The Role of Entropy in Design Theory and Methodology

Waseem A. Khan    Jorge Angeles

Centre for Intelligent Machines &
Department of Mechanical Engineering
McGill University
Montreal, Quebec, Canada
Outline

1. Introduction
2. The Nature of Design
3. The Concept of Entropy in MTC
4. Design Complexity
5. Conclusions
Introduction

The Word “Design”

- “Design” is derived from the Latin “designare,” meaning “to mark out”
- It bears many meanings in English, e.g.,
  - Noun: To designate the product, the process, etc
  - Verb: To conceptualize a product intended to satisfy a human need
Engineering Design

- Engineering design is the design of engineering/technical products
- A pervasive activity that appears in every engineering project
- A few schools of thought have appeared that attempt to formalize the design activity

Research is still needed to lay the foundations in a broadly acceptable framework
Existing Design-Process Models

Three most frequently cited models

French’s Design-Process Model

- Conceptual design: Design specifications, synthesis of alternatives, selection of promising alternatives
- Embodiment design: Sketches and preliminary design drawings, analysis and optimization
- Detail design: Elaborate report on the design activity and manufacturing drawings, etc.

Design process layout after French (1999)
Conceptual Design

Three main schools in conceptual design

- **German School**: Provides guidelines approved by VDI
- **Suh’s Axiomatic Design**: Provides two axioms and an outline of a design methodology. The information axiom explores the concept of *information content*
- **Taguchi’s Robust Design**: Based on the concept of robustness, aims to maximize the signal-to-noise ratio

Conceptual design is characterized by the absence of a mathematical model
The Concept of Information Entropy

Reference


- Provides insight on the quantification of the amount of information in a message
- Provides a measure on the capacity of a communication channel

The theory is associated with the amount of freedom of choice that one has in constructing (or interpreting) a message.
Shannon’s Formulation of Information Entropy

\[ H = - \sum_{i=1}^{n} p_i \log(p_i), \quad \sum_{i=1}^{n} p_i = 1 \]

- \( p_i \): the probability of an outcome
- \( \log(p_i) \): the logarithm of \( p_i \) to a certain base

The choice of the logarithmic base is open

- Binary or base 2 (\( H \) is measured in \textit{bits})
- Naperian or base \( e \) (\( H \) is measured in \textit{nats})
- Briggs or base 10 (\( H \) is measured in \textit{decibels})

Base 2 is preferred in Information Theory
Example

Communicate via telephone by spelling out the name *Anne*

We risk the ambiguity of *homophones* in the process

- For “A”, ambiguity is between “A,” “8” and “H”
  \[ H_A = - \sum_1^3 \frac{1}{3} \log_2(3) \]
- For “N”, ambiguity is between “N” and “M”
  \[ H_N = - \sum_1^2 \frac{1}{2} \log_2(2) \]
- For “E”, ambiguity is between “E,” “B,” “C,” “G,” “P,” “T,” and “V”
  \[ H_E = - \sum_1^8 \frac{1}{8} \log_2(8) \]
Example (Cont’d)

The total information content for the four messages becomes

\[ H = H_A + 2 \times H_N + H_E = 6.585 \text{ bits} \]
Example (Cont’d)

- We can use the *International Alphabet*
- In this case, $H = 0$ bits

<table>
<thead>
<tr>
<th>alpha</th>
<th>November</th>
</tr>
</thead>
<tbody>
<tr>
<td>bravo</td>
<td>Oscar</td>
</tr>
<tr>
<td>Charlie</td>
<td>papa</td>
</tr>
<tr>
<td>delta</td>
<td>Quebec</td>
</tr>
<tr>
<td>echo</td>
<td>Romeo</td>
</tr>
<tr>
<td>foxtrot</td>
<td>sierra</td>
</tr>
<tr>
<td>golf</td>
<td>tango</td>
</tr>
<tr>
<td>hotel</td>
<td>uniform</td>
</tr>
<tr>
<td>India</td>
<td>Victor</td>
</tr>
<tr>
<td>Juliett</td>
<td>whiskey</td>
</tr>
<tr>
<td>kilo</td>
<td>xray</td>
</tr>
<tr>
<td>Lima</td>
<td>yankee</td>
</tr>
<tr>
<td>Mike</td>
<td>zulu</td>
</tr>
</tbody>
</table>

