

SUPPORTING FUNCTIONAL REASONING IN DESIGN

Amaresh Chakrabarti
Engineering Design Centre, Department of Engineering
University of Cambridge, Trumpington Street
Cambridge CB2 1PZ, UK

ABSTRACT

Reasoning about functions is essential in conceptual design. It is inherently hard, and thus needs to be supported, which should include identification and elaboration of function to describe a design problem, generation of solutions, and their evaluations. This paper describes an approach for supporting (i) representation of design problems in terms of their functions, and (ii) representation and synthesis of solutions to instantaneous parts of these functions.

1 INTRODUCTION

Investigation into how the knowledge of functionality (of design problems and possible solutions) guides, constrains and shapes the design activity, is an important strand of present design research. One goal of this research is to understand the conceptual design activity (i.e., how functional requirements of a design problem are transformed into schematic descriptions of design solution concepts), and to develop (computational) methods to support this process. In conceptual design, a design problem is clarified in terms of its functions, various solutions are generated to satisfy these functions, and a host of evaluations are performed to check and ensure that the proposed solutions indeed can provide the required functionality. Conceptual design is characterised by reasoning in terms of functions. The idea of functional reasoning [Freeman & Newell, 1971; Yoshikawa, 1981, 1985; Grabowski & Benz, 1988, 1989; Schmekel, 1989; Chakrabarti & Bligh, 1991] in conceptual design is to allow reasoning at the functional level in order to support (i) identification and elaboration of functions required to describe a design problem, (ii) generation of solutions to satisfy these functions, and to support (iii) evaluation of these solutions for their suitability to satisfy specified functions.

This paper describes an approach to support the first two of the above activities. Simulation and evaluation of solutions is not the present focus of this research (see Joskowicz & Sacks [1991], Faltings [1990] for some of the principal approaches). A formalism for describing a design problem is proposed, and a number of processes for function identification and elaboration are suggested. A scheme is then proposed which can be used to describe and detail these temporal functions. Finally, an implemented representation, which can be used to describe an instantaneous part of a temporal function, and a synthesis procedure that can be used to generate a set of conceptual solutions to such function, are described.

2 REPRESENTATION OF A DESIGN PROBLEM AND IDENTIFICATION AND ELABORATION OF ITS FUNCTIONS

2.1 Representing a Design Problem

Following concepts will be used here to describe a design problem [for further detail, see Chakrabarti, 1994]:

Object: (mental models of) a physical thing; these could be described in terms of a set of attributes, or lower-level objects and their relationships. In fig.1a, there are two objects: a support, and a abstract object *ground* (which is immovable).

Parameter: attributes such as colour, density, shape, velocity, etc., used to describe (the state of) things. In fig.1, for instance, the support could be described as a horizontal plane of finite dimensions.

Relationship: describes relationships between objects, such as the spatial relationships between various members of a design. In fig.1a, there is a fixed-connection between the two objects, which implies that they can only move together.

Situation: a set of objects, relationships, and (potential or active) processes with unique values of their parameters. For the case in fig.1a and 1b, this includes the objects, their relationships, and equilibrium as the active process as a result of a balance between the weight of the support and the reaction to this force from the ground (see situation-1 in fig.1c).

