Evaluation of the Program for Synthesis of Solution Principles

Amaresh Chakrabarti
EDC, Department of Engineering,
University of Cambridge
Trumpington Street, Cambridge, UK

Introduction

This paper reports on the progress of the work on synthesis of solution principles proposed in the earlier colloquia [1, 2, 3]. In the last colloquia, we mainly focused on how the synthesis approach works and provided some examples of its potential. This report focuses mainly on the experience gained by trying to create a database which provides an efficient means of generating a wide variety of principles.

Application Focus

In the area of sensor designs, various types of sensors are available. The classifying characteristics identified so far, which by no means are exhaustive, are: (i) single variable transformers versus multi-variable transformers, (ii) instantaneous versus temporal, (iii) no feedback versus feedback, (iv) continuous versus discrete.

Example 1: single variable, no feedback

In this case, a proof mass causes an inertia force to be generated due to the input acceleration, which is transformed into a deflection by a spring (beam). The resulting deflection closes in/out a small gap, causing a change of tunnel current across the gap.

Example 2: single variable, feedback

In this version of the above concept, tunnel current is nullified by feedback and the required electrostatic compensation is measured.

Example 3: multiple variable, continuous, no feedback

In this case, the accelerometer contains a beam attached to the object which senses the input acceleration. The other end of the beam is attached to a mass. Acceleration of the object causes the beam to deflect, causing some of its fibres to tense and others to compress. Both these are measured using strain gauges, and their difference is an indication of the acceleration.

Example 4: single variable, temporal, feedback

In this version of the tunnelling principle in example 2, the output current is kept in a tapping mode with an electrostatic driving voltage, and the change in acceleration causes the time between the tappings.

Example 5: multiple variable, discrete, no feedback

In this velocity measuring instrument, two magnetic coils are kept a fixed distance apart, and the object which senses the velocity moves under these coils. The object has a magnetic projection which of the tunnelling principle in example 2, the output current is kept in a tapping mode with an electrostatic driving voltage, and the change in acceleration causes the time between the tappings.

The present report concentrates on single variable, continuous, instantaneous transformers with no feedback involved.

The Synthesis Approach

The earlier papers [1, 2, 3, 4] describe the synthesis approach in greater detail. Here is a brief summary. The approach is to concatenate a number of device building blocks into chains of device building blocks, described here as solution principles, which transform the input to be sensed into an output which represents it. In the examples given in the previous section, the input was acceleration or velocity, and the output was current or voltage. The building blocks therefore need to have an input and an output each, and the building blocks constituting a solution principle transforms the input to be sensed into a number of intermediate variables before producing the required output.

Creating the Database of Device Building Blocks

Device building blocks are identified by analysing various existing sensing devices to find common building blocks across them, as well as various physical effects not used in the devices considered. We call this random population of the database. We then used the synthesis approach to see what kinds of principles it used. Each kind was then analysed to see what caused them, so as to enhance or avoid those types.

Kinds of Principles Generated

• *Using different physical effects*

The major part of the problem in the accelerometer problem is how to get some known electrical output as a measure of acceleration, while the minor part is how to transform that electrical output into a representative voltage. A designer's objective would naturally be to maximise the variety of principles in the major problem. In this category, the principles were different in terms of the underlying physical effects they used. For instance, contrast a piezo-type principle with a strain-gauge type principle. The piezo type uses a mass to transform the input acceleration into a force, a force-stress relationship to transform the force into a stress, and a stress-strain relationship to transform this stress into a strain, just as the strain-gauge type does. In the piezo case this strain is then converted into an electrical charge using a piezo-effect, while in the strain-gauge type, the strain causes by definition a length deformation leading to a change in resistance. These two use separate physical effects to do the transformation. They also use separate effects to transform charge/resistance into voltage, but this is the minor part of the problem.

We certainly wish to retain the capability to generate this type of principles. How?

• *Using different variables from the same physical effect for the major part of the problem*

Consider these two principles. In both the case, a mass-spring arrangement is used to transform the acceleration into a position change. The first case transforms this position change to change the separation between two capacitive plates to change their capacitance, while the second principle uses the position change to change the overlapping area of the capacitance. They use, for the transformation between position and capacitance, the same principle of capacitance (which changes with separation between plates, change in permittivity of the dielectric between the plates, and area of overlap between them).

