

An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Introduction and Knowledge Representation

Amaresh Chakrabarti and Thomas P. Bligh

Engineering Design Centre, Department of Engineering, University of Cambridge, UK

Abstract. *Conceptual design is an early phase in the design process, which involves the generation of solution concepts to satisfy the functional requirements of a design problem. There can be more than one solution to a problem; this means that there is scope for producing improved designs if one could explore a solution space larger than is possible at present. Computer support to conceptual design could be effective to this end, if an adequate understanding of the required design knowledge and subsequent tools for its representation and manipulation were available.*

This three-part series of articles describes one approach to synthesis of solutions to a class of mechanical design problems; these involve transmission and transformation of mechanical forces and motion, and can be described by a set of inputs and outputs. The approach involves (1) identifying a set of primary functional elements and rules of combining them, and (2) developing appropriate representations and reasoning procedures for synthesising solution concepts using these elements and their combination rules; these synthesis procedures can produce an exhaustive set of solution concepts, in terms of their topological as well as spatial configurations, to a given design problem.

Part I provides an overview of the scope and the approach, adopted in the entire series, to identify the design knowledge required for synthesis, and a method for its validation. It specifically focuses on the extraction and representation of this knowledge. Part II describes synthesis of topological (graph structure) descriptions of possible solutions to a given problem. Part III describes a procedure for producing spatial configurations of these solutions.

Keywords. computer support, concept generation, conceptual design, knowledge representation and reasoning, mechanical design, transmission design, functional synthesis, functional reasoning, functional modelling

1. Motivation and the Problem

Conceptual design is considered the activity of transforming the functional requirements of a design

Correspondence and offprint requests to: Amaresh Chakrabarti, Engineering Design Centre, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.

problem into a solution concept or concepts for fulfilling the requirements. There can be more than one solution to a problem, and therefore scope exists for better, if not optimum, designs if a larger space of solutions can be explored.

Engineering design can be broadly divided into four phases, as is widely accepted in the field of design research (Hubka 1982; Pahl and Beitz 1984; French 1985; VDI 2221, 1985). In order of procedural progress, these are: task clarification, conceptual design, embodiment design, and detail design. Decisions taken during conceptual design, which is one of the earlier phases, affect all the downstream phases of design. The amount of information increases as a design becomes increasingly detailed, and therefore each decision taken upstream in the design process has a multiplied effect on the downstream phases.

Many consider the design activity as being non-rational (Gordon 1961; Bogen 1969; de Bono 1970; Ornstein 1972), while some consider it can be controlled and taught (Gordon 1961; de Bono 1970; Adams 1974). However, many thought processes can be recognised as rational after the event, although their origin might not be under full conscious control (Hubka 1982). Moreover, while the above approaches might improve a designer's skill, there would still remain many problems to designing optimum solutions. Designers often do not consider many concepts as potential solutions to a given problem, owing to reasons which include the following:

- their bias towards specific solutions;
- their lack of awareness of other solutions;
- the impossibility of manually considering more than a few solution alternatives within the given time constraint.

Catalogues of existing designs are available (Herkimer 1952; Beggs 1955; Hix and Alley 1958) to enhance the designer's awareness of the design

knowledge that is continually being extended. However, their accessibility is limited, because they only give information about the end product of design, i.e. its product description; this does not give any process information, i.e. information about how it was conceived in, and detailed to, this form.

Research into a systematic approach to design, as is stressed by numerous authors (Matousek 1963; Glegg 1969; Pitts 1973; Yoshikawa 1981; Pahl and Beitz 1984), which could provide designers with a means of progressing from a design problem to its solutions, allowing an extensive and objective exploration of the *solution space* without restraining their creativity, is important. This should include:

- an objective (i.e. solution-neutral) definition of the design problem;
- an unbiased generation of possible solution alternatives;
- an objective (comparative) evaluation of the solution alternatives.

Even if methods for objective generation and evaluation of solutions were available, the problem of the lack of knowledge about existing design solutions and about other databases, and limited information-handling ability, would still limit designers from using these methods effectively. Computer-aided design can be important from this respect. Apart from providing a unifying framework for supporting the two main components of design, i.e. knowledge (previously in books, catalogues, etc.) and operational tools (previously the drawing equipment, calculating machines, etc.) (Coyne et al. 1990), on which a better information management could be effected, it would improve the possibility of generating and examining a much larger number of solutions than is possible at present. Computers have already proved useful in the later and post-design activities, which include drafting, surface and solid modelling, graphics, analysis, and scheduling and process-planning tasks (Besant and Lui 1986). However, they are still very primitive in supporting the high-level decision-making activity involved in the earlier and ill-informed stages of design, of which conceptual design is one.

In this series of three articles, an approach to identifying the knowledge required to represent the functional requirements of a specific class of design problems (mechanical transmission problems, i.e. problems involving mechanical forces and motion) and their possible solutions is described, and computational methods developed for solving (synthesis of solution alternatives) part of these problems are presented. The discussion involves issues concerning

the extraction, representation, and manipulation of the above design knowledge, within a specified framework for validation of this knowledge. The discussion does not include comparative evaluation of solution alternatives.

Conceptual design is characterised by reasoning in terms of functions. Functional reasoning approaches (Freeman and Newell 1971; Yoshikawa 1981, 1985; Grabowski and Benz 1988, 1989; Schmekel 1989; Chakrabarti and Bligh 1991), which allow problems and solutions to be described in terms of their functions, and allow reasoning about them, can thus be important in this phase. The work described here is entirely from the functional point of view. The synthesis procedure described, therefore, is named functional synthesis. The procedure is an embodiment of the scheme proposed in Chakrabarti and Bligh (1991), where the reasons for preferring this to other schemes is elaborated.

In Part I the structure of design problems and their (existing) solutions is discussed within a specified framework for validation, and constructs required to represent them are extracted. In Part II a method for synthesis of graph-structures of solution concepts which satisfy the "kind requirements" of the problem is presented. Part III discusses a method for producing spatial configurations of these concepts that are able to satisfy the orientation and sense requirements.

2. The Functional Reasoning Approach

The word "function" is regarded here as a description of the action or effect (intended to be) produced by an object, i.e. what it (is intended to do or) does. A "functional representation", therefore, should allow one to describe objects (which in the design context are design problems and solutions) in terms of their known functions. For example, the function of a shaft can be described as: *a shaft transmits torque*. Here *shaft* is an object, and its function is to *transmit torque*.

The idea of functional reasoning in conceptual design is to reason at the functional level in order to generate solutions to specified design problems, and to evaluate given solutions for suitability to specified problems. The term "functional synthesis" is used here to refer to the "generation" part of functional reasoning. Each model for functional reasoning consists of two parts:

- a functional representation of the objects to be reasoned about, and
- a reasoning scheme.

There are two existing functional representations.

One is a natural-language-like, non-mathematical representation, where verbs are used to describe what an object does, or is supposed to do (Freeman and Newell 1971; Johnson 1988; Lai and Wilson 1989). An example would be this informal description: a *shaft* (object) *transmits torque* (function). An advantage of this representation is that it is close to the way designers express their ideas. However, in general, natural language lacks precision and, in a sense, objectivity. It is difficult to formalise this representation in a generalised way, and there are many compound functions for which a standard name such as *transmit* does not exist.

The other representation is a mathematical representation of function, where it is expressed as a transformation between input and output. It is formalisable, and therefore is more suitable for a computational environment. However, if a man-machine environment is to be provided, using this representation, the commonly used functions expressed in the first representation would have to be mapped into the latter representation before any general functional reasoning support environment could be developed. From here on, the word function will be used to describe any function covered by the above representations, with the assumption that the objective definition of each such function is precisely known.