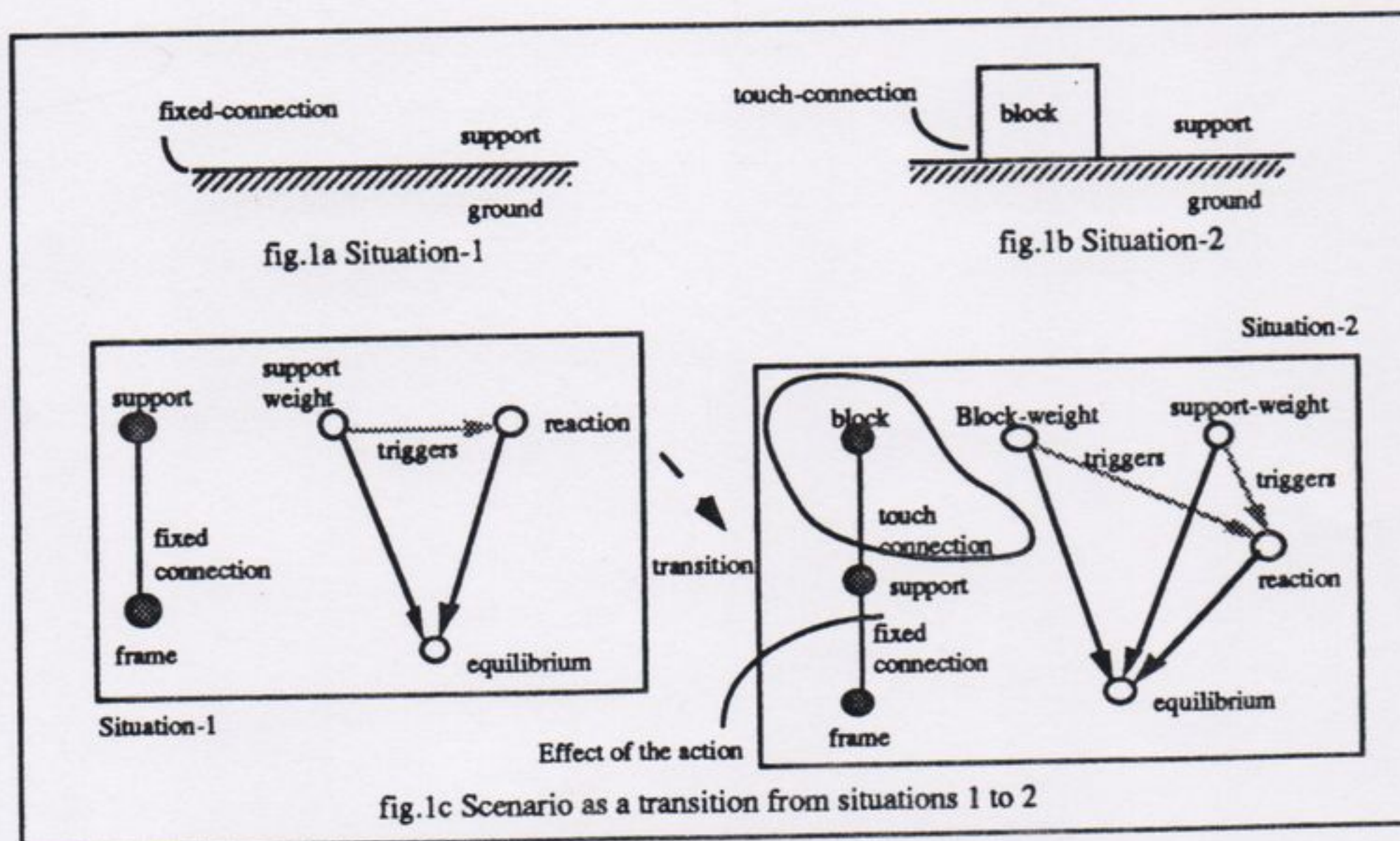


Fig. 1 An example for actions, situations and scenarios

Processes: those which change a situation: could be described in terms of the action, input situation(s), and output situation(s); for instance, if an unbalanced force acts on an object, the

object's state of velocity changes.

Action: changes introduced into a situation; this could be in terms of changes in the objects (including the introduction of new objects), or changing relationships between objects. If a block is introduced into the above situation such that a touch-connection is established between the support and the block (see fig.1b), this action will activate the weight (gravity process) to act on the support, leading to the activation of reaction process as a consequence; these two forces then activate the process of equilibrium, thereby allowing the situation to be as in fig.1b with the three objects, their two relationships with the whole system as static.

Scenario: An ordered set of situations. Fig.1c describes the scenario for transition from situation-1 (in fig.1a) to situation-2 (in fig.1b).

Essentially, a designer has a mental model of what the present situation is, and how this would change (including the scenario where the situation does not change), given a set of assumed actions. One could call this the *otherwise-expected* scenario (the scenario without the design). The designer also has a picture of a desired situation or situations, which is not the final situation in the otherwise-expected scenario. Let us call the scenario, initially defined as the transition between the initial situation (same as that of the otherwise-expected scenario) and the desired situation, as the *desired* scenario. A design problem is defined here as the transformation between these two scenarios, see fig.2. The outcome of the design process is a body of information, which can be used to describe the actions, objects and relationships, involved in the desired scenario, in sufficient degree of detail so as to warrant (i.e., justify and enable) the later processes such as manufacturing to take place.

