INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

COMPUTATIONAL REPRESENTATIONS FOR MULTI STATE DESIGN TASKS AND ENUMERATION OF MECHANICAL DEVICE BEHAVIOUR

Somasekhara Rao Todeti[1] and Amaresh Chakrabarti [1]
(1) Indian Institute of Science, Bangalore, India

ABSTRACT
The role of a computer evolved from supporting modeling and analyzing concepts (ideas) to supporting generation of concepts. Research into methods for supporting conceptual design using automated synthesis attracted much attention in the past decades. To understand how designers synthesize solution concepts for multi-state mechanical devices, ten experimental studies have been conducted in our work. Observations from these studies would be used as the basis for developing knowledge required for the multi-state design synthesis process. In this paper, we propose a computational representation for expressing the multi-state design task and for enumerating multi-state behaviors of kinematic pairs and mechanisms. This representation would be used to help formulate computational methods for the synthesis process to develop a system for supporting design synthesis of multiple state mechanical devices by generating a comprehensive variety of solution alternatives.

Keywords: Automated synthesis, multiple state, conceptual design, mechanical device, function representation, enumeration of behaviors

1 INTRODUCTION
Mechanical design could be seen as a process of transforming a perceived need into a description of a physical structure that uses mechanical engineering principles to satisfy the need. In conceptual design, functional requirements are transformed into the concept of a solution. Conceptual design has the most significant influence on the overall product cost [1], [2]. Conceptual design is a difficult task [3], [4], which relies on the designer's intuition and experience to guide the process. A major issue within this task has been that often not many potential solutions are explored by the designer during the design process [5], [6], [7]. The major reasons for this have been the tendency to delimit a design problem area too narrowly and thus not being able to diversify the possible set of design solutions, possible bias towards a limited set of ideas during the design process, and time constraints [8]. Evidence from earlier research suggests that a thorough exploration of solution space is more likely to lead to designs of higher quality [9]. Therefore, a support system, automated or interactive, that could help generate a considerable variety of feasible design alternatives than currently possible at the conceptual design phase is important to the development of intelligent CAD tools that can play a more active role in the mechanical design process, especially in its earlier phases.
Li [8] defines the operating state of a mechanical device by a set of relations between its input and output motions that remain unchanged within the state. A multiple state device has more than one operating state. Other researchers [10-13] define state in various other ways. The definition of state used by Li [8] is adopted in this research work.

2 RESEARCH OBJECTIVES
The overall aim of this research is to develop a generic computational system to support designers during the conceptual phase of mechanical design to synthesize a wider variety of design alternatives than currently possible for multiple state mechanical devices. To understand the process through which engineering designers synthesize multiple state devices, ten experimental studies are conducted. This would throw light on how (and how well) multiple state synthesis tasks are currently handled, what one could learn from these, and how one could help improve these tasks with computational tools. The research work presented in this paper is a continuation of our previous work [14], where a multiple state design task is specified, by the researcher, using an appropriate set of functions necessary for the functioning of an intended device. Ten designers, including the researcher, took the

same design problem described in terms of these functions, and worked individually to generate as many design alternatives as possible. In this paper, a computational representation for functions describing a multi-state design task is proposed. Behaviours of the pairs and mechanisms generated by the designers during the synthesis processes studied earlier are expressed using the proposed representation as a demonstration of its capability. This representation is intended to serve as a basis for developing synthesis methods for use in a computational tool planned for generating concept solutions for multi-state mechanical design tasks.

3 LITERATURE STUDY

The work of Li [8] is the only prior research that has directly addressed multi-state synthesis tasks. He used the configuration space approach to represent and retrieve the behaviour of a kinematic pair and developed ADCS system for automatically generating solutions of mechanical devices satisfying a given multi- state design task. However, ADCS is limited to generation of a single design solution for each design specification, rather than a comprehensive list of all possible design alternatives for a given task – a critical drawback if the goal is to support generation of a wider range of concepts. Single state design synthesis [15-23] is limited to synthesizing either single kinematic pair or multiple kinematic pairs for single input and single output, using simulation–based, configuration space, or grammatical approaches.

The representation for specifying functions proposed by Li [8] based on effort-motion state of a component does not consider the type of effort, the type of motion, and axis of motion. i.e. in this representation torque and force, rotation- translation, rotation-rotation and translation-translation and motions along x and motion along y are not distinguishable. However, our experimental studies show that designers generate proposals that distinguish among types of effort, types of motion, motion type transformations and axes of motions. Further, since motion types are currently not distinguished, when pairs are automatically synthesised together into mechanisms, there is a possibility of one component, with one type of motion, from one pair getting connected to the connecting components with of other pairs with a different motion. This can lead to motion mismatch. Another limitation of ADCS is that it is limited only to two motion axes as it uses 2D configuration space. In summary, there is a need for a richer representation for multi-state mechanical device tasks and behaviours of kinematic pairs.

4 REPRESENTATION OF STATES OF MULTIPLE STATE MECHANICAL DEVICES

A door attached with a latch, is a four-state mechanical device. This is analyzed for its states, and its functioning within each of the states.

4.1 Multi-state functioning of a door attached with a latch

The functions of a door are to allow or prevent the transfer of energy or material. When a door is in the locked state, it completely prevents movement of these to and from the room. When it is in the