3. An Overview of the Framework for Knowledge Extraction to Validation

The work reported in this three-part paper is focused, as a starting point, specifically on mechanical transmission designs. We first analyse a set of existing designs to extract a set of fundamental power transmission elements and the ways in which they can be combined, and develop their representations which are amenable to computer manipulation; we then have a set of building blocks adequately represented such that they can be used to create not only the concepts of the systems from which they were extracted in the first place, but also a range of new concepts. To give an overview of how the extraction, representation,

manipulation and validation of design knowledge would be carried out, let us suppose we have analysed a set of problems and hypothesised a set P of representation constructs as adequate for representing these problems. Suppose we have also analysed the power flow paths (sequence of elements through which power is transmitted between the inputs and outputs) of a set of solutions corresponding to the above problems, to hypothesise and represent their parts as a set S of structures, and their connections as a set R of ways of combining the elements of S . This constitutes the *extraction* and *representation* of knowledge. Now we can write procedures, which would use R on S to produce a set of solutions to any of the above problems in P . This constitutes the *manipulation* of the knowledge. If now the above procedures, for each known problem that the knowledge was extracted from, can generate at least its corresponding known solutions and preferably others, using S and R alone, then we consider the knowledge to be *validated*, where the knowledge constitutes those which were extracted and represented, as well as that residing in the procedures.

4. An Overview of the Problem, the Nature of its Solutions, and the Overall Problem-Solving Approach

In functional reasoning, the idea is to express a design problem and its solutions in terms of a *common language*, based on their functional representation; this then enables one to generate, compare and modify problems and solutions.

In mechanical transmissions, the design objective is to transmit and transform forces and motions having various characteristics which may change with time. In functional terms, the design objective may be considered as a transformation between the temporal characteristics of given input variables and that of the required output variables (Fig. 1). This objective constitutes the intended temporal function of the product to be designed. This transformation is



Fig. 1. The general transmission design problem.

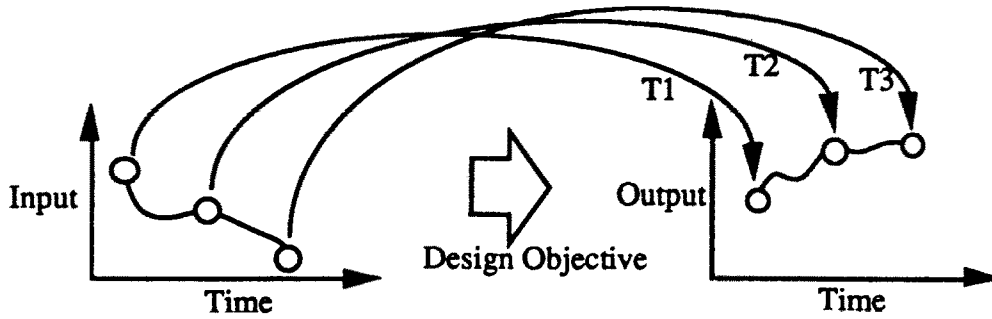
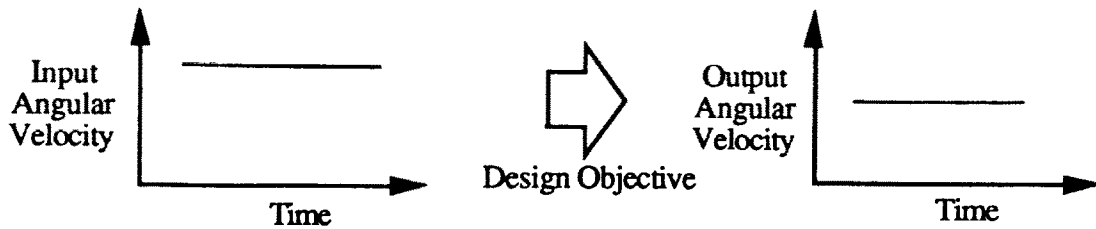
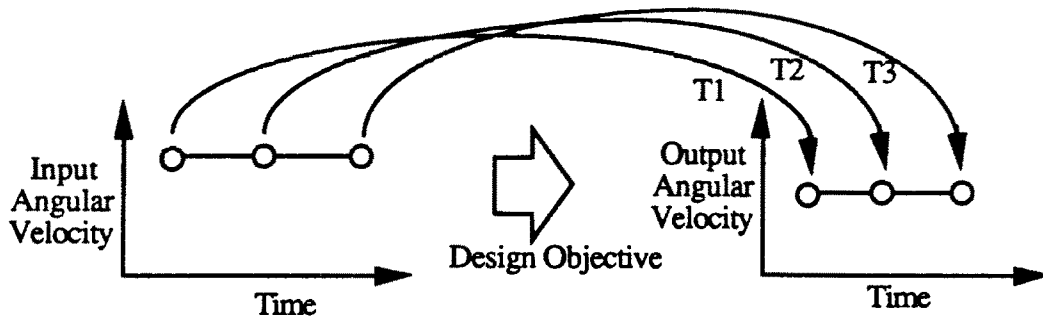


Fig. 2. The transmission problem in Fig. 1 is equivalent to the ordered set of transformations T1 then T2 and then T3.



(a)



(b)

Fig. 3. The transmission problem in Fig. 3(a) is equivalent to the ordered set of transformations T1 to T2 to T3. (a) An example transmission design problem. (b) T1, T2, T3: instantaneous input–output transformations.

equivalent to an ordered set of input–output transformations, each of which occurs for an *instant* of time (Fig. 2). For instance, one design problem may be expressed as a transformation of a constant angular velocity at a given position in a given direction into another constant angular velocity of a smaller magnitude at a different position in the opposite direction. The problem is shown in Fig. 3(a), which is equivalent to an ordered set of instantaneous transformations T_1 , T_2 , T_3 (Fig. 3(b)).

A transmission design solution, depending on the level of design detail, is an abstract or a detailed concept of one, or a set of, physical structures combined in an orderly way, so that it produces at least the outputs required by the design objective, when specified inputs are provided. A solution usually

consists of the ideas of physical entities having three kinds of functions: structures that take the primary role in transforming and transmitting forces and energy, structures that couple or connect, and, structures that take the unwanted forces away. For example, one solution to the problem in Fig. 3 could be a system consisting of an input pinion (small gear) and an output gear driven by the pinion (Fig. 4). Here the gears are the primary structures which transform energy, the meshing of the gear teeth provides the joint, and the bearings provide the supports.