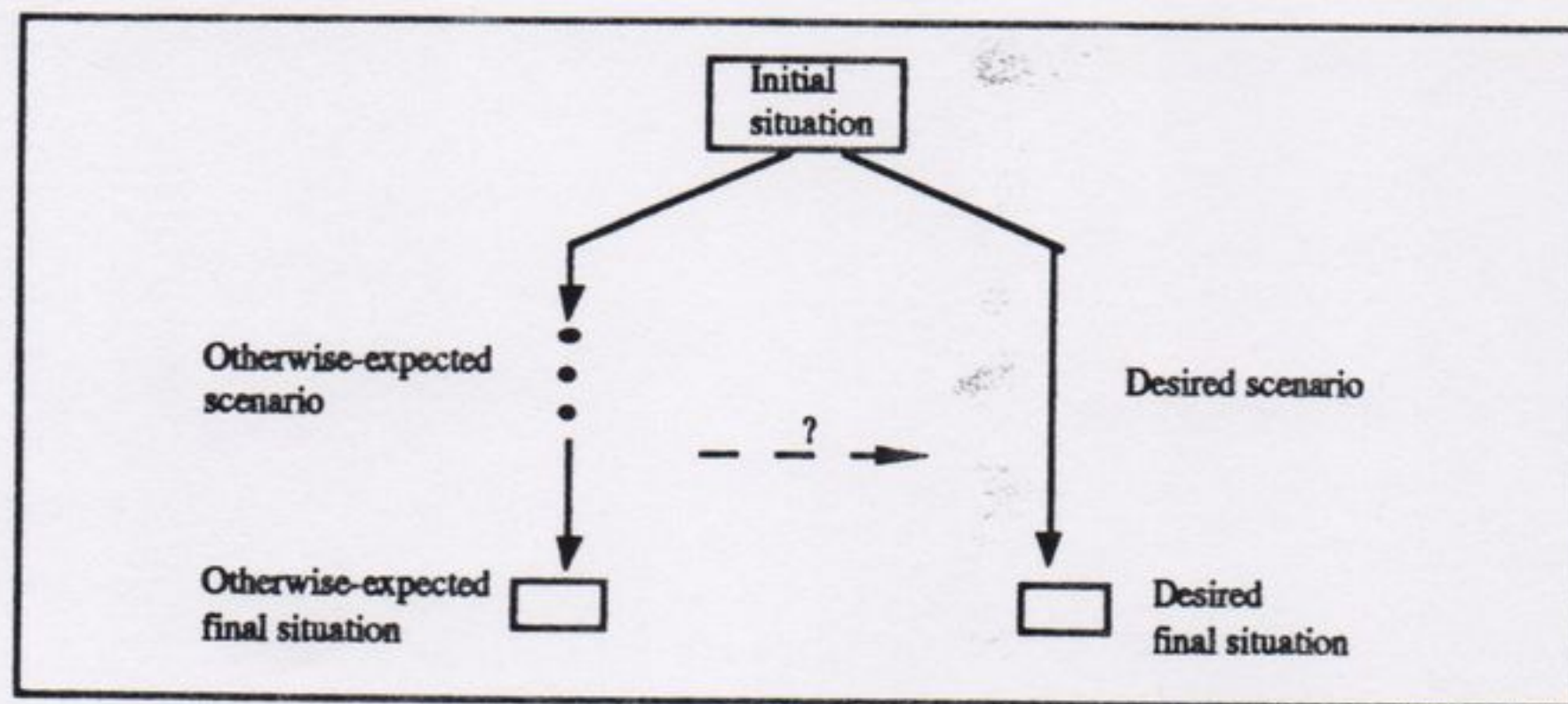


Fig. 2 A proposed definition of a design problem.

2.2 Proposed Processes Of Function Identification And Elaboration

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. The activities by which design functions are identified and elaborated involves comparing the desired scenario with the present

scenario. The process of this comparison, however, is difficult. Firstly, the desired scenario is at most only incompletely known in terms of its objects and their relationships (particularly the design and how it relates to the rest of the situation, which we might call the context). Further, some actions are intended to be used in the context, and other actions are assumptions*. Similarly, the otherwise-expected scenario also is at most incompletely defined, although less so than the desired scenario. A set of possible ways of identifying and elaborating functions are given below [for more detailed discussion, see Chakrabarti, 1994].

Action identification problem: Involves finding possible modes of action within the scenarios other than the given ones. Some of these involve making assumptions requiring validation at later stages, while others can be conceived only in the context of specific solutions.

Transformation problem: involves comparing the desired with the other-wise expected scenario, and identifying the reasons which might have led to the undesired, and whose change would bring about the desired scenario.

Implementation problem: involves finding and detailing the potential solutions.

Side effects and additional problems: involves identifying new functions which is required by a solution to a previous function, as a side effect; this could also involve identifying additional functions, initially not part of the original problem, which the solution satisfies as a side effect. The other two modes of function identification considered here are (i) analogous problems, and (ii) finding mistakes in the previously carried out problem identification activities. The first one involves considering (part of) the functions associated with a previously known (similar) design for the present design; the other ones is self-evident.