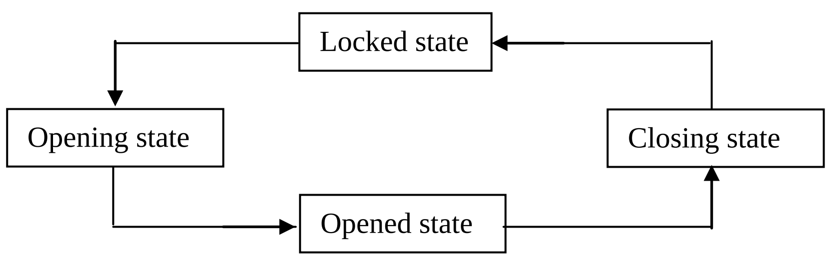
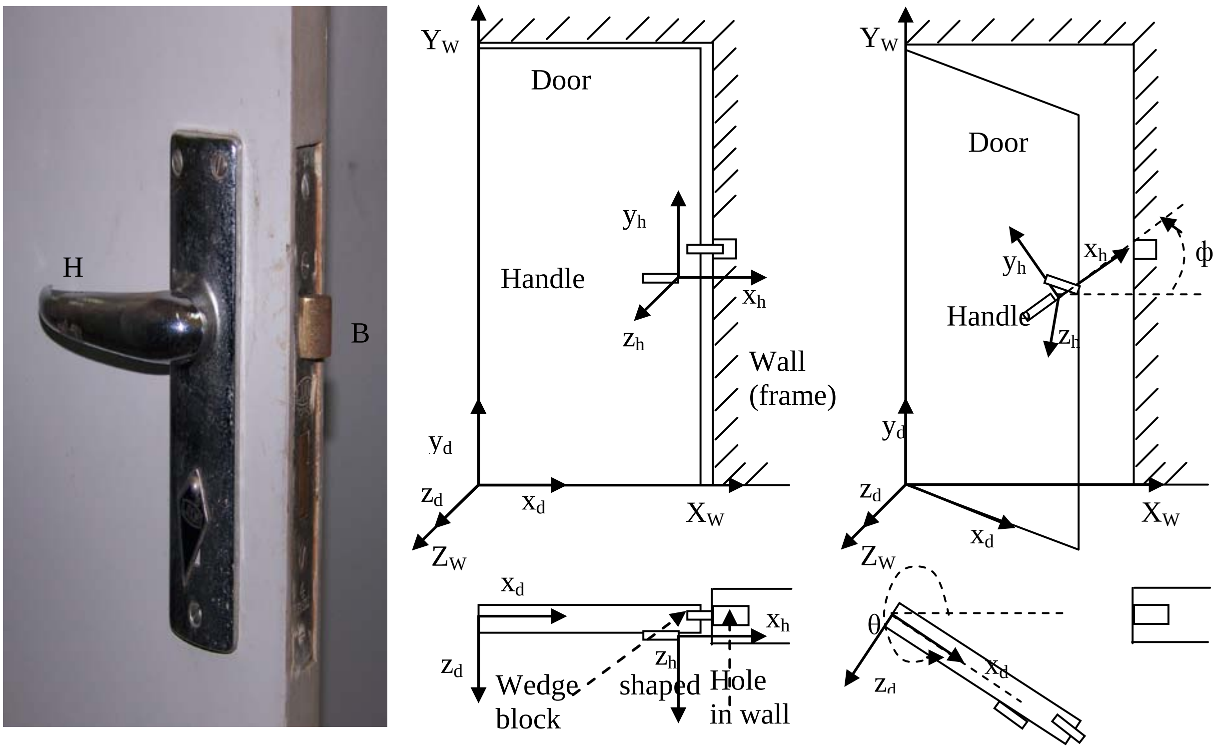


Figure 1.States and state transitions of door attached with latch

opened state, it allows movement of these to and from the room. In between these two states, there are opening and closing states as shown in Figure 1. One way in which a door achieves these functions is when a latch is attached to it. The functions in each state depend on the type of latch attached to a door. For the type of latch shown in Figure 2(a), the functions performed in each state are:

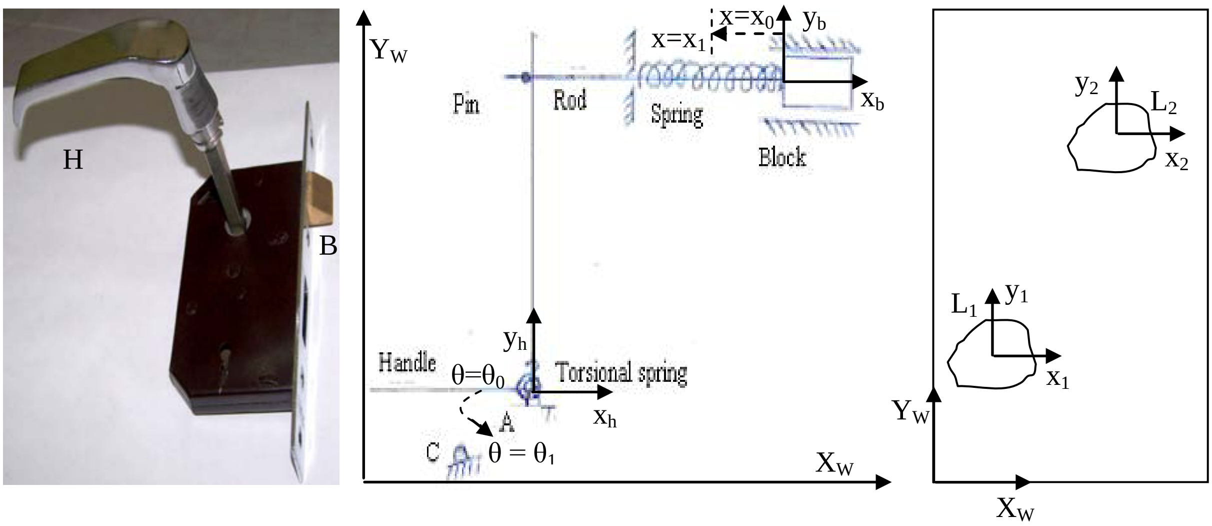
1. Locked state: Function1: the door is pushed but it does not move.
   Function2: the door is pulled, but it does not move.

2. Opening state: Function3: the handle is rotated by applying an effort.
   Function4: keeping the effort in the function3 on the handle, another effort is applied to the door, and the door opens.
   Function5: as the effort on the handle is released, the handle rotates back.
3. Opened state: Function 6: as the door is pulled, it opens further.
   Function 7: as the door is pushed, it closes further.
4. Closing state: Function8: by applying further effort on the door, it comes to the locked state, where further push or pull does not move the door.