The problem-solving approach taken here is to solve a given design problem in the following three steps:

1. Find solutions, only in terms of structures that actively contribute to the required energy transformation, which would satisfy one of the

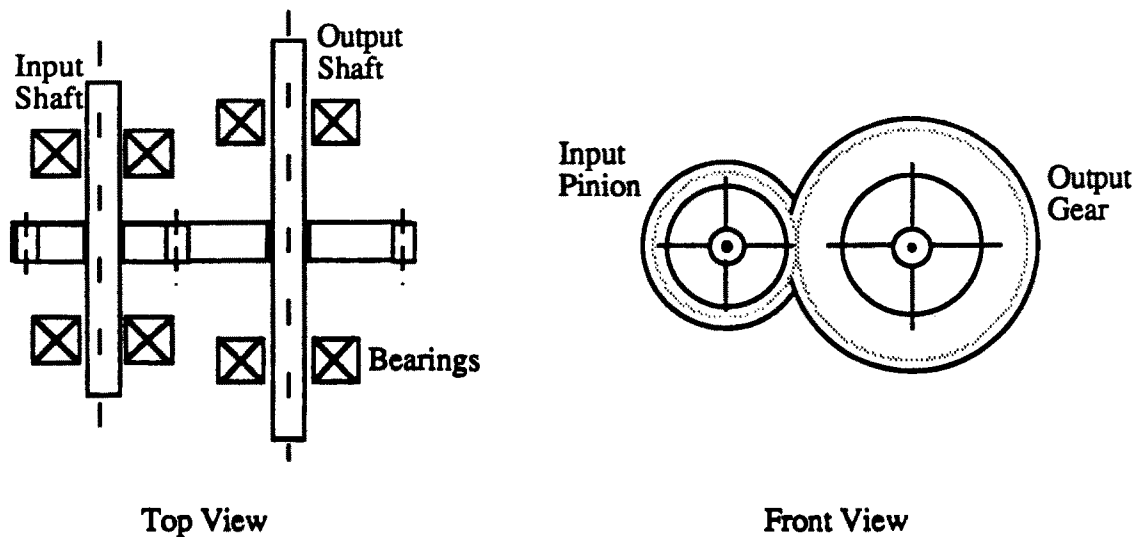


Fig. 4. A solution (a gear-pair) to the design problem shown in Fig. 3.

instantaneous transformations of the problem. This forms the *horizontal problem redefinition*, where only those solutions are synthesised which can satisfy at least an instantaneous function (i.e. a part) of the problem.

2. Choose joints and supports to enable the above structures to provide the above instantaneous function. This forms the *vertical problem redefinition*, where auxiliary functions required by the above structures are chosen.
3. Carry out *temporal reasoning* about the above solutions to ensure that they satisfy the complete temporal function required by the problem. This is a framework which should provide the scope for exploring how well the concepts generated in the above steps function, leading to their evaluation and modification.

In these papers, representations and reasoning procedures for supporting horizontal problem redefinition (i.e. instantaneous synthesis) are described. A set of basic constructs for describing instantaneous requirements of problems and solutions are extracted and represented, procedures are described for synthesising solutions to given problems using the above knowledge, and how this knowledge is validated is discussed. The other two steps, i.e. vertical redefinition and temporal reasoning, are under development.

5. Related Work

Related work in synthesis of mechanical devices includes Ulrich and Seering (1989), Hoover and

Rinderle (1989), Finger and Rinderle (1990), Kota and Chiou (1992), and Prabhu and Taylor (1988, 1989). Except for Kota and Chiou, all of the above use a bond graph representation; for information on bond graphs, see Paynter (1961) and Rosenberg and Karnopp (1983).

The work of Prabhu and Taylor is the closest to the work presented in this article. Like us, they use a set of functional primitives as building blocks which have orientations, magnitudes and positions as distinct characteristics, and propose to produce functional networks of these primitives as solutions to multiple-input-output problems. On the basis of a formal mathematical representation (bond graph), these authors provide a number of fundamental theorems regarding generic algorithms to build systems using a chosen spanning set of functional primitives, maximum and minimum bounds on their complexity, etc.

However, the present work is different from the above in a number of ways. First, unlike in Prabhu and Taylor, the scope of the present work is not only for constant input-output effort and flow variables, but also for systems with time-varying inputs and outputs; see the approach to problem solving in Section 4. Second, the representation in the present work consists of a set of possible topological and spatial relationships between the inputs and outputs of its functional primitives sufficient to represent mechanical designs, while in Prabhu and Taylor these primitives are essentially bond graph elements, and thus unlikely to be able to generate alternative spatial configurations of the solutions generated. Third, their approach to problem solving is a successive refinement process in terms of a composition of mappings (scalar rendering, scalar design and vector modification), and

is different from the present approach where topological synthesis of solutions is followed by successive spatial constraint propagations on them, so as to generate their alternative spatial configurations. Prabhu and Taylor solve the scalar rendering problem first, and then introduce new elements to solve the vector problem, but in the present approach, topologies of solutions are synthesised once and for all and are then checked for feasibility during the spatial configuration steps. Also, in Prabhu and Taylor heuristics are needed for minimising the number of elements in the solution alternatives, but the algorithm in the present work obviates this need by taking this as a parameter to be specified by the designer. Finally, although Prabhu and Taylor discuss a number of properties for a potential algorithm for doing synthesis, the algorithm is not provided, and nor are its implementation and results, so that little scope is given for their evaluation.

Kota and Chiou use a matrix-based function representation for the problem as well as the building block mechanisms, and use a set of matrix transformation rules to concatenate the building blocks to solve a given problem. However, the present approach, with motion transformations as its functional primitives, attempts to support synthesis at a more fundamental level. Ulrich and Seering use a set of rules from control theory in order to produce topological descriptions of single-input–output transducer designs. All the other researchers mentioned above use a set of “behaviour-preserving transformations” to transform bond graph descriptions of device behaviour into topological descriptions of gear systems. The work described in this paper differs from that of these researchers in at least three ways:

- The synthesis procedure adopted here is founded on a component-based compositional approach, rather than a “transformational” or “design and debug” approach.
- The synthesis procedure applies from single-input–output to multiple-input–output systems, and can be exhaustive. It is based on the development of distinct graph structures which provide the required function.
- The procedure produces topological as well as spatial configurations of designs.

Kusiak et al. (1991) describe a systems-theory-based synthesis approach which uses a rule-based, object-oriented programming paradigm; these rules are applied repeatedly to produce increasingly compound structures, such as identifying the supports and joints required in a mechanical vice. The representation is

too abstract (i.e., devoid of spatial information) to reason especially about auxiliary functions of mechanical devices in general, and the approach seems wasteful in that it produces compound structures that do not satisfy the intended function of the problem. However, with appropriate representations, the approach should be more suitable for “vertical problem redefinitions” than for “horizontal redefinitions”.

It should be noted that the approach taken here is also considerably different from that taken by the classical “mechanisms” approach (Reuleaux 1963; Harrisberger 1965; Woo 1967; Freudenstein and Maki 1983; Hoeltzel et al. 1987; Hoeltzel and Chieng 1990). In the mechanisms approach, a set of connections, or “kinematic pairs”, as they are described in the mechanisms terminology, are topologically combined to provide the required “degrees of freedom” (i.e., relative mobility between connected parts) at specified input–output points. The “links” connect two or more of these kinematic pairs. In the present approach, on the other hand, the first step is to combine structures to form the essential concept, and only then are the joints and supports (i.e., connections) chosen to enable the structures to function in the ways stipulated.

There are several reasons why the mechanisms approach does not seem to be the appropriate choice in this case. First, while the approach could be very effective in the design of complex mechanisms, it is not generalisable and integrable with designs in other domains. Second, it does not support problem decomposition leading through a progressive movement from the general to the specific (as it chooses structures, joints and supports, all at the same time), and therefore it increases the possibility of “combinatorial complexity”. Moreover, it does not provide individual functional reasons as to why each element (i.e., a structure, a joint or a support) should be there in a solution concept, and hence makes the control of design decisions difficult.