3 REPRESENTING TEMPORAL FUNCTIONS

When a situation undergoes a transition, a number of changes in its parameter values as well as connectivity relationships take place. A temporal function can be defined in terms of the actions taken and the differences between the initial and resulting situations. For instance, the functionality of a door latch can be described at one level in terms of a state diagram connecting three distinct situations: *door closed*, *door open*, and *door potentially open*, with required actions between them for transitions between them. Lower level functions can be described by adding extra assumptions and then looking at specific parts of this problem. For instance, once *door closed* is constrained into providing a zero degree of freedom between the fixed frame and the otherwise movable door (by assuming that when these two have zero degree of freedom, no considerable gap exists between the door and the frame), one could assume possible actions which should lead to change of situation, such as rotation of a handle leading to a motion

* Each such assumption immediately restricts the kinds of solutions which could now be considered. For instance, the action of turning the bottle around restricts the possibility of considering breaking the bottle after solidifying the wine, whereafter taking the solid-wine out before heating it to the initial state. Although this solution is not a viable one, these assumptions do not necessarily restrict only the non-viable ones.

leading from a zero degree of freedom to a one degree of freedom between door and frame (see figure 3). At this stage, one could say that a part of the functionality of a door latch is to move a point in space in a certain direction, given an input rotation (action) in a certain way. Finally each such temporal part of function can be described in terms of a series of instantaneous functions. An instantaneous synthesis procedure, developed in Cambridge, can then be used to synthesise concepts to satisfy such instantaneous functions. The synthesis procedure is briefly described below [for further details, see Chakrabarti & Bligh, 1991].

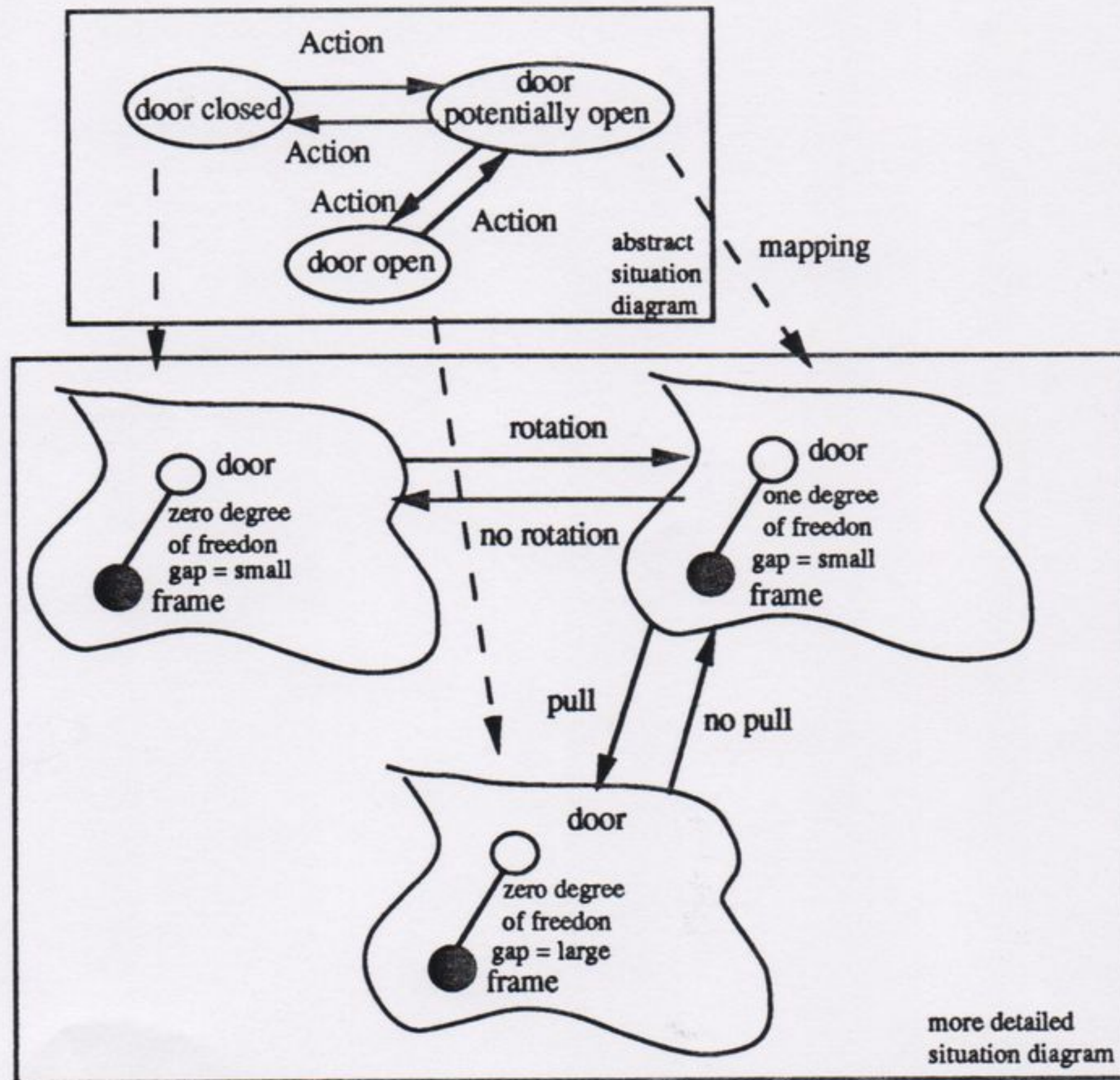


Fig. 3 Use of situation diagrams to express temporal functions.