(a) (b) (c)

Figure 2. The door attached with latch



(a) (b) (c)

Figure 3 The door latch, its structure and its components

The door latch is disassembled to further study its components and interfaces, and how this latch attains the functioning of the door when attached to it. The latch and components (Component H is the handle and Component B is the wedge shaped block) are shown in Figure 3(a).
From an understanding of these components and interfaces, the overall structure of the latch, and its functioning, the latch is modelled as shown in Figure 3(b). The latch has an L-shaped handle hinged at A, a torsion spring connected to the handle at A, a block, a rod attached to the block and a spring arrangement, where the spring is confined between the block and a support with a hole through which the rod can translate, a small pin attached to the rod protruding perpendicular to the plane of the paper, and a stop at C. This is a plane mechanism. The world coordinate system (X_w, Y_w), local coordinate systems for the handle (x_h, y_h) and the block (x_b, y_b) are shown in Figure 3(b). The motion transformations between the handle (H) and the block (B) due to various efforts are described below as five functions.

- F1: If effort is applied on the handle, the handle rotates (from $\theta = \theta_0$ to $\theta = \theta_1$), and simultaneously the block translates inside(from $x=x_0$ to $x=x_1$).
- F2: Even if effort is kept applied in the same direction when the handle is $\theta = \theta_1$, the handle does not rotate any further due to the obstruction C, and the block remains at $x = x_1$.
- F3: If the effort is released from the handle, the handle rotates back to $\theta = \theta_0$ from $\theta = \theta_1$ and simultaneously the block also translates back to $x = x_0$ from $x = x_1$.
- F4: Now if effort is applied on the block, the block translates from $x = x_0$ to $x = x_1$, but there is no motion in the handle.
- F5: If the effort on the block is released, the block goes back to $x = x_0$ from $x = x_1$ but the handle does not move.

4.2 Framing a door latch design task

By taking two components L_1 and L_2, to act as the handle and the block respectively, a five-function design task can be devised for the above five functions as follows. The world coordinate system (X_W, Y_W), the local coordinate system(x_1, y_1) for L_1 and the local coordinate system(x_2,y_2) for L_2 are as shown in Figure 3(c).

- F1: When torque is applied on L_1 in the counter-clockwise direction along z-axis, L_1 rotates in counter-clockwise direction along z –axis from $\theta = \theta_0$ to $\theta = \theta_1$, and simultaneously L_2 translates along x-axis in negative direction from $x = x_0$ to $x = x_1$.
- F2: Even if torque is kept applied in the same direction on L_1, when L_1 is at $\theta = \theta_1$, L_1 does not rotate beyond $\theta = \theta_1$, and L_2 remains at $x = x_1$.
- F3: If the torque is released on L_1, when L_1 is at $\theta = \theta_1$ then L_1 rotates back in the clockwise direction from $\theta = \theta_1$ to $\theta = \theta_0$ and L_2 simultaneously translates from $x = x_1$ to $x = x_0$ along x-axis.
- F4: Now if force is applied on L_2 along x-axis in the negative direction, L_2 translates along x-axis in the negative direction from $x = x_0$ to $x = x_1$, but L_1 remains at $\theta = \theta_0$.
- F5: If the force on L_2 is released, L_2 translates back to $x = x_0$ from $x = x_1$, but L_1 remains at $\theta = \theta_0$.

These five functions are given to the designers to generate as many solutions as possible.

5 COMPUTATIONAL REPRESENTATION OF THE DOOR LATCH DESIGN TASK

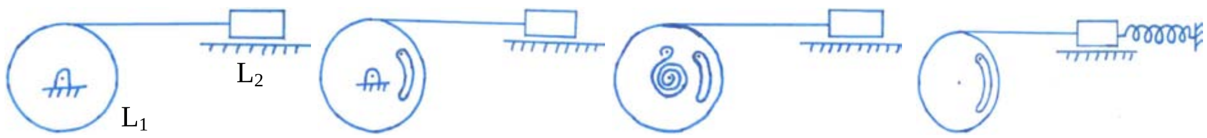
If the design synthesis has to be automated or supported in an intelligent manner,, there should be a computational representation for design tasks, which should be able to represent the functions involved in theses tasks. In the design task analyzed earlier, there are two components L_1 and L_2 that act as input or output components during the various functions of the task. Each component acting as input or output component is currently allowed at-most one degree of freedom, which is also called fixed axis motion [8]. Motion for the component can be associated with an axis around which it occurs. By taking L_1 as the first component and L_2 as the second component in this function representation, the above five functions can be formally represented as follows:

- F1: $<(L_1,$ torque, z, +, rotation, z, +, $\theta_0, \theta_1),(L_2,0,0,0,$ translation, x,-, $x_0, x_1)>$
- F2: $<(L_1,$ torque, z,+,0,0,0,θ1, $\theta_1),(L_2,0,0,0,0,0,0, x_1, x_1)>$

- F3: <(L_1,0,0,0, rotation, z,-, θ_1, θ_0),(L_2,0,0,0,translation,x,+, x_1, x_0)>
- F4: <(L_1,0,0,0,0,0,0, θ_0, θ_0),(L_2,force, x, -, translation, x,-, x_0, x_1)>
- F5: <(L_1,0,0,0,0,0,0, θ_0, θ_0),(L_2,0,0,0, translation, x,+, x_1, x_0)>

The nine parameters within the first parenthesis in each function belong to component L_1, the nine parameters within the second parenthesis belong to L_2. In this representation, each component has a nine parameter tupple (p1,p2,p3,p4,p5,p6,p7,p8,p9). The first parameter (p1) denotes the component identifier. The next three parameters (p2, p3, p4) describe the effort applied on a component. p2 is the type of effort, which is either torque, force or no effort. p3 is the axis along which this effort is applied, which is currently limited to x, y or z. p4 is the direction along the axis on which the effort is applied. p4 can take values of +, 0, or -. The next three parameters (p5, p6 and p7) relate to motion. p5 is the type of motion, which is either rotation, translation or no motion. p6 is the axis of motion. p7 is the direction along that axis. The next two parameters (p8, p9) describe the initial and the final configurations of the component, which may undergo configuration change due to the application of the effort. If no configuration change occurs, the initial and final configurations are taken to be the same. So the quantity spaces [31] for the parameters are: p2= {force, torque, 0}, where 0 is when no external effort is applied. p3={x, y, z, 0}, where 0 is when no external effort is applied. p4= {+, 0, -}, where + is when effort is along the positive direction of the axis of motion, 0 is when no external effort is applied. p5= {rotation, translation, 0}, where 0 is if no motion occurs. P6={x, y, z, 0}, where 0 is when no motion occurs. P7= {+, 0, -}, where + is when motion is along the positive direction of the axis; 0 is if no motion occurs. The values for P8 and p9 are taken as the unknown variables in the functions of the task representation and their values will be known from synthesizing solutions for the design task. .