We certainly need these variants as well, but perhaps at a later stage of development when the designer has decided on using capacitive principles. We could group all three building blocks together into a generic block which transforms a position change into a capacitance change. However, the danger is, that this will deprive us of generating principles which perhaps transform in some other ways an acceleration into a change in permittivity, which would have given us an elegant solution to the problem. This is also possible for both the major and the minor parts of the problem.

• *Principles containing parts which can be transformed using fundamentally the same but representationally more/less granular building blocks*

Consider the transformation between a force change and a position change. We could do this using a spring which describes this as a force-to-position transformation, or could do this using a combination of a force-to-stress, stress-to-strain and a strain-to-position transformation, which is fundamentally how a spring works. If we keep both in the database, we run into the danger of duplication. Keeping this only as a force-position relationship allows us to use them in principles with less building blocks, but does not allow its constitutive relationships in other principles. This is possible for a major as well as a minor part of the problem.

• *Using different alternative single building blocks to do the same transformation*

Take the example of transforming a rotation into a current. This can be done using two alternative building blocks: either by Wiedemann effect, or by an eddy-current based configuration. These are useful again perhaps at a later stage, where the designer wishes to explore principles which contain this transformation.

<u>Issues</u>

The above examples indicate that in order to suppress inflation in the number of solution principles which are detailed variants, while still retaining the width of variety between them, we need to consider the following:

• *Variety is characterised by variety of variables and relationships*

We need to increase the number of variables which are used by at least one building block of the database. We need to increase the number relationships between these variables. Using the more granular constituents rather than the single spring relationship in the previous section gives us a number of new variables and a number of additional relationships. The first example above also shows that unless the relationships which connect the variables exist, these solution principles cannot be generated.

• *Detail is characterised by alternative ways of achieving the same relationship*

We need to avoid duplication of relationships. This can be either in terms of the same relationship written more than once (as different single blocks are used to do the relationship), or the same transformation written in more than one ways which are fundamentally the same but are but different expressions (as in the spring relationship case).

• *We also need to concentrate on solving the major part(s) of the problem*

We need to avoid spending time on finding creative alternatives for a minor part of the problem. For instance, transforming current into voltage is the minor part of the accelerometer problem and we could avoid this. The difficulty is that it is hard to key in what is and is not impossible into a generic database. What is minor for this problem may turn out to be major for another. The route envisaged is to allow designers the opportunity of dealing with more than one, alternative, output variables, transforming to any of which will suffice as a solution to the problem. The efficiency can be increased by putting in additional constraints that transformation between these alternatives within the solutions is not allowed.

Experiment with databases and a measure of their effectiveness

The above analysis indicates that in order to analyse a database for its effectiveness, it may be useful to represent the variables and relationships it has (relationships are building blocks and variables are their input/outputs). The effectiveness of a database is the number of solutions it produces with change in the number of building blocks allowed to be used in a principle. The amount of time it takes to do this have all been within minutes, and thus have not been considered, although for much larger problems, this may be an issue would have to be considered as a trade-off with the number of solutions.

Establishment of a measure of effectiveness of a database

If a database is described as a network of relationships between a set of variables, and two of its variables are described as the input and output required of a given design problem (in the case investigated these are acceleration and voltage respectively), then the number of solutions generated are all the routes that are possible between the input and output nodes of this directed graph. Therefore, there can be variables and relationships which are not connected in an effective way to be able to contribute to the generation of solutions (for instance that a relationship exists between flow-rate and voltage and flow-rate is not connected via any other relationships to any other variable). This implies that simply populating the database will not necessarily make it more effective.

We have developed a set of rules to simplify a given network (database) such that all variables and relationships which certainly do not contribute to the problem under investigation are eliminated before measuring its effectiveness. The simplified network is then calculated for the average value of the number of inputs and outputs from its nodes, and its standard deviation. The better a database is, the more is number of nodes and average number of inputs and outputs are, and the less is its standard deviation. The rationale behind this is that if there are more variables in a database, and these are potentially useful in the sense that they are highly connected (described by a high value of average and a low value of standard deviation), then it is more likely to produce a larger and varied number of principles.