6. Imposed restrictions

6.1. Restriction on the Rules of Combination

The synthesis problem is restricted to considering only those primary structures which have a single input type and a single output type, and a single input point and a single output point (Fig. 5). The rule of combination is restricted to combining two entities by connecting one input–output point of one entity with one input–output point of the other, only when the characteristics of the input–output variables at the

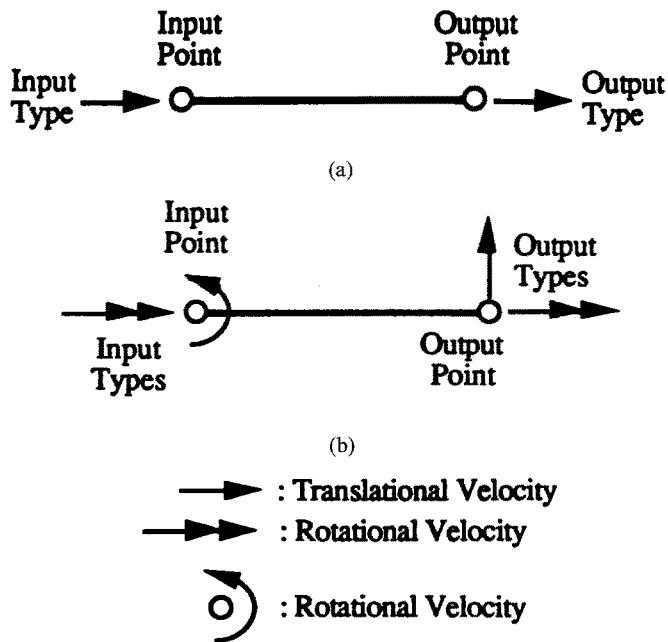


Fig. 5. Structures having single and multiple input-output types. (a) A single-input-type-single-output-type primary structure (a tie-rod). (b) A multiple-input-type-multiple-output-type structure (a part of a flexible shaft).

connection are the same. However, in general, a primary structure could have multiple input or output types at an input or output point. It is possible to construct these latter entities from the former ones using various other rules of combination (Chakrabarti 1991). However, these primary structures are not considered in the synthesis problem tackled in this article. Multiple-input-output problems, discussed here, are defined by single-input-output types at each of their input-output points. These are solved by generating, as solutions, networks of single input-output type single input-output point primary structures, such that the intermediate connections of these networks have more than one primary structure join to share inputs and outputs of the same type.

6.2. Orthogonality Restrictions

The present synthesis problem is restricted to considering those configurations (i.e., spatial arrangements) which obey the orthogonality restrictions. There are two orthogonality restrictions:

- *Orthogonality Restriction 1* The input and output vectors of a primary structure can be either parallel or perpendicular to its length vector, where a length vector is defined as a vector having a magnitude equal to the distance between the input and the output points, and is directed from the input point towards the output point.

- *Orthogonality Restriction 2* The input and output vectors of any primary structure should be either parallel or perpendicular.

The necessity for introducing these restrictions arises from the fact that the number of potential spatial arrangements of a design solution could, in many instances, be infinite. For instance, a lever arm, for a specific input rotation at a specific position in space, could be placed along any radius of a circle having its centre at the position of the rotation and in the plane perpendicular to the rotation vector. Any attempt to write a procedure for the exhaustive generation of spatial arrangements for a solution concept containing such possibilities would produce combinatorial problems, and hence the restrictions.

7. Extraction of Knowledge

A transmission design problem can be viewed as an input-output transformation. The design process can be viewed as another transformation which transforms a design problem into its possible solutions (Fig. 6). In this research, known simple devices are analysed from the point of view of their input-output in order to identify the primary solution structures and their contribution to the overall transformation. Following the power flow path from the known input to the output point, the motion transmission elements are identified as the primary structures, because they carry energy. These structures can be viewed as transformers which transform some characteristics of the input-output variables. For instance, in Fig. 7 the structure C transforms only the position of the input at p_3 to the output at p_4 , while the structure A transforms both the position (from p_1 to p_2) and the kind (from force to torque) of the input-output variables. Note that when two structures are connected to each other, the energy-receiving structure takes the output of the other structure as its input.

To summarise, primary structures are transformers which can transform various characteristics of the input-output energy variables. The rules of combination of these transformers are conjectured to be as follows:

1. Two structures can be connected only by connecting the input of one structure with the output of the other.
2. A connection is possible only when the input involved in the connection has the same characteristics as the output involved.

Note that the inputs and outputs can be more than

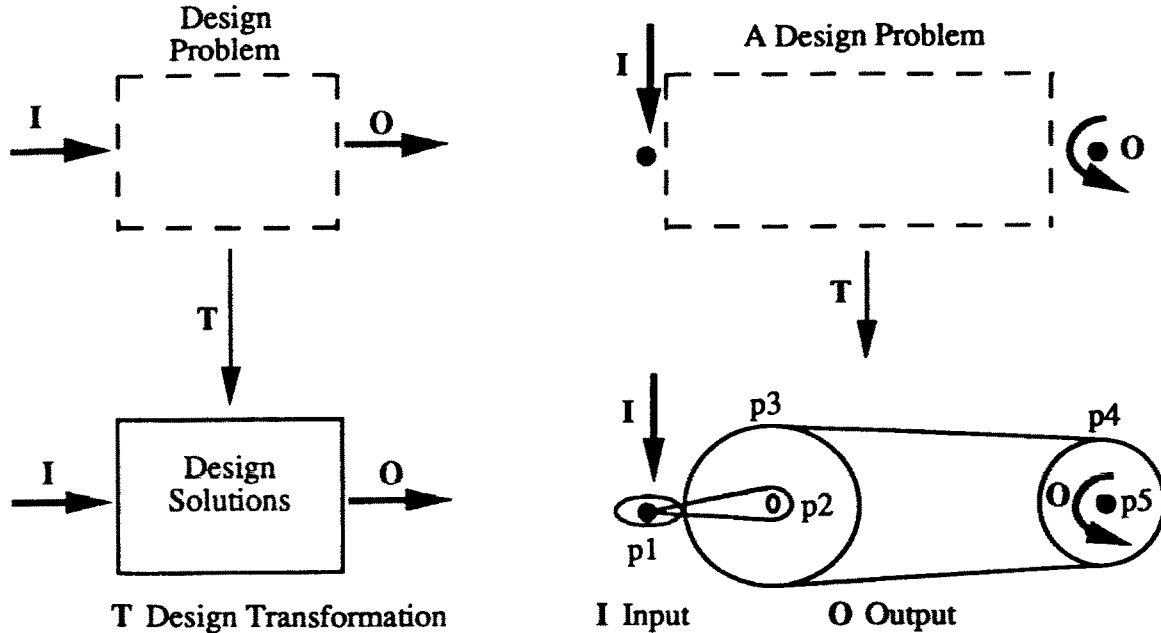


Fig. 6. Single-output-single-output (SISO) design transformation. Left: the general SISO transformation. Right: an example (a bicycle drive).

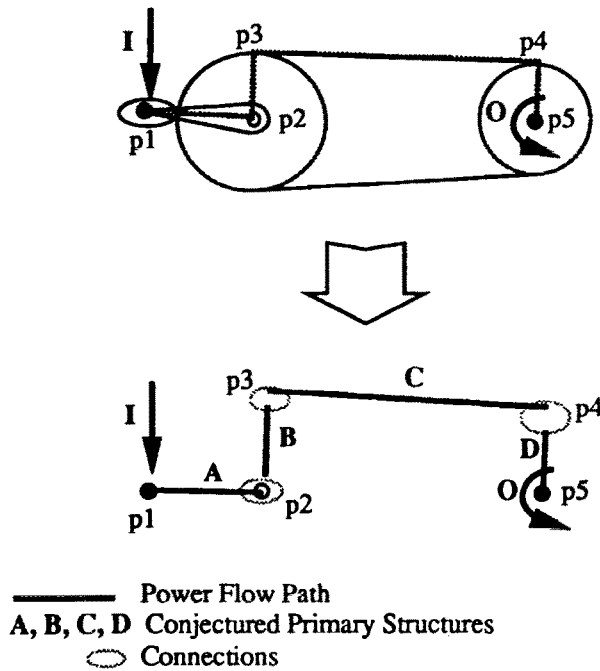


Fig. 7. Extraction of design knowledge.

one. In general, a connection in a multiple-input-multiple-output (MIMO) system is such that the energy-receiving structures take, as inputs, the summation of the outputs from the other structures. The general rules of combination for a system to be valid (i.e., to be feasible in principle) can be summarised as:

1. A set of structures can be connected at a common

connection only by connecting the outputs of one of its subsets with the inputs of the rest.