6 REDUNDANT SPRINGS AND PREPROCESSING OF DESIGN TASK



(a) (b) (c) (d)

Figure 4.

After analyzing the given five functions, F1 is focused on first by one designer. In F1, since rotary motion is transformed into translatory motion, a rope and drum (SP1) is generated as a solution SP1, see Figure 4(a). Drum acts as L_1 and sliding block acts as L_2. As SP1 has satisfied F1, next F2 is selected and SP1 is modified with a pin and slot arrangement producing SP11 shown in Figure 4(b). For F3, a torsional spring between the frame and the drum is connected producing SP111 shown in Figure 4(c). This modification still does not satisfy F3 because when torque is released from L_1, both L_1 and L_2 have to move back. But here only L_1 rotates back but not L_2. So a spring is connected between L_2 (the sliding block) and the frame, and the torsional spring is removed as it is now redundant, producing SP1111 shown in Figure 4(d). SP1111 is evaluated for F4 and F5, and it is found that they are satisfied. SP1111 satisfies all the five functions. If it is examined why this case of redundant spring arises, it is due to F3 and F5.

- F3: <(L_1,0,0,0, rotation, z,-,θ_1, θ_0),(L_2,0,0,0,translation,x,+,x_1,x_0)>
- F5: <(L_1,0,0,0,0,0,0, θ_0, θ_0),(L_2,0,0,0, translation, x,+, x_1,x_0)>

In F3 L_1 has to rotate back and L_2 also has to translate back. In this F3, there is no external effort on L_1 or L_2 but both components move. This means that there exists a spring (or gravitational) force release acting on L_1 or L_2. In F5, only L_2 has to translate backwards. This means that there exists a spring or gravitational force release acting on L_2. So if L_2 is given spring (or gravitational) energy to return, it can satisfy both F3 and F5, avoiding redundant springs. Introduction of a spring can resist external effort applied and oppose motion of components. So it has to be assumed that when external effort is applied on atleast one component in a function, then that external effort is dominating, else spring effort dominates. In this case, let F_p be the spring force, which activates motion in L_2 and thus in L_1 in F3. Again this same F_p activates the motion in L_2 in F5. In F1, F2, and F4, there are external efforts, so they are dominating efforts in those functions. The design task of five functions after preprocessing is as follows:

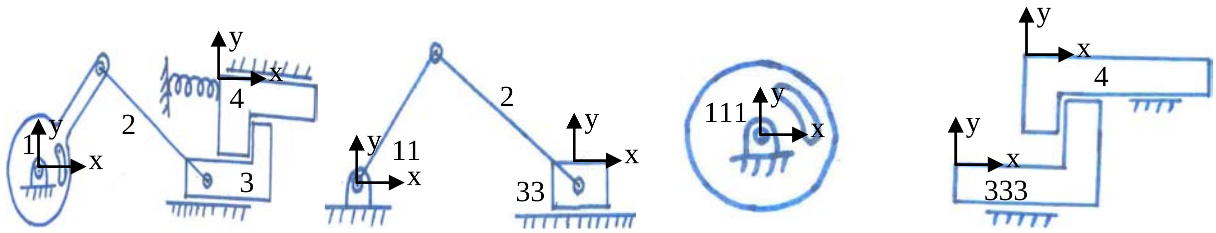
- F11: < (L_1,torque, z, +, rotation, z, +, θ_0, θ_1), (L_2,0,0,0, translation, x,-,x_0,x_1)>
- F22: <(L_1, torque, z,+,0,0,0,θ_1, θ_1),(L_2,0,0,0,0,0,0,x_1, x_1)>
- F33: <(L_1,0,0,0, rotation, z,-,θ_1, θ_0),(L_2,F_p,x,+,translation,x,+,x_1,x_0)>
- F44: <(L_1,0,0,0,0,0,0,θ_0, θ_0),(L_2,force, x,-, translation,x,-,x_0,x_1)>
- F55: <(L_1,0,0,0,0,0,0, θ_0, θ_0),(L_2,F_p, x, +, translation, x, +, x_1,x_0)>

Thus the preprocessing the functions of the given design task helps in avoiding the case of redundant springs and identifying the components to which springs have to be connected and.

7 ENUMERATION OF MECHANICAL DEVICE BEHAVIOURS

In this section, enumeration of behaviors of kinematic pairs and mechanisms is explored using the representation proposed in Section 5. Some of the solutions synthesized by the designers are used as cases in this enumeration.

7.1 Solution1



(a) (b) (c) (d)

Figure 5

The mechanism shown in Figure 5(a) is one of the solutions generated by a designer. For F1, a slider crank mechanism shown in Figure 5(b) is generated. For F2, The crank is modified with a slot and pin. For F3, a torsional spring (which is later removed as it is redundant) is connected between the frame and the crank. As slider crank mechanism can not satisfy F4, the slider crank mechanism is modified by adding a new component(component4) besides component3 and their geometries and interfaces between these two components are modified to satisfy F1, F2,F3 and F4 and a spring is added between the frame and component4 to satisfy F3 and F5. So the solution shown in Figure 5(a) is a combination of a spring, a slider crank mechanism shown in Figure 5(b), the pair shown in Figure 5(c) and the pair shown in Figure 5(d). Component11 and Component111 are attached together to rotate along z-axis and Component33 and Component333 are attached together to translate along x-axis. Let us call the slider crank mechanism shown in Figure 5(b) as structure1 (or str1), its crank (Component 11) as str1.1 and its slider (Component 33) as str1.2, the pair shown in Figure 5(c) as str2 and its rotating disk (Component 111) as str2.1 and the ground as str2.2, and the pair shown in Figure 5(d) as str3 and its components Component333 as str3.1 and Component 4 as str3.2. The behavior of the slider crank mechanism shown in Figure 5(b) is enumerated as follows:

- B01:<(str1.1, torque, z,+, rotation, z,+,0,180),(str1.2, 0,0,0, translation, x,-, 10,8)>
- B02:<(str1.1, torque, z,+, rotation, z,+,180,360),(str1.2, 0,0,0, translation,x,+,8,10)>
- B03:<(str1.1, torque, z,-, rotation, z,-,180,0),(str1.2, 0,0,0, translation, x,+, 8,10)>
- B04:<(str1.1, torque, z,-, rotation, z,-,360,180),(str1.2, 0,0,0, translation,x,-,10,8)>
- B05:<(str1.1, 0,0,0, 0,0,0,0,0), (str1.2, force, x,+, 0,0,0, 10,10)>
- B06:<(str1.1, 0,0,0,0,0,0, 0,0),(str1.2, force, x,-, 0,0,0, 10,10)>
- B07:<(str1.1, 0,0,0,rotation,z,+,0^+,180^-),(str1.2, force, x,-, translation, x,-, 10^-,8^+)>
- B08:<(str1.1,0,0,0,rotation,z,-,180^-,0^+),(str1.2, force, x,+, translation, x,+, 8^+,10^-)>
- B09:<(str1.1, 0,0,0, 0,0,0,180,180), (str1.2, force, x,-, 0,0,0, 8,8)>
- B010:<(str1.1, 0,0,0,0,0,0, 180,180),(str1.2, force, x,+, 0,0,0, 8,8)>
- B011:<(str1.1, 0,0,0, 0,0,0,180^+,360^-), (str1.2, force, x,+, translation,x,+,8^+,10^-)>
- B012:<(str1.1, 0,0,0, 0,0,0,360^-,180^+), (str1.2, force, x,-, translation,x,-,10^-,8^+)>
- B013:<(str1.1, 0,0,0,0,0,0, 360,360),(str1.2, force, x, -, 0,0,0, 10,10)>

- B014:<(str1.1, 0,0,0,0,0,0, 360,360),(str1.2,force, x, +, 0,0,0, 10,10)>

The behavior of the rotating disk constrained to rotate in particular angular range shown in Figure 5(c) can be enumerated as follows:

- B11:<(str2.1, torque, z,+, rotation, z,+,0,60), (str2.2,0,0,0,0,0,0,0,0,0)>
- B22:<(str2.1, torque, z,+, 0,0,0,60,60), (str2.2,0,0,0,0,0,0,0,0,0)>
- B33:<(str2.1, torque, z,-, rotation, z, -, 60,0), (str2.2,0,0,0,0,0,0,0,0,0)>
- B44:<(str2.1, torque, z,-, 0,0,0, 0,0), (str2.2,0,0,0,0,0,0,0,0,0)>

The behavior of the pair shown in Figure 5(d) can be enumerated as follows:

- B111:<(str3.1, force, x,-,translation, x,- ,6,3),(str3.2, 0,0,0, translation, x,-, 10,7)>
- B222:<(str3.1, force,x,+,translation,x,+,3,6),(str3.2, 0,0,0, 0,0,0,7,7)>
- B333:<(str3.1, 0,0,0,0,0,0,6,6),(str3.2, force,x,+,translation,x,+,7,10)>
- B444:<(str3.1, 0,0,0,translation,x,+,3,6),(str3.2, force, x,+, translation, x,+, 7,10)>
- B555:<(str3.1, 0,0,0,0,0,0,6,6),(str3.2, force,x,-,translation,x,-,10,7)>

So it seems possible that when F1,F2,F3,F4 are F5 given as input to a computer as a required multi-state design task , the computer should be able to preprocess these functions and place spring efforts as needed, producing F11,F22,F33, F44 and F55. The computer could search its database and retrieve the slider crank mechanism shown in Figure 5(b) as a solution, because of B01 satisfying F11 and B08 satisfying F33. For satisfying F22, it may search its database again to retrieve the pair shown in Figure 5(c) to satisfy F22, while retaining the already satisfied F11 and F33. As F44 and F55 are yet to be satisfied, the computer could again search its database to retrieve the pair shown in Figure 5(d) to satisfy F44 and F55 while retaining the already satisfied functions F11, F22, and F33. F11 would be satisfied by B01, B11 and B111. F22 would be satisfied by B22. F33 would be satisfied B08, B33 and B444. F44 would be satisfied by B333, and F55 would be satisfied by B555.

7.2 Solution2

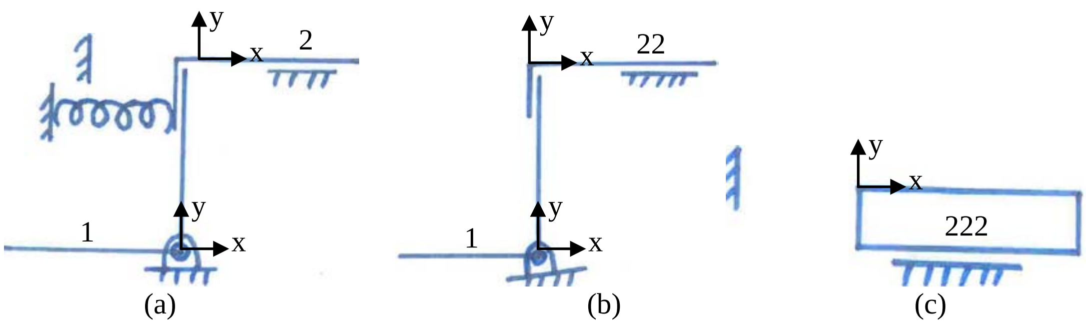


Figure 6

The mechanism shown in Figure 6(a) is a solution generated by one designer. F1 is focused on first, and the mechanism shown in Figure 6(b) is generated. For F2, the grounded obstruction before component2 as shown in Figure 6(a) is generated. For F3, a spring is connected between Component2 and the frame, satisfying F4 and F5 as well. It can be said that the mechanism shown in Figure 6(a) is a combination of a spring, the mechanism in Figure 6(b) and the pair in Figure 6(c). Component22 in Figure 6(b) and Component222 in Figure 6(c) are attached to translate together, making a single component (Component2) shown in Figure 6(a). Let us call the pair shown in Figure 6(b) as structure4 (or str4), its components Component 1 as str4.1 and Component 22 as str4.2, and the pair shown in Figure 6(c) as str5 and its translating block (Component 222) as str5.1 and the ground as str5.2.