The International Alphabet

Angeles, J.  | Entropy in Design  | 12
Design Complexity

Proposal

Regard the entropy of a design as its *complexity*

Design complexity ⇔ Diversity in the design-solution

Definitions

- **Function**: A generic task imposed by the need to be satisfied by means of the object under design
- **Function-Carrier**: A component or assembly intended to implement a function
- **Design Specifications**: A quantitative condition to be met by the object under design
Computing the Design Complexity

Assumptions:

1. The implementation of a given function $F$ requires $N_c$ carriers;

2. $F$ is decomposed into $N$ subfunctions $f_1, f_2, \ldots, f_N$, each to be implemented with $\nu_i$ identical carriers

$\Rightarrow \quad \nu_1 + \nu_2 + \nu_N = N_c$

As a consequence,

1. $\nu_i$ can take on any value between 0 and $N_c$: $0 \leq \nu_i \leq N_c$;

2. Complexity is a minimum when all $N_c$ carriers are identical, i.e., when $N = 1$; a maximum when $N = N_c$ and $\nu_i = 1, \ i = 1, 2, \ldots, N$
Computing the Design Complexity (Cont’d)

Obtain the frequency of occurrence of each function-carrier, i.e.,

\[ \phi_i = \frac{\nu_i}{N_c}, \quad i = 1, \ldots, N \]

and hence,

\[ \sum_{1}^{N} \phi_i = 1 \]

whence it is apparent that the frequencies \( \phi_i \) are all positive and lying within the interval \([0, 1]\).

This set of values behaves like a \textit{discrete probability distribution}. 
Computing the Design Complexity (Cont’d)

As each $\nu_i$ can take on any value between 0 and $N_c$, we have $W$ ways of choosing the identical carriers, given by (combinatorics)

$$W = \frac{N_c!}{\nu_1!\nu_2!\cdots\nu_N!}$$

$W$: a measure of the diversity of the design solution proposed.

Remarks:

- Above $W$ only accounts for one function, and we may have several functions;
- we’d better have a measure of diversity that is additive;
- this measure would allow us to compute the diversity of the design solution to implement all functions as the sum of the partial diversities.
Computing the Design Complexity (Cont’d)

⇒ We adopt a logarithmic measure:

\[
\log(W) = \log \left( \frac{N_c!}{\nu_1!\nu_2! \cdots \nu_N!} \right)
\]

Remark: As computing \( W \) involves factorials, the computation of its logarithm is rather cumbersome. Use Sterling’s formula to approximate the natural logarithm of the factorial of a positive integer \( Z \):

\[
\ln(Z!) \approx Z \ln(Z) - Z
\]

which is a good approximation for “large” \( Z \). For \( Z \) of the order of 10, the error in the approximation is of about 13%.
With this approximation,

$$\ln(W) \approx -N_c \sum_{1}^{N} \phi_i \ln(\phi_i)$$

⇒ a measure of the diversity of the design solution to implement function $F$, considering *all* function-carriers.
Computing the Design Complexity (Cont’d)

If we want a measure of the diversity of the solution per function-carrier, we have to divide $\ln(W)$ by $N_c$ to obtain the complexity $K_F$ of a given design-solution to implement function $F$:

$$K_F = - \sum_{1}^{N} \phi_i \ln(\phi_i) \text{ (nats)}$$

which would give $K_F$ in “nats.”

If we want $K_F$ in bits, we have to use binary logarithms (in spite of Sterling’s formula being valid only for natural logarithms!):

$$K_F = - \sum_{1}^{N} \phi_i \log_2(\phi_i) \text{ (bits)}$$
Computing the Design Complexity (Cont’d)

If we have $N_F$ functions $F_j$, each with a complexity $K_j$, to implement in a design-solution, then the total complexity $K$ of the solution proposed is

$$K = - \sum_{1}^{N_F} K_j$$
Example 1

Problem

Design an autonomous vehicle capable of transporting bundles of veneer in a production plant

Constraints

1. Tricycle design
2. Use three conventional wheels
3. Autonomous $\iff$ two motors under computer control
Example 1 (Cont’d)

Analysis of Functions

We have one main function:

Move the veneer bundles at a constant height from the floor, enough to negotiate small obstacles, with the capability of negotiating curves.