2. A connection is possible only when the input-output involved in the connection have the same characteristics.

Now, given a system of connected structures, where the constituting transformations of each structure (which structure does what) and the connections (which structures' input is connected to the output of which structures) involved are known, it is possible to analyse that system to:

1. check whether or not it is a valid system;
2. identify, for a valid system, its transformations, i.e. the transformation between the characteristics of its input-output variables.

“Function”, as used in this paper, is defined as a transformation between a set of input characteristics and a set of output characteristics. The above conclusions imply that, given the functions of the constituting structures of a system, and the rules of their valid combination, the function of the system so constituted can be deduced. Taking the bicycle drive in Fig. 7 as the example, we could define structures A, B, C and D as four transformers, whose functions are as follows:

- Structure A transforms an input force at a specific position p_1 into an output torque at a different position p_2 .
- Structure B transforms an input torque at p_2 into an output force at p_3 .

- *Structure C* transforms an input force at p_3 into an output force at p_4 .
- *Structure D* transforms an input force at p_4 into an output torque at p_5 .

Now, the bicycle drive in Fig. 7 could be described as a system formed by connecting the output of structure A to the input of structure B, the output of structure B to the input of structure C, and, the output of structure C to the input of structure D. Using the rules of combination enables the system to be verified as valid. Similarly, we can use the functions of the above-mentioned structures in conjunction with the specified combination rules to deduce the overall function of the system, which is that of transforming an input force at a specific position p_1 into an output torque at some other position p_5 .

The above method allows us to deduce the overall function of a given system, using the information about its constituent structures, connections, the functions of the constituent structures, and their rules of combination. Now we seek a formal representation of this knowledge, so that procedures embodying the functional reasoning scheme proposed by Chakrabarti and Bligh (1991) can be written to synthesise systems, using the represented knowledge, for solving given functional requirements.

8. Representation of Knowledge

This is a part of the design knowledge discussed in Section 3, and consists of constructs for representing design problems and solutions, but not the procedures used to produce solutions from problems.

8.1. Representation of Design Problems

An instantaneous *multiple-input-multiple-output* (MIMO) design problem can be viewed as a transformation between the characteristics of a set of instantaneous input vectors and output vectors (of which *single-input-single-output* (SISO), *single-input-multiple-output* (SIMO) and *multiple-input-single-output* (MISO) systems are special cases). A vector would have a kind, an orientation in space, a sense of its orientation, a magnitude and a position in space associated with it (Fig. 8). Within the confines of the orthogonality restrictions introduced in Section 6.2, the constructs for representing a MIMO design problem involving a transformation between m inputs and n outputs are:

input-1 kind: (*force/torque/linear motion/angular motion*)

orientation: ($i/j/k$) [$i/j/k$ are unit vectors in a Cartesian coordinate system]

sense: (+/−)

magnitude: (some number)

position: ($x_1i + y_1j + z_1k$)

input-2 kind: ...

orientation: ...

...

...

input- m ...

...

output-1 kind: (*force/torque/linear motion/angular motion*)

orientation: ($i/j/k$)

sense: (+/−)

magnitude: (some number)

position: ($x_{m+1}i + y_{m+1}j + z_{m+1}k$)

output-2 kind: ...

orientation: ...

...

...

output- n ...

...

8.2. Representation of Solutions

As the input and output are vectors having specific characteristics, the solution structures are vector transformers which transform a set of input vectors into a set of output vectors. The input–output points of the vectors associated with a transformer specify their positions in space, and the spatial separation between these input–output points becomes the position transformation by the transformer. The input–output vectors of a structure are related by various physical principles, which determine the relative orientations, senses, and magnitudes of the vectors involved. All these lead to a representation (Fig. 9) in which a vector transformer is represented by a 3-tuple of vectors, i.e. an input vector (I-vector), an output vector (O-vector), and a length vector (L-vector). The length vector is defined as a vector joining the input point to the output point, and is directed from the input point to the output point. This vector is created to explicitly reason about the position changes involved in a solution. An I-vector or an O-vector has a kind, orientation, sense, magnitude and position, while an L-vector has a position and orientation (given by the position of the I-vector, and the line joining the positions of the I- and O-vectors), sense (directed from the input point towards the output point), and magnitude (spatial separation between the input point and the output point). These