The behavior of the mechanism shown in Figure 6(b) can be enumerated as shown follows:

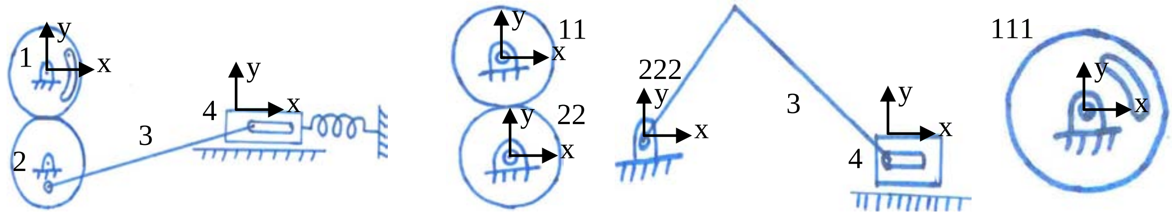
- B1:<(str4.1, torque, z,+, rotation, z,+,0,15),(str4.2, 0,0,0, translation, x,-, 8,7)>
- B2:<(str4.1, torque, z,-, rotation, z,-,15,0),(str4.2, 0,0,0, 0,0,0, 7,7)>
- B3:<(str4.1, 0,0,0, 0,0,0,0,0),(str4.2, force, x,+, translation, x,+, 7,8)>
- B4:<(str4.1, 0,0,0, rotation, z,-,15,0),(str4.2, force, x,+, translation, x,+, 7,8)>
- B5:<(str4.1, 0,0,0,0,0,0,0,0),(str4.2, force, x,-, translation, x,-, 8,7)>

The behavior of the pair shown in Figure 6(c) can be enumerated as shown follows:

- B11:<(str5.1, force,x,-,translation,x,-,10,8),(str5.2,0,0,0,0,0,0,0,0,0)>
- B22:<(str5.1, force,x,-,0,0,0,8,8), (str5.2,0,0,0,0,0,0,0,0,0)>
- B33:<(str5.1, force,x,+,translation,x,+,8,10), (str5.2,0,0,0,0,0,0,0,0,0)>

It seems possible that a computer searches its database and retrieves the pair shown in Figure 6(b) for matching F11 with B1, F33 with B4, F44 with B5 and F55 with B4. The only unmatched function is F22. For this, the computer may search its database again and retrieves the pair shown in Figure 6(c) to satisfy F22 along with the other functions when combined with the mechanism shown in Figure 6(b) and a spring. F11 is satisfied by B1 and B11. F22 is satisfied by B22. F33 is satisfied B4 and B33. F44 is satisfied by B5 and B11. F55 is satisfied by B3 and B33.

7.3 Solution3



(a) (b) (c) (d)

Figure 7.

The mechanism shown in Figure 7(a) is a solution generated by one of the designers. The gear pair (teeth not drawn) shown in Figure 7(b) is pre-selected to build a solution on it. Later, a connecting rod (Component3) and a slider are added as shown in Figure 7(a) to satisfy F1. For F2, a slot and pin arrangement is made in Component1. For F3, a spring is connected between the slider and the frame. For F44, the revolute joint between the slider and the connecting rod is modified to a higher pair. Finally, the mechanism shown in Figure 7(a) satisfies all five functions. The mechanism shown in Figure 7(a) can be seen as a combination of the gear pair shown in Figure 7(b), modified slider crank mechanism in Figure 7(c), and the pair shown in Figure 7(d). Let us call the gear pair in Figure 7(b) as structure6 (or str6), its components Component11 as str6.1 and Component22 as str6.2, the mechanism in Figure 7(c) as str7 and its components, Component 222 as str7.1 and Component4 as str7.2 and the pair in Figure 7(d) as str8 and its components, Component111 as str8.1 and the ground as str8.2.

The enumeration of gear pair shown in Figure 7(b) is as follows:

- B1:<(str6.1, torque, z,+, rotation, z,+,0,360),(str6.2, 0,0,0,rotation,z,-,360,0)>
- B2:<(str6.1, torque, z,-, rotation, z,-,360,0),(str6.2, 0,0,0,rotation,z,+,0,360)>
- B3:<(str6.1, 0,0,0,rotation,z,-,360,0), (str6.2, torque, z,+, rotation, z,+,0,360)>
- B4:< (str6.1, 0,0,0,rotation,z,+,0,360), (str6.2, torque, z,-, rotation, z,-,360,0) >

The enumeration of partial behavior of modified slider crank mechanism shown in Figure 7(c) is as follows:

- B111:<(str7.1, torque, z,-, rotation, z,-,270,240), (str7.2, 0,0,0, translation,x,-,9,8)>
- B222:<(str7.1, 0,0,0,rotation,z,+,240,270), (str7.2, force, x,+, translation,x,+,8,9)>
- B333:<(str7.1, 0,0,0,0,0,0,0,0),(str7.2, force, x,-, translation,x,-,9,8)>
- B444:<(str7.1, 0,0,0,0,0,0,0,0),(str7.2, force, x, +, translation,x,+,8,9)>

The enumeration of pair shown in Figure 7(d) is as follows:

- B11:<(str8.1, torque, z,+, rotation, z,+,0,60), (str8.2,0,0,0,0,0,0,0,0) >
- B22:<(str8.1, torque, z,+, 0,0,0,60,60), (str8.2,0,0,0,0,0,0,0,0)>
- B33:<(str8.1, torque, z,-, rotation, z, -, 60,0), (str8.2,0,0,0,0,0,0,0,0)>
- B44:<(str8.1, torque, z,-, 0,0,0, 0,0), (str8.2,0,0,0,0,0,0,0,0)>

It can be said that if the gear pair is pre-selected, computer could search its database and retrieve the modified slider crank mechanism shown in Figure 7(c) for F11, F33, F44 and F55. As F55 is still not satisfied, computer would search its database and retrieve the pair shown in Figure 7(d). In the combination of mechanisms shown in Figure 7(b)-(d), Component11 shown in Figure 7(b) rotates together synchronously with Component111 shown in Figure 7(d), and Component22 shown in Figure 7(b) rotates together synchronously with Component222 shown in Figure 7(c). F11 is satisfied by B1, B11 and B111, F22 by B22, F33 by B4, B33 and B222, F44 by B333, and F55 by B444.