4 INSTANTANEOUS SYNTHESIS

4.1 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 (figure 4). The constructs for representing a MIMO design problem involving a transformation between m inputs and n outputs are:

Problem Inputs, Outputs
 Inputs $x_1, x_2, x_3 \dots x_m$
 Outputs $x_{m+1}, x_{m+2}, x_{m+3} \dots x_n$
 x_i (1..m+n) Kind: (*force / torque / linear motion / angular motion*)
 Orientation: ($i / j / k$) [$i/j/k$ are unit vectors in a Cartesian co-ordinate system]
 Sense: (+ / -)
 Magnitude: (some number)
 Position: ($x|i + y|j + z|k$)

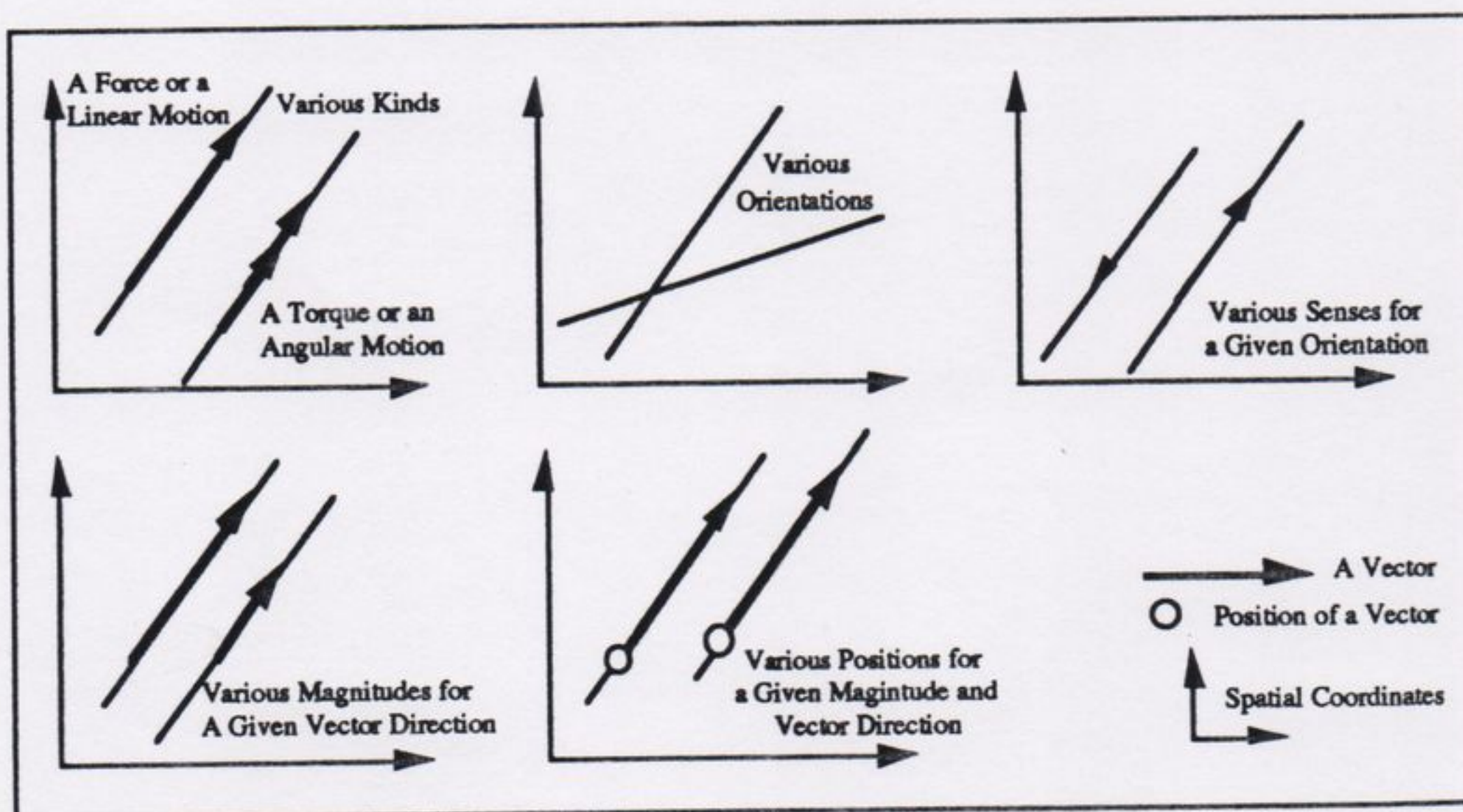


Fig. 4 Various characteristics associated with an I/O vector in mechanical transmission design.