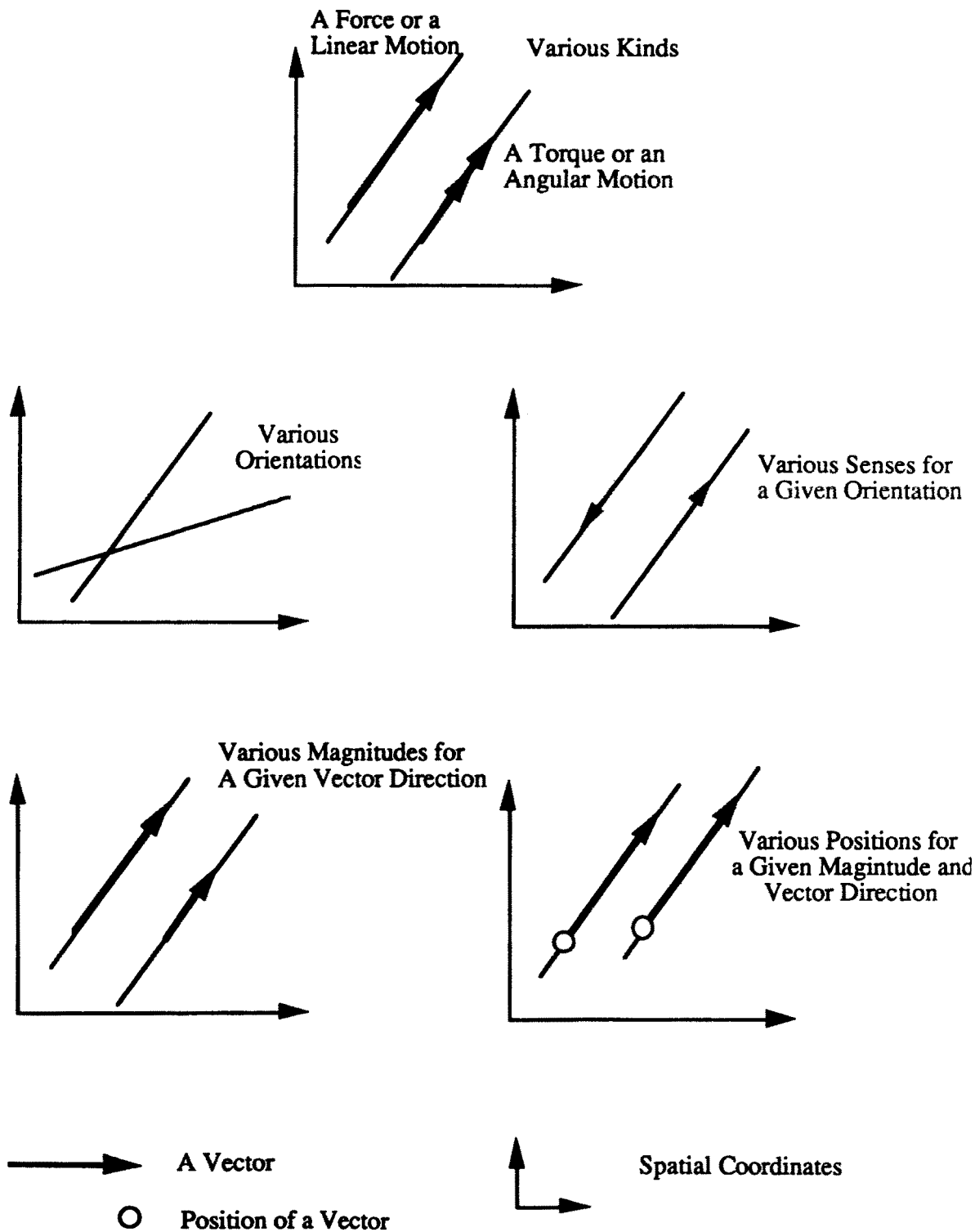


Fig. 8. Various characteristics associated with an input-output vector in mechanical transmission design.

characteristics are variously coupled, depending on the characteristics of the specific transformer involved.

Mathematically, the spatial relation between two vectors can be expressed by using a combination of

two properties: whether the lines of their action are parallel (P), and whether their lines of action intersect (I). Using the combination of these two properties (and their negatives), we find the four possible spatial



- I/O Point
- I Input Vector
- O Output Vector
- L Length Vector

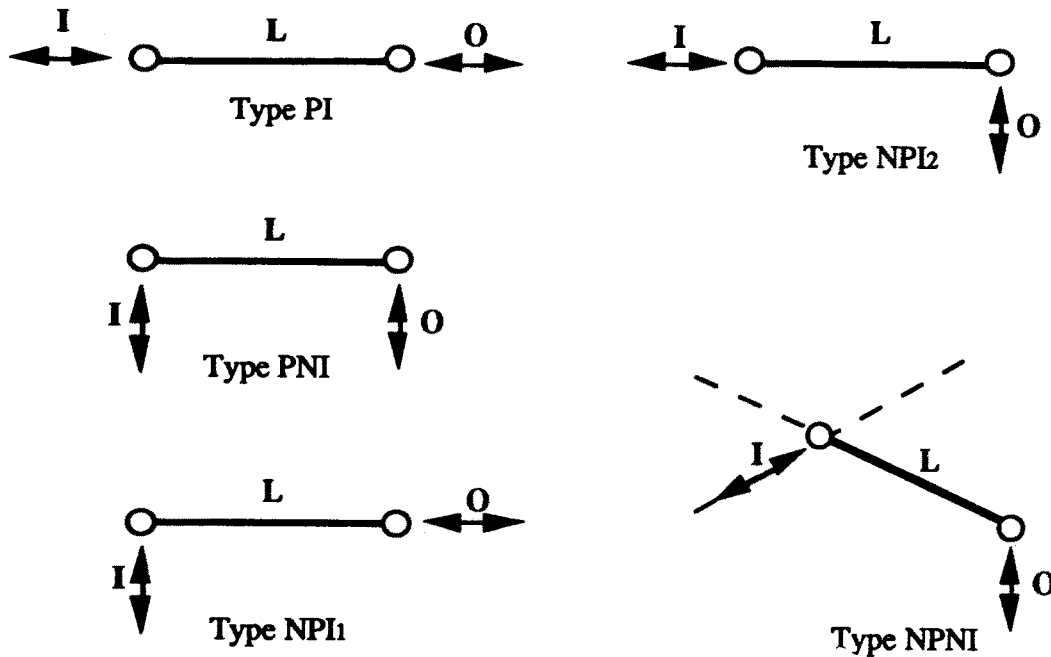
Fig. 9. Representation of an SISO solution structure (transformer).

relations; these are: parallel and intersecting (PI), parallel and non-intersecting (PNI), non-parallel and intersecting (NPI), and non-parallel and non-intersecting (NPNI).

Using the spatial relations among its vector characteristics, a known structure can be typified into one, or a combination, of the types shown in Fig. 10. So, a shaft would be a structure of type PI (Fig. 10) having torque (and/or angular motion) as the input and output kinds which are parallel and intersecting.

A lever would be a structure of NPNI type, having torque and force (or vice versa) as the input and output kinds respectively.

This representation gives us the constructs to capture the orientation and sense transformations between the input–output vectors of a transformer. An orientation transformation of a transformer would be a 3-tuple of vector orientations representing a valid combination of the orientations of its I-vector, L-vector and O-vector. For example, a shaft can have a valid orientation transformation (*iii*) where all the vectors would be oriented along the *i*-vector. A sense transformation for a specific orientation transformation of a transformer would be similarly represented by a 3-tuple containing a valid combination of the senses of its I-, L-, and O-vectors corresponding to their specific orientations. A shaft, for instance, can have a sense transformation (+ - +) which, in conjunction with the orientation transformation, would mean that its I-, L- and O-vectors would be in the positive *i*-, negative *i*-, and positive *i*-directions respectively. The shaft, for the same orientation transformation (*iii*), could also have three other sense transformations



- I/O Point
- I Input Vector
- O Output Vector
- L Length Vector

Fig. 10. Various possible spatial relations between the input–output vectors of a transformer.

(+ + +), (- + -) and (- - -). The above four transformations describe that fact that for a shaft, which is oriented along the *i*-axis (thus having its input-output also oriented along the same axis), its input and output have the same sense, and this is irrespective of which end of the shaft is considered its input or output.

Once the type of orientation transformation for a structure (a transformer) is known (say parallel and intersecting), the valid orientation transformations for the structure can be computed by considering the three possible space coordinates. Then, the valid sense transformations can be found by first finding the combinatorially possible sense transformations (2³ possibilities, as each of the three vectors can have a choice of a +/- sense), and then identifying the valid sense transformations which satisfy the natural constraints. For instance, for a mechanical lever, with its input rotation, length, and output motion oriented respectively along *i*-, *j*- and *k*-axes, a positive rotation for a positive length vector would necessarily produce a positive motion at its output; thus (+ + +) is, and (+ + -) is not, a valid sense transformation for a lever with the orientation transformation (*ijk*). The orientation and sense transformations for a mechanical lever are shown in Fig. 11.

In general, a MIMO vector transformer can then be represented as shown in Fig. 12, in terms of a 3-tuple vectors, i.e., a set of input vectors, a set of output vectors, and a length vector. The input-output vectors in the above input-output vector-sets are the components of the input-output vector at the input-output that are taken/contributed by various transformers connected to that input-output point.

The position transformation of a transformer is given by the characteristics of its length vector(s). This is represented by the following equation:

$$\begin{aligned} \text{position transformation} &= \text{position (O-vector)} \\ &\quad - \text{position (I-vector)}. \end{aligned}$$

So the position transformation will be known if the information about magnitude and direction of the L-vector is available (or the positions of the O- and I-vectors of the transformers are known).

If two transformers are to be joined then the rule for their combination is given by the requirement that the joining input-output vectors have to be at the same position:

$$\begin{aligned} \text{position (O-vector)}_{\text{one structure}} \\ = \text{position (I-vector)}_{\text{other structure}} \end{aligned}$$

The magnitude transformation for a structure is defined as the ratio between the magnitude of the O-vector and that of the I-vector, and is governed by physical principles or constraints, which include the law of conservation of energy. For example, the magnitudes of the input force and the output torque (or vice versa) are related by the L-vector. In the case of a shaft, this ratio is 1, i.e. the input and output torques are of the same magnitude.