8 SPATIAL DESIGN TASK REPRESENTATION

The four state door attached with a latch, described in Section 4.1, is a spatial mechanical device. The world coordinate system (X_W, Y_W, Z_W), local coordinate system for the door (x_d, y_d, z_d) and local coordinate system for the handle (x_h, y_h, z_h) are fixed as shown in Figure 2(b). All the three coordinate systems are right handed coordinate systems. The door can rotate only around the y-axis and the handle can rotate only around the z-axis. Rotational movement for the door (θ) along the y-axis and the handle (ϕ) along the z-axis are measured in counter clockwise direction as shown in Figure 2(c). As the latch is attached to the door, the handle has one degree of freedom relative to the door. The door rotates along the y-axis, which has one degree of freedom relative to the world coordinate system. So the handle has two rotational degrees of freedom relative to the global coordinate system. To ensure that no component has more than one degree of freedom, i.e. fixed axis motion, let us assume that the door is fixed and the wall rotates in a direction opposite to that of the door.

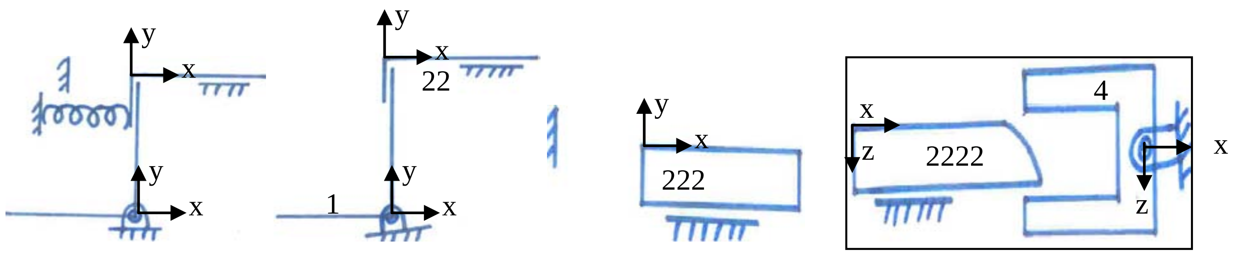
Let us call the handle as L_h and the wall as L_w. The four state spatial door attached with the latch can be represented using the proposed representation as follows:

1. Locked state: F1: <(L_h,0,0,0,0,0,0, θ_0, θ_0),(L_w,torque,y,-,0,0,0, ϕ_0, ϕ_0)>
   F2: <(L_h,0,0,0,0,0,0, θ_0, θ_0),(L_w, torque,y,+,0,0,0, ϕ_0, ϕ_0)>
2. Opening state: F3: <(L_h, torque, z,+, rotation, z,+, θ_0, θ_1),(L_w, 0, 0, 0, 0, 0, 0, ϕ_0, ϕ_0)>
   F4: <(L_h, torque,z,+,0,0,0, θ_1, θ_1),(L_w, torque, y,-,rotation, y, -, 360, ϕ_1)>
   F5: <(L_h, 0,0,0,rotation,z,-, θ_1, θ_0),(L_w, 0,0,0,0,0,0, ϕ_0, ϕ_0)>
3. Opened state: F6: < (L_h, 0, 0, 0, 0, 0, 0, θ_0, θ_0),(L_w, torque, y,-, rotation, y,-, ϕ_1, ϕ_2)>
   F7: < (L_h, 0, 0, 0, 0, 0,0, θ_0, θ_0),(L_w, torque, y, +, rotation, y,+, ϕ_2, ϕ_3)>
4. Closing state: F8: < (L_h, 0, 0, 0,0,0,0, θ_0, θ_0), (L_w, torque, y, +, rotation, y, +, ϕ_3, ϕ_0)>

There is a function, F5, in which there is motion without application of external effort; so this needs a spring or gravitational potential energy. After preprocessing the above functions, they are as follows:

- F111:<(L_h, 0, 0, 0, 0, 0,0, θ_0, θ_0), (L_w, torque, y,-, 0, 0,0, ϕ_0, ϕ_0)>
- F222: <(L_h, 0,0,0,0,0,0, θ_0, θ_0), (L_w, torque, y,+,0,0,0, ϕ_0, ϕ_0)>
- F333: < (L_h, torque, z, +, rotation, z, +, θ_0, θ_1), (L_w, 0, 0, 0,0,0,0, $\phi0$, $\phi0$)>
- F444: < (L_h, torque, z,+, 0, 0,0, θ_1, θ_1),(L_w, torque, y,-,rotation, y,-,360, ϕ_1)>
- F555: < (L_h, E_p, z,-, rotation, z,-, θ_1, θ_1),(L_w, 0,0,0,0,0,0, ϕ_0, ϕ_0)>
- F666: < (L_h, 0, 0, 0, 0, 0,0, θ_0, θ_0), (L_w, torque, y,-, rotation, y,-, ϕ_1, ϕ_2)>
- F777: < (L_h, 0, 0, 0, 0, 0,0, θ_0, θ_0), (L_w, torque, y, +, rotation, y, +, ϕ_2, ϕ_3)>
- F888: < (L_h, 0, 0, 0, 0, 0,0, θ_0, θ_0), (L_w, torque, y, +, rotation, y, +, ϕ_3, ϕ_0)>

One of the solutions for this design task is a combination of two mechanisms, shown in Figures 8(a) and 8(d). The mechanism shown in Figure 8(a) is a combination of the mechanism shown in Figure 8 (b) and the pair in Figure 8(c). Component22 shown in Figure 8(b), Component222 shown in Figure 8(c) and Component2222 shown in Figure 8(d) translates together like a single component. When door



(a) (b) (c) (d)

Figure 8. A solution for door attached with latch

is slammed, it enters into a locked state from closing state due to the convex wedged shape of Component2222. In the mechanism shown in Figure 8(d), Component2222 is allowed only to translate, and Component4 is allowed only to rotate. Component1 is the handle; Component4 is the wall with a hole.