4.2 Representation of Solutions

As the input/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 I/O points of the vectors associated with a transformer specify their positions in space, and the spatial separation between these I/O points becomes the position transformation by the transformer. The I/O 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. 5) 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 characteristics are variously coupled, depending on the characteristics of the specific transformer involved.

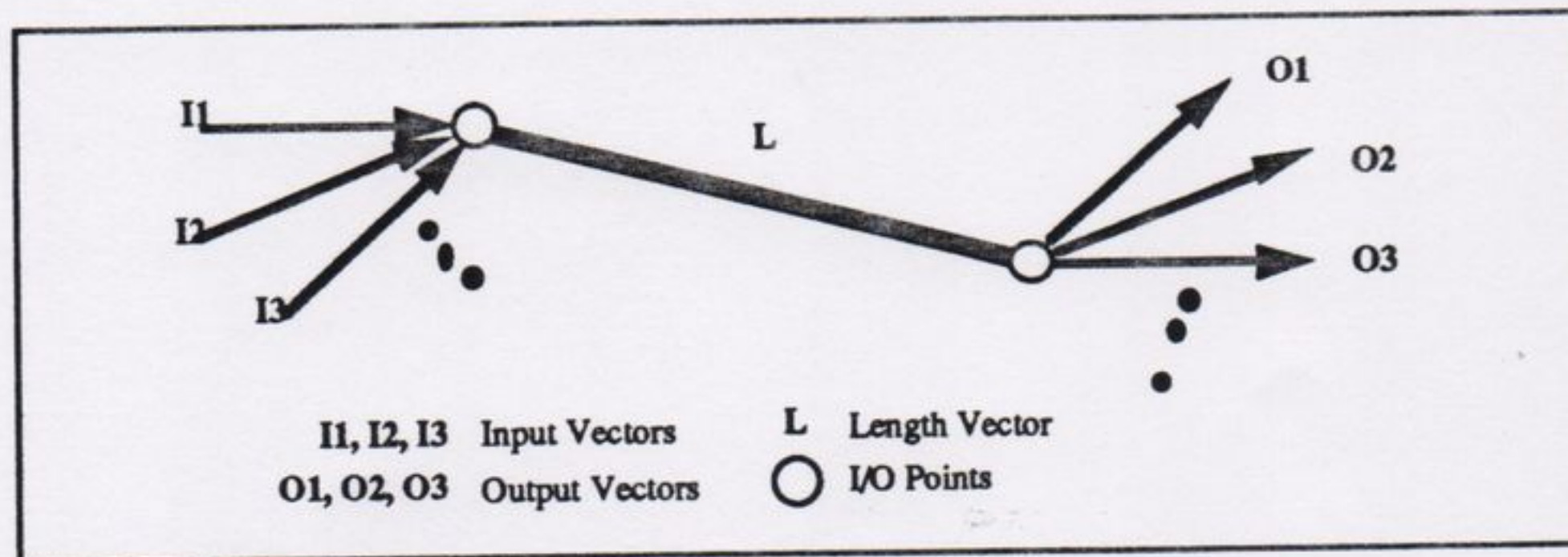


Fig. 5 Representation of a general transformer

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 relations, namely, 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. 6. So, a shaft would be a structure of type PI (Fig. 6) having torque (and/or angular motion) as the input and output kinds which are parallel and intersecting. A solution is defined as a network of structures defined using the above representation, and its I/O properties as well as spatial layouts can be computed from the characteristics of the individual structures of which it is composed.

4.3 Synthesis Procedure

A synthesis procedure has been developed which can take a database of structures, such as shaft, lever, wedge etc., represented as described in section 4.2, and a specification of a problem in terms of its instantaneous function using the representation in section 4.1, and use search and constraint propagation techniques to generate an exhaustive set of conceptual solution layouts to solve the problem. For the door latch problem, one such function is to provide an output translation from an input force, and a solution generated is shown in Fig. 7.