The magnitude transformation is expressed here by a magnitude transformation factor which is defined as magnitude transformation factor

$$= \text{magnitude (O-vector)/magnitude (I-vector)}.$$

In the most general case the rule of combination at a connection is defined as

$$\begin{aligned} \sum \text{input effort magnitudes} \\ = \sum \text{output effort magnitudes,} \\ \text{and} \\ \text{each input flow magnitude} \\ = \text{each output flow magnitude.} \end{aligned}$$

Here, *effort* is used to denote the force-like components, and *flow* is used for motion-like components of energy, as in bond graph notations (Paynter 1961; Rosenberg and Karnopp 1983).

Once the various transformations (kind, orientation, sense, position and magnitude) for a set of structures are found and stored, these can be used to compute the various transformations for a compound structure which is formed as a combination of structures from the above set. For example, a crank can be defined as a combination of a shaft and a lever such that the torque output of the lever is taken as the input of the shaft (Fig. 13). So, for the crank:

kind transformation:	force ↔ torque
orientation transformation:	(<i>ijk k k</i>), (<i>ik jjj</i>) ...
sense transformation:	(+ + + + +), (+ + + - -) ...
position transformation:	L-vector (lever) + L-vector (shaft)
magnitude transformation:	magnitude (O-vector) _{shaft} /magnitude (I-vector) _{lever}

9. Summary and Conclusions

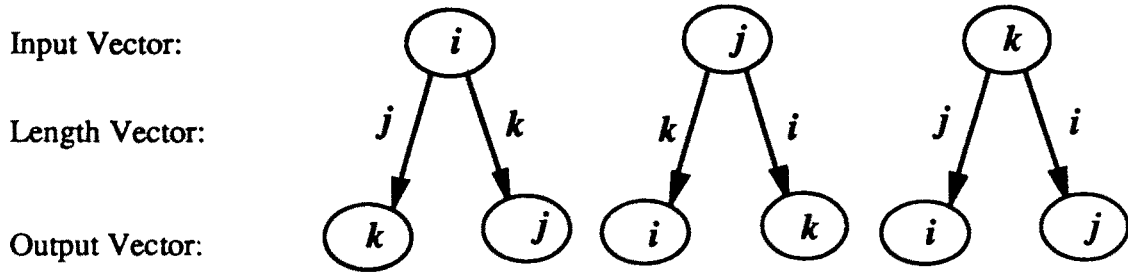
Conceptual design is an essential activity in design, and an appropriate computational support could help

Transformer: Mechanical lever

Kind Transformation: Force and/or Linear Motion <-> Torque and/or Angular Motion

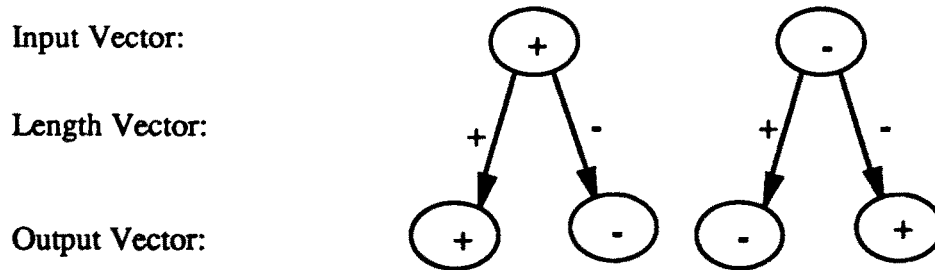
Type of Orientation Transformation: Non-parallel, Non-Intersecting (NPNI)

Valid Orientation Transformations:



- Valid Orientation Transformations:**
1. (ijk)
 2. (ikj)
 3. (jki)
 4. (jik)
 5. (kji)
 6. (kij)

Valid Sense Transformations for the Orientation Transformation (i j k):



- Valid Sense Transformations:**
1. $(+++)$
 2. $(+--)$
 3. $(-+-)$
 4. $(---)$

Fig. 11. Representation of the knowledge of valid orientation and sense transformations for a mechanical lever.

produce improved designs. In this three-part article, the need for computer aids in this phase is emphasised, and an approach to the extraction, representation and manipulation of design knowledge for functional

synthesis of mechanical systems, within a specified framework for validation, is discussed. The approach is based on a functional reasoning scheme described by Chakrabarti and Bligh (1991); the strategy is to

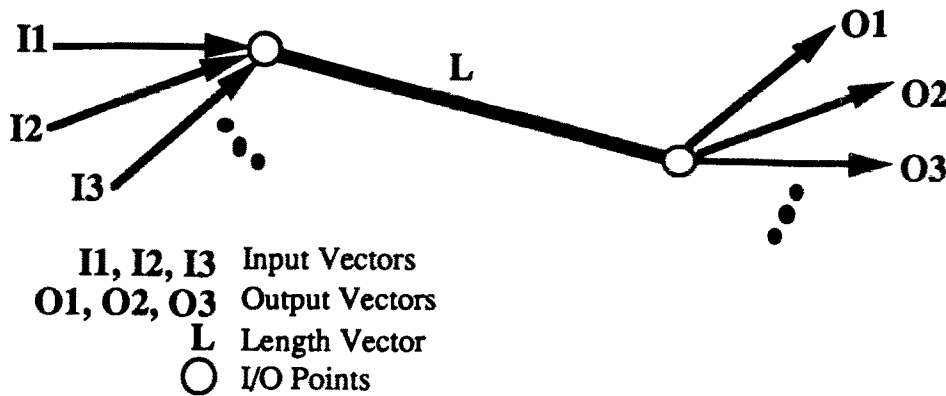


Fig. 12. Representation of a MIMO structure (transformer).

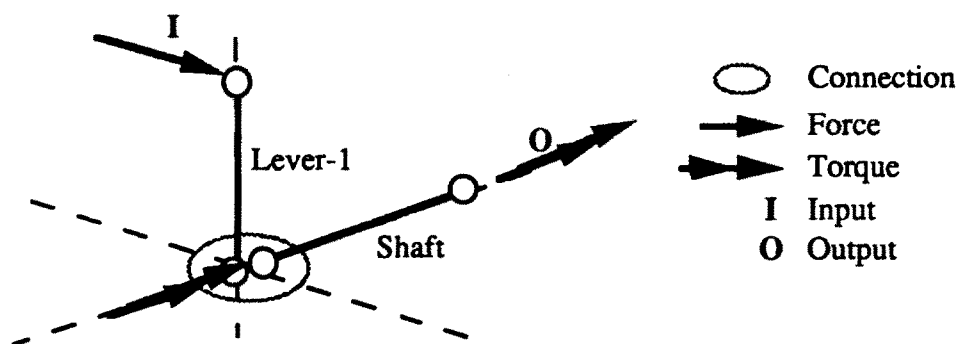


Fig. 13. A lever and a shaft connected in a series forms a crank.

solve the problem in three steps, i.e. “horizontal redefinition”, “vertical redefinition”, and temporal reasoning, of which the first step is discussed in this article.

Part I describes extraction and representation of knowledge about design problems and solutions under consideration, which consists essentially in identifying constructs for representing the functions of the problems and the primary structures that constitute their existing solutions. Parts II and III involve a description of the synthesis and spatial configuration procedures for generation of solutions to problems using the above knowledge.

