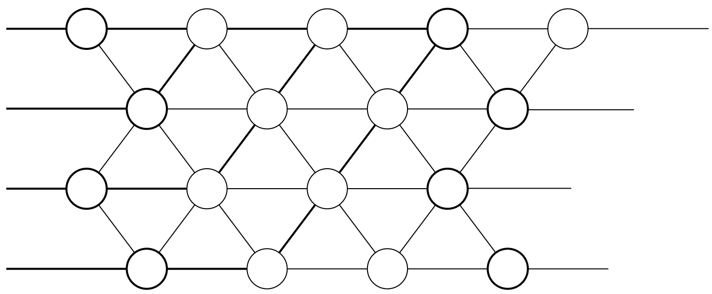


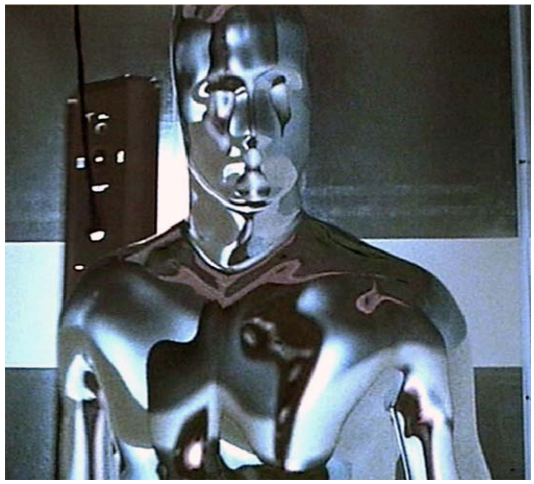
Figure 4 Neuro-mechanical network corresponding to the first two layers in the neural network.

Applications

- As a generic structure using optimisation for mechatronic system design
- As a model for future massively redundant mechatronic systems
- As an explanation model for biological systems

Application?

T-1000 from T2



Example

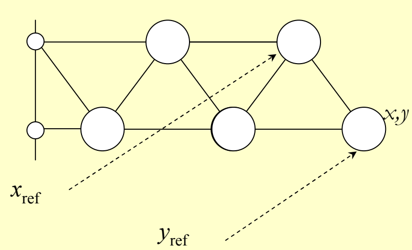


Figure 7. A neuro-mechanical network with five neuron, eight actuators and two inputs.

The objective function

$$f_{obj} = \int \left(x - x_{ref}\right)^2 dt + \int \left(y - y_{ref}\right)^2 dt$$

All the 30 static gains are optimisation variables

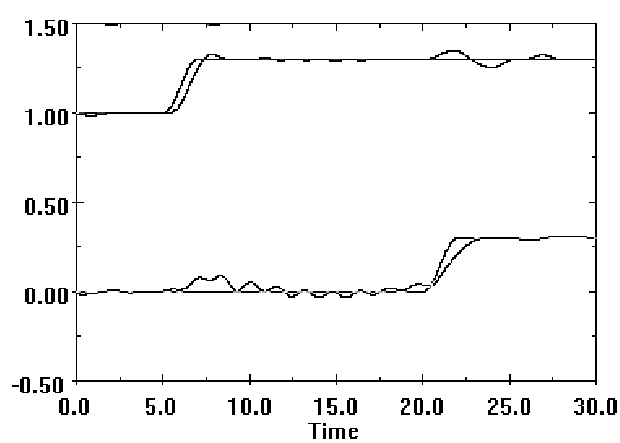


Figure 9. The reference and actual positions in x- and y- directions at the tip.

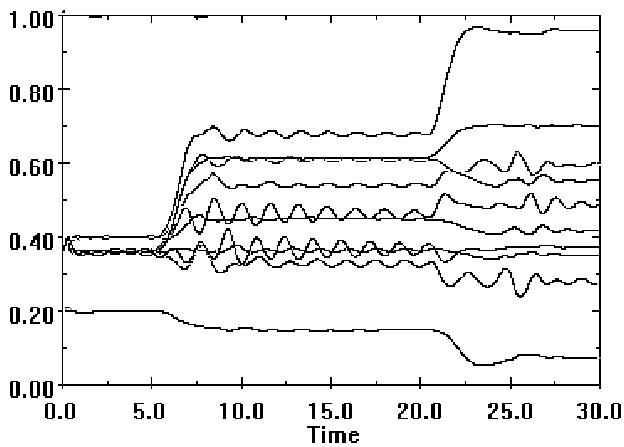
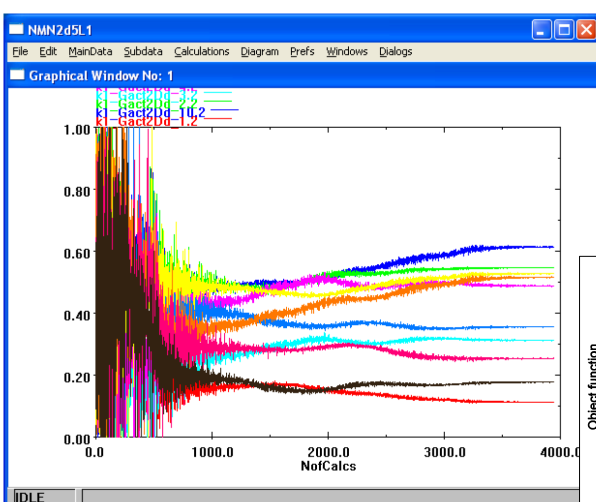
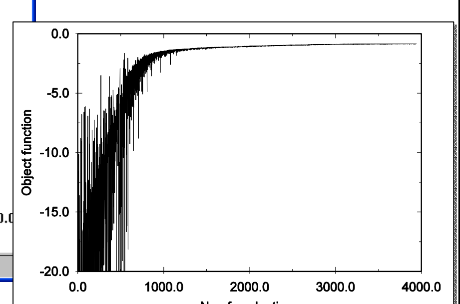


Figure 10. The corresponding actuator positions

Optimum found in 3940 evaluations (131 evaluations/parameter)





Discussion

- The Neuro mechanical elements can be viewed as engineering stem cells
- They can be used to produce “organic” structures, and parts can have degrees of freedom to be specialised as signal processing, actuators, and pure structure.
- They can also be used to generate more conventional systems through topology optimisation

Publications related to the topics

- Krus P. 'The Relation Between Performance, Design Space, Refinement and Information'. International Design Engineering Technical Conferences Computers and Information in Engineering Conference ASME/IDETC, Las Vegas, Nevada, USA, 2007.
- Krus P., Andersson J. "An Information Theoretical Perspective on Design Optimization" Proceedings of 2004 DETC:Design Engineering Technical Conference. Salt Lake city, Utah.
- Krus P, Karlsson M,’Neuro-Mechanical Networks – Self-Organizing Multifunctional Systems' Presented at the 'Ninth Bath International Fluid Power Workshop', Bath, UK 2002.