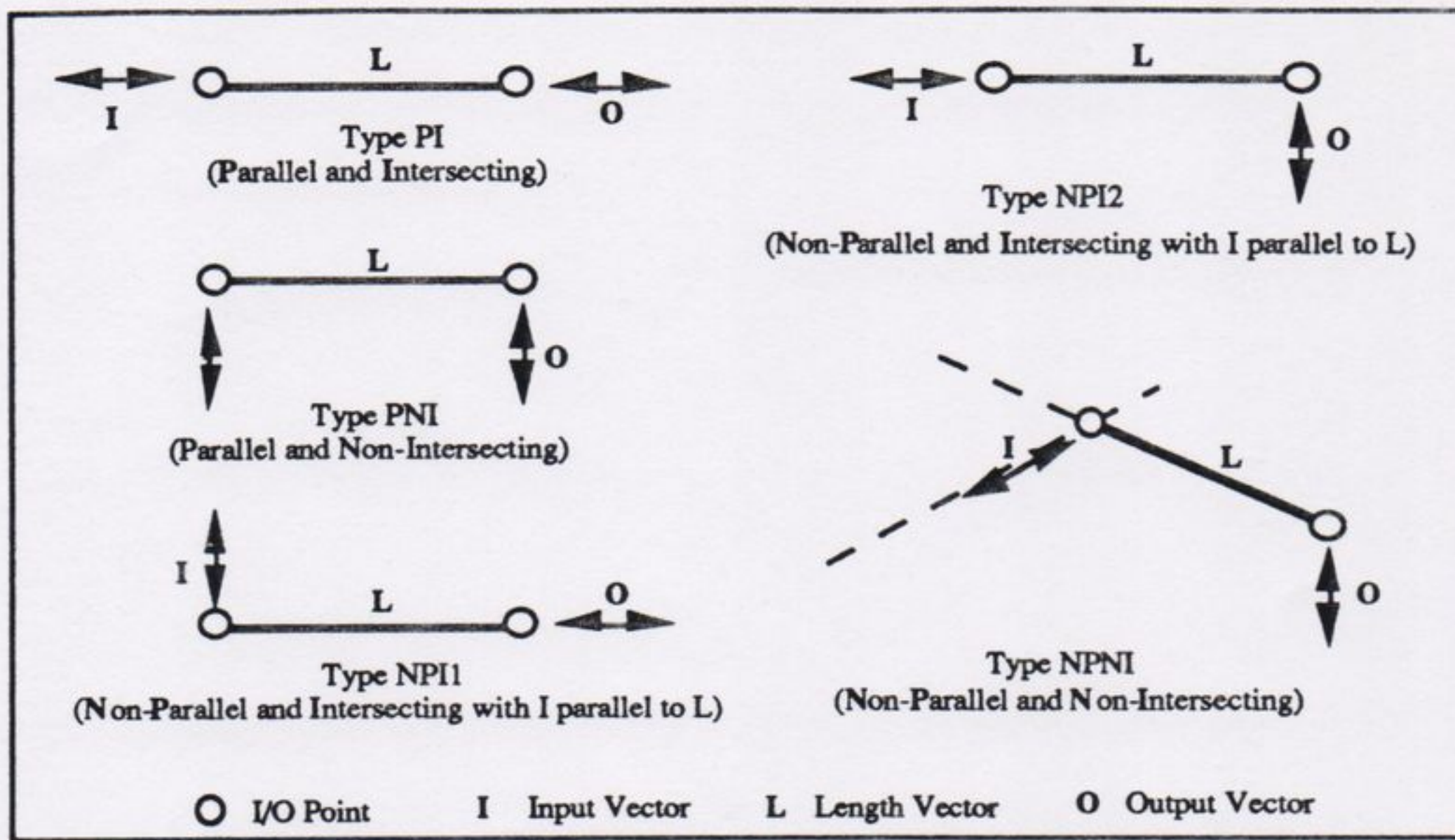


Fig. 6 Various possible spatial relations between the I/O vectors of a transformer.