We can either a) decompose this function into two possible subfunctions: $F_1$, drive, and $F_2$, steer, or b) keep the “move” function as one single subfunction, which would make subfunction $\equiv$ function.
Example 1 (Cont’d)

Two Possible Options

(a) ⇒ We use two different motors to implement the two subfunctions, so that $N_c = 2; \nu_1 = \nu_2 = 1$

(b) ⇒ We use two identical motors to implement the single function “move” ⇒ $N_c = 2, \nu_1 = 2$

We have now several possible actuation modes, as described below:
Example 1 (Cont’d)

Actuation alternatives for an autonomous tricycle

1 Cannot be passive
S = Steered
D = Driven
Example 1 (Cont’d)

Case (a) with two identical “fixed” wheels
An Alternative Embodiment of Case (a)

Produced by Gabriel Hernández (2007)
Example 1 (Cont’d)

Design complexity when two distinct motors are used

\[ K = - \sum_{1}^{2} 0.5 \log_2(0.5) = 1.0 \]

Design complexity when two identical motors are used

\[ K = - \sum_{1}^{1} 1.0 \log_2(1.0) = 0.0 \]

Apparently, the concept with identical motors is more promising
Example 2

Problem

Design a fast robot for pick-and-place operations requiring three independent translations and one rotation about a vertical axis

Remarks:

- Desired motion is similar to that of the tray of a waiter (no tilt allowed);
- Motion has four dof: three translations and one rotation $\Rightarrow$ Four function carriers are needed;
- $\Rightarrow$ function *move* can be divided into 1, 2, 3 or 4 subfunctions, depending on how the subfunctions are implemented.
Solution 1

We use four distinct motors, thereby leading to a serial robot, e.g., the Adept Cobra s600

\[
\nu_1 = \nu_2 = \nu_3 = \nu_4 = 1, \quad N_c = 4
\]

\[
\Rightarrow \phi_1 = \phi_2 = \phi_3 = \phi_4 = \frac{1}{4}
\]

\[
K_1 = -4 \left[ \frac{1}{4} \log_2 \left( \frac{1}{4} \right) \right]
\]

\[
= 4 \times \frac{1}{4} \times 2
\]

\[
= 2
\]
Example 2 (Cont’d)

Solution 2

We use three distinct motors, thereby leading to a serial-parallel robot, e.g., the ABB IRB660-1

\[
\nu_1 = 1, \quad \nu_2 = 2, \quad \nu_3 = 1, \quad N_c = 4
\]

\[
\Rightarrow \quad \phi_1 = \frac{1}{4}, \quad \phi_2 = \frac{1}{2}, \quad \phi_3 = \frac{1}{4}
\]

\[
K_2 = -2 \left[ \frac{1}{4} \log_2 \left( \frac{1}{4} \right) \right] - \frac{1}{2} \log_2 \left( \frac{1}{2} \right)
\]

\[
= 2 \times \frac{1}{4} \times 2 + \frac{1}{2}
\]

\[
= 1.5
\]
Example 2 (Cont’d)

Solution 3

We use two distinct motors, thereby leading to another serial-parallel robot, namely, the ABB FlexPicker

\[ \nu_1 = 3, \quad \nu_2 = 1, \quad N_c = 4 \]

\[ \Rightarrow \phi_1 = \frac{3}{4}, \quad \phi_2 = \frac{1}{4} \]

\[ K_3 = -\frac{3}{4} \log_2 \left( \frac{3}{4} \right) - \frac{1}{4} \log_2 \left( \frac{1}{4} \right) \]

\[ = \frac{3}{4} \left[ 2 - \log_2(3) \right] + \frac{1}{4} \times 2 \]

\[ = \frac{8}{4} - \frac{3}{4} \log_2(3) = 0.8112 \]
Example 2 (Cont’d)

Solution 4

We use four identical motors, thereby leading to a parallel robot

\[ \nu_1 = 4, \quad N_c = 4 \]

\[ \Rightarrow \quad \phi_1 = 1 \]

\[ K_4 = -1 \times \log_2(1) \]

\[ = 0 \]
Example 2 (Cont’d)

Examples of Solution 4 ($K_4 = 0$)

(a) The Adept Quattro S650; and (b) the McGill SMG
Conclusions

- Proposed *complexity* as a measure of the *diversity* content of a design solution at the conceptual stage.
- The proposed measure is computable with the information content introduced in the mathematical theory of communication.
- The concept was illustrated by examples from *robot design*.