Experiment 1:

Original number of building blocks: 28
Original number of variables: 19

Simplified number of building blocks: 12
Simplified number of variables: 9

Number N cumulative of solutions for a maximum r allowed building blocks per solution principle are:

at r=1	N=0
at r=2	N=0
at r=3	N=0
at r=4	N=0
at r=5	N=1
at r=6	N=2
at r=7	N=5
at r=8	N=5
at r=9	N=5
at r=10	N=5

Average value of I/O per node: 25/9 = 2.667

Experiment 2:

Original number of building blocks: 45
Original number of variables: 26

Simplified number of building blocks: 21
Simplified number of variables: 13

Number N cumulative of solutions for a maximum r allowed building blocks per solution principle are:

at r=1	N=0
at r=2	N=0
at r=3	N=0
at r=4	N=0
at r=5	N=1
at r=6	N=2
at r=7	N=5
at r=8	N=6
at r=9	N=6
at r=10	N=7

Average value of I/O per node: 42/13 = 3.231

Experiment 3:

Original number of building blocks: 60
Original number of variables: 28

Simplified number of building blocks: 29
Simplified number of variables: 16

Number N cumulative of solutions for a maximum r allowed building blocks per solution principle are:

at r=1	N=0
at r=2	N=0
at r=3	N=0
at r=4	N=1
at r=5	N=5
at r=6	N=11
at r=7	N=19
at r=8	N=32
at r=9	N=56

Average value of I/O per node: 58/16 = 3.625

Experiment 4:

Original number of building blocks: 68
Original number of variables: 28

Simplified number of building blocks: 43
Simplified number of variables: 21

Number N cumulative of solutions for a maximum r allowed building blocks per solution principle are:

at r=1	N=1

at r=2	N=1
at r=3	N=2
at r=4	N=4
at r=5	N=11
at r=6	N=26
at r=7	N=53

Average value of I/O per node: 86/21 = 4.10

As we see, the combination of the three indices, simplified number of variables, simplified number of building blocks and average value of I/O per node, together

Ideas Generated By the Program

The computer generated a large number of ideas. The ideas fell in the four categories mentioned earlier. The designers so far considered four principles [5], of which two fall in the category of single variable, no-feedback, continuous, instantaneous devices which are considered in this category. Both these apart from a wide variety of other principles have been generated by the program which indicates its potential for stimulating designers' thinking.

With the introduction of new variables, new combinations came up which used a number of domains (such as electrical, optical etc) to solve the problem as opposed to initial single or bi-domain solutions. The principles generated interesting, simple ideas as well as complex or variant ideas. We choose three to give a flavour of the potential of the program.

Principle 1: Tolman Effect

This principle uses as a single building block the phenomenon of the development of a potential across a conductor due to electron inertia, when it is accelerated.

Principle 2: Triboelectricity

In this principle, a mass and spring is used to cause a position change due to the acceleration, and this position change is transformed into a voltage by a single building block based on the phenomenon of triboelectricity (relative movement of some materials cause a potential to be induced across them).

Principle 3: Thermoelasticity

In this principle, a mass is used to transform the acceleration into a force, is used to stress a thermoelastic material which develops a voltage across the material.

Summary and Conclusions

A large number of devices and effects have been analysed to create a database of building blocks for synthesis of devices. A number of experiments have been done with a variety of devices to understand what causes an effective database which maximises the potential for generating a wide variety of principles. A method for representing, and simplifying databases is used, using which three indices have been identified which together give a reasonable index of the database effectiveness. The program generated many more ideas including those generated by the designers in the category considered here. However, much further work needs to be put in to include all the other categories of devices.

References

[1] Chakrabarti, A. "Towards a Support Framework for the Generation and Exploration of Conceptual Design Solutions" in Workshop on Knowledge Sharing Environment for Creative Design of Higher Quality and Knowledge Intensifies, 12-13 January, 1995

[2] Chakrabarti, A. and Johnson, AL.. "An Approach to Automated Synthesis of Solution Principles for Transducer Designs" in Workshop on Knowledge Sharing Environment for Creative Design of Higher Quality and Knowledge Intensifies, January, 1996.

[3] , T. and Johnson, AL.. "Functional Modelling for the Design of Micro mechanisms" in Workshop on Knowledge Sharing Environment for Creative Design of Higher Quality and Knowledge Intensiveness, 12-13 January, 1995

[4] Chakrabarti, A. "Designing by functions", PhD thesis, Cambridge University Engineering Department, 1991

[5] Burgess, S., Moore, D., Edwards, K., Shibaike, N., Klaubert, H. and Chiang, H-S. "Design Application: The Design of a Novel Micro-Accelerometer" in Workshop on Knowledge Sharing Environment for Creative Design of Higher Quality and Knowledge Intensiveness, 12-13 January, 1995