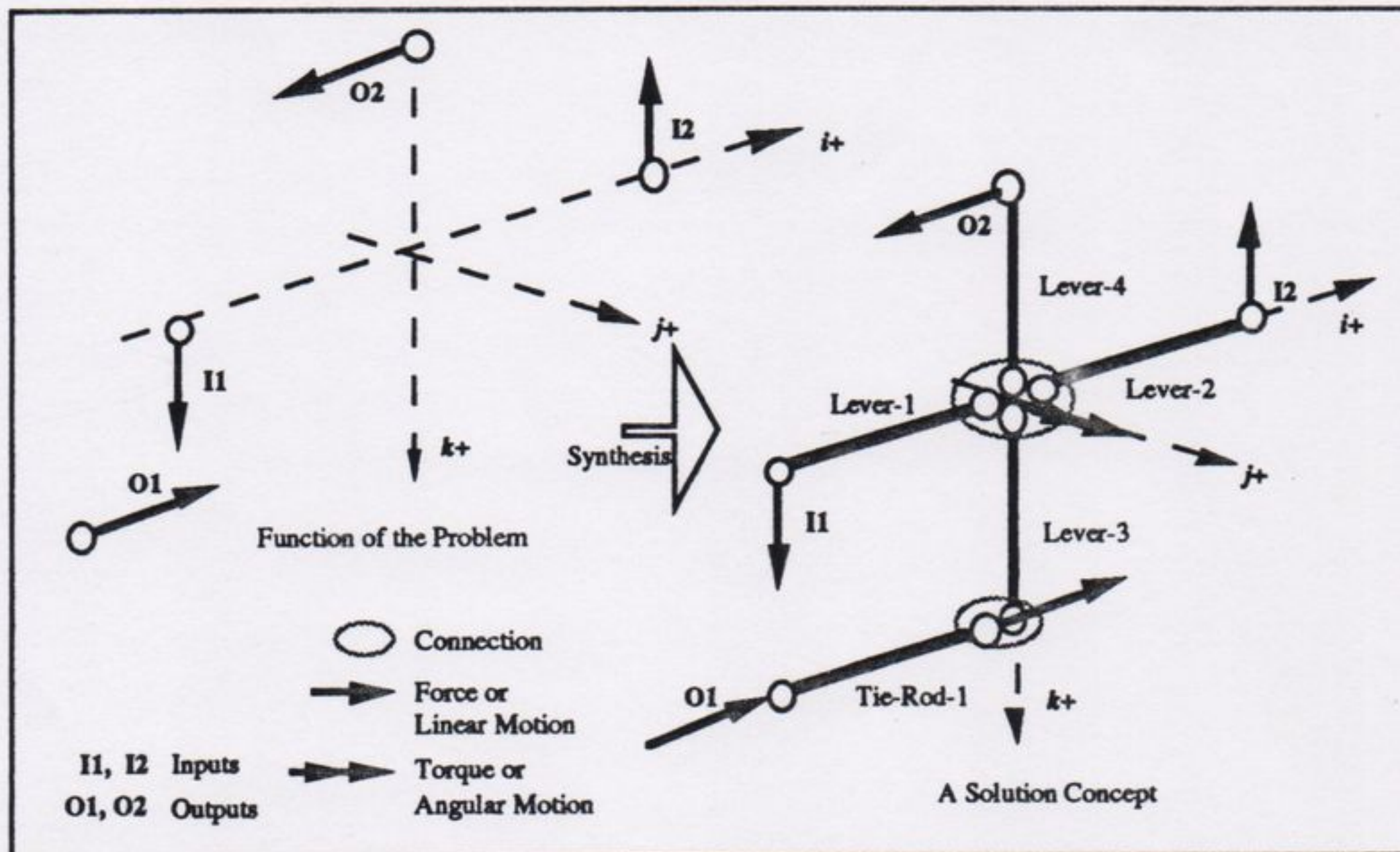


Fig. 7 Representations of an instantaneous function of the door-latch problem and one of its solutions

5 SUMMARY

Three major activities need to be supported in functional reasoning: (i) function identification and elaboration in a design problem, general of solutions and their evaluations. In this paper,

an approach to describing design problems, and identifying and elaborating their functionality is proposed; and a synthesis procedure and its representations for solving instantaneous parts of a given problem function is described.

REFERENCES

- Chakrabarti, A. "Requirements Identification: A Central Issue in Design Research", To be published in *Proc. of East-West Conference on IT in Design*, Moscow, 3-5 September, 1994.
- Chakrabarti, A. and Bligh, T.P. "Towards a Decision-Support Framework for Mechanical Conceptual Design", *Proc. ICED*, vol. 1, 384-389, Zurich, 27-29 Aug. 1991.
- Chakrabarti, A. *Designing by Functions*, Chapter 2, Ph.D. Thesis, Department of Engineering, Cambridge University, UK, 1991.
- Chakrabarti, A., and Bligh, T.P. "A Knowledge-Based System for Synthesis of Single Input Single Output Systems in Mechanical Conceptual Design", *Proc. 4th Int. Joint Euro-US Conf. on AI and Expert Systems Applications*, Houston, 401-406, 1992.
- Faltings, B. "Qualitative Kinematics in Mechanisms", *Artificial Intelligence*, vol. 44, pp. 89-119, 1990.
- Freeman, P., and Newell, A. "A Model for Functional Reasoning in Design," *Proc. Second International Joint Conference on Artificial Intelligence*, London, 621-633, 1971.
- 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, 179-196, 1988.
- Joskowicz, L., and Sacks, E.P. "Computational Kinematics", *Artificial Intelligence*, vol. 51, pp. 381-416, 1991.
- Schmekel, H. "Functional Models and Design Solutions", *Annals of the CIRP*, 38(1), 129-132, 1989.
- Yoshikawa, H. "General Design Theory and a CAD System", *IFIP Man-machine Communication in CAD/CAM in T. Sata and E. Warman (eds.)*, 35-88, 1981.