Acknowledgements

The authors thank all the reviewers for their thoughtful comments on the manuscript of this article; these helped improve both its content and its readability. Amaresh Chakrabarti wishes to acknowledge the Nehru Trust for Cambridge University, India, the Cambridge Philosophical Society, Cambridge University Engineering Department, the Trustees of the Lundgren Fund of Cambridge University, the Cambridge Engineering Design Centre, the Northbrook Society, Darwin College, the Gilchrist Educational Trust, and the Leche Trust, for financial support. This work formed part of a PhD thesis at the University of Cambridge between 1987 and 1991 (Chakrabarti 1991).

References

- Adams, J. L. *Conceptual Blockbusting, A Guide to Better Ideas*, London and New York: W. W. Norton, 1980, 2nd edn (1st edn 1974).
- Beggs, J. S. *Mechanism*, McGraw-Hill, New York, 1955, ch. 13.
- Besant, C. B. and Lui, C. W. K. *Computer-Aided Design and Manufacture*, London: Ellis Horwood, 1986.
- Bogen, J. E. The Other Side of the Brain: An Oppositional Mind, *Bulletin of the Los Angeles Neurological Society*, 1969.
- Chakrabarti, A. *Designing by Functions*, PhD Thesis, Department of Engineering, University of Cambridge, 1991.
- Chakrabarti, A. and Bligh, T. P. Towards a Decision-Support Framework for Mechanical Conceptual Design, *Proceedings ICED, Zurich, 27–29 August 1991*, vol. 1, pp. 384–9.
- Coyne, R., Roseman, H., Radford, A., Balachandran, M. and Gero, J. S. *Knowledge-Based Design Systems*, Sydney, Addison-Wesley, 1990.
- De Bono, E. *Lateral Thinking: A Textbook of Creativity*, Penguin, London, 1970.
- Finger, S. and Rinderle, J. R. A Transformational Approach for Mechanical Design Using a Bond Graph Grammar, *EDRC Report no. 24-23-90*, Carnegie-Mellon University, Pittsburgh, 1990.
- Freeman, P. and Newell, A. A Model for Functional Reasoning in Design, *Proceedings Second International Joint Conference on Artificial Intelligence, London*, pp. 621–33, 1971.
- French, M. J. *Conceptual Design for Engineers*, London: The Design Council, 1985.
- Freudenstein, F. and Maki, E. R. Development of an Optimum Variable-Stroke Internal-Combustion Engine Mechanism from the Viewpoint of Kinematic Structure, *ASME Journal of Mechanisms, Transmissions and Automation in Design*, vol. 105, pp. 259–66, June 1983.

- Glegg, G. L. *The Design of Design*, Cambridge University Press, Cambridge, England, 1969.
- Gordon, W. J. J. *Synectics*, New York: Harper & Row, 1961.
- Grabowski, H. and Benz, T. Functional Modelling in Intelligent CAD Systems, *Pre-prints of the Second IFIP WG 5.2 Workshop on Intelligent CAD, Cambridge, UK, September 1988*, pp. 179–96.
- Grabowski, H. and Benz, T. Implementing the Design Methodology, *Pre-prints of the Third IFIP WG 5.2 Workshop on Intelligent CAD, Osaka, Japan, 26–29 September 1989*, pp. 17–37.
- Harrisberger, L. A Number Synthesis Survey of Three Dimensional Mechanisms, *Journal of Engineering for Industry*, pp. 213–20, May 1965.
- Herkimer, H. *Engineers' Illustrated Thesaurus*, New York: Chemical Publishing, 1952.
- Hix, C. F. Jr and Alley, R. P. *Physical Laws and Effects*, New York: John Wiley, and London: Chapman & Hall, 1958.
- Hoeltzel, D. A. and Chieng, W-H. Knowledge-Based Approaches for the Creative Synthesis of Mechanisms, *Computer-Aided Design*, vol. 22, no. 1, pp. 57–67, January–February 1990.
- Hoeltzel, D. A., Chieng, W-H. and Zissimides, J. Knowledge Representation and Planning Control in an Expert System for the Creative Design of Mechanisms, *AI EDAM* vol. 1, no. 2, pp. 119–37, 1987.
- Hoover, S. P. and Rinderle, J. R. A Synthesis Strategy for Mecyanical Devices, *Research in Engineering Design*, vol. 1, pp. 87–103, 1989.
- Hubka, V. *Principles of Engineering Design*, Butterworth, London, 1982.
- Johnson, A. L. Functional Modelling: A New Development in Computer-Aided Design, *Preprints of the Second IFIP WG 5.2 Workshop on Intelligent CAD, Cambridge, UK, September 1988*, North-Holland.
- Kota, S. and Chiou, S-J. Conceptual Design of Mechanisms Based on Computational Synthesis and Simulation, *Research in Engineering Design*, vol. 4, pp. 75–87, 1992.
- Kusiak, A., Szczerbicki, E. and Park, K. A Novel Approach to Decomposition of Design Specifications and Search for Solutions, *International Journal for Production Research*, vol. 29, no. 7, pp. 1391–1406, 1991.
- Lai, K. and Wilson, W. R. D. FDL – A Language for Function Description and Rationalization in Mechanical Design, *ASME Journal of Mechanisms, Transmissions and Automation in Design*, vol. 111, pp. 117–23, March 1989.
- Matousek, R. *Engineering Design: A Systematic Approach*, London: Blackie, 1963.
- Ornstein, R. *Psychology of Consciousness*, Oxford: W. H. Freeman, 1972.
- Pahl, G. and Beitz, W. *Engineering Design*, London: Design Council, 1984.
- Paynter, H. M. *Analysis and Design of Engineering Systems*, Cambridge, Mass.: MIT Press, 1961.
- Pitts, G. *Techniques in Engineering Design*, London: Butterworths, 1973.
- Prabhu, D. R. and Taylor, D. L. Some Issues in the Generation of the Topology of Systems with Constant Power–Flow Input–Output Requirements, in S. S. Rao (ed.), *Proceedings of the ASME Design Automation Conference, Kissimmee, Florida, September, 1988*, pp. 41–8.
- Prabhu, D. R. and Taylor, D. L. Synthesis of Systems from Specifications Containing Orientations and Positions Associated with Generalized Flow Variables, Submitted to the ASME Design Automation Conference, 1989.
- Reuleaux, F. *The Kinematics of Machinery: Outline of a Theory of Machines*, New York: Dover, 1963 (1st edn 1876).
- Rosenberg, R. C. and Karnopp, D. C. *Introduction to Physical System Dynamics*, McGraw-Hill, New York/London, 1983.
- Schmekel, H. Functional Models and Design Solutions, *Annals of the CIRP*, vol. 38, no. 1, pp. 129–32, 1989.
- Ulrich, K. T. and Seering, W. P. Synthesis of Schematic Descriptions in Mechanical Design, *Research in Engineering Design*, vol. 1, no. 1, pp. 3–18, 1989.
- VDI 2221. *Systematic Approach to the Design of Technical Systems and Products*, Dusseldorf: VDI-Verlag, and Berlin: Beuth Verlag, 1985.
- Woo, L. S. Type Synthesis of Plane Linkages, *ASME Journal of Engineering for Industry*, pp. 159–65, February 1967.
- Yoshikawa, H. Design Theory for CAD/CAM Integration, *Annals of the CIRP*, vol. 34, no. 1, 173–8, 1985.
- Yoshikawa, H. General Design Theory and a CAD System, in T. Sata and E. Warman (eds), *IFIP Man–Machine Communication in CAD/CAM*, pp. 35–88, 1981.