Let us call the pair in Figure 8(d) as Structure9 (or str9), and its components Component2222 as str9.1 and Component4 as str9.2.

Enumeration for the behavior of the pair shown in Figure 8(d) is as follows:

- B111:<(str9.1,0,0,0,0,0,0,0,10,10),(str9.2,torque, y,-, 0,0,0,0,0,0)>
- B222:<(str9.1, 0,0,0,0,0,0,0,10,10),(str9.2, torque, y,+, 0,0,0,0,0,0)>
- B333:<(str9.1, force,x,-,translation,x,-,10,8),(str9.2, 0,0,0,0,0,0,0,0,0,0)>
- B444:<(str9.1, 0,0,0,0,0,0,0,8,8),(str9.2, torque, y,-, rotation,y,-,360,0)>
- B555:<(str9.1, 0,0,0,0,0,0,0,8,8),(str9.2, torque, y,+, rotation,y,+,0,360)>
- B666:<(str9.1, 0, 0, 0, 0, 0, 0, 10,9), (str9.2, torque, y,+, rotation, y,+, 10,0)>
- B777:<(str9.1, force, x,+, translation, x,+,8,10), (str9.2, 0,0,0,0,0,0,0,0,0,0)>
- B888:<(str9.1, force, x,+ ,translation, x,+,8,10), (str9.2, 0,0,0,0,0,0,0,10,350)>

Enumeration of behaviors of the kinematic pairs shown in Figure 8(b) and Figure 8(c) is discussed in Section 7.2.

Any one of the mechanisms shown in Figure 5(a), Figure 6(a), or Figure 7(a), when combined with the pair shown in Figure 8(d), forms a solution for the design task explained above, which has F111,F222,F333,F444,F555,F666,F777 and F888 as functions..

9 FINDINGS AND PROJECTIONS

All the functions that comprised the design tasks and behaviors of kinematic pairs and mechanisms have been possible to be represented using the proposed representation. Moreover, it has been possible to describe the spatial features of the effort applied, motion carried out, and the resulting changes in configuration, something hitherto not possible with existing representations. The practice followed by all designers in synthesizing solutions for a given multi-state design task can be categorized into two strategies: one is to generate an initial solution proposal for one function within the design task, and keep modifying it till it satisfies all the functions of the multi-state design task; the other is to retrieve one kinematic pair or a mechanism and keep modifying it till it satisfies all the functions of the design task. We argue that by making an appropriate set of kinematic pairs (and mechanisms) and their modified versions within a database, and retrieving an appropriate combination of pairs and mechanisms, it should be possible to generate at least the same, and possibly wider solution space for a given multi-state design task, as/than generated by all these designers taken together. It can be observed that the multi-state design synthesis is an iterative process. If a single kinematic pair or mechanism does not satisfy all the functions, another kinematic pair or mechanism can be retrieved to satisfy the unsatisfied requirements, while maintaining the already satisfied requirements. This process would go on till all the functions are satisfied. These kinematic pairs and mechanisms could be stored in a computer database, and a suitable retrieval process of pairs and mechanisms could be used to generate solutions for a given multi state design task. Each component in a pair would be associated with a motion axis. Along this motion axis only, newly retrieved pairs or mechanism can be attached. As to which kinematic pair or mechanism should be retrieved depends on the motion axis and associated component to which it should be attached. Solutions can also be developed by pre-selecting a kinematic pair or mechanism, as explained in Section 7.3. Modification of interfaces can be useful to satisfy some functions, as seen in Section 7.3. The solution to a multi-state design task can be a single kinematic pair or mechanism, or a combination of these. Preprocessing of the functions may be necessary to determine where springs are needed and avoid redundant springs. Spring or gravitational potential energy can be used interchangeably, when motions are needed without external effort.

10 CONCLUSIONS AND FUTURE WORK

A richer representation than currently available for multi-state design tasks and functions has been developed. The behaviors of kinematic pairs and mechanisms are possible to be represented using the representation. Further work involves developing heuristic rules for retrieving, combining or modifying pairs and mechanisms to be used in supporting the multi-state design synthesis process.

REFERENCES

[1] Dixon J.R. and Welch R.V. Guiding conceptual design through behavioral reasoning. *Research in Engineering Design,* 1994, 6,169–188.

[2] National Research Council. *Improving engineering design: designing for competitive advantage,* 1991(National Academy Press ,Washington, DC).

[3] Mabie H.M. and Reinholtz C.F. *Mechanisms and dynamics of machinery, Volume 4,* 1987 (Wiley, New York).

[4] Seering W.P. and Ulrich K.T. Synthesis of schematic descriptions in mechanical design. *Research in Engineering Design*,1989,1,3–18.
[5] Adams J.L. *Conceptual blockbusting*, 1986(Addison Wesley, New York).
[6] Chakrabarti A and Bligh T.P. An approach to functional synthesis of solutions in mechanical conceptual design. Part I: Introduction and knowledge representation. *Research in Engineering Design*, 1994, 6, 127–141.
[7] Jansson D.G. and Smith S.M. Design fixation. *Design Studies*, 1991, 22(1), 3–11.
[8] Li C.L. *Conceptual design of single and multiple state mechanical devices: an intelligent CAD approach, PhD thesis*, 1998(Department of Mechanical Engineering, University of Hong Kong)
[9] Langdon P.M. and Chakrabarti A. *Browsing a large solution space in breadth and depth in. In The 12th International Conference in Engineering Design (ICED'99)*, Vol. 3, 24-26 Aug 1999.
[10] Forbus K.D. *Qualitative reasoning, CRC Hand-book of Computer Science and Engineering*,1996 (CRC Press).
[11] Forbus K. D. Introducing actions into qualitative simulation. *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, California,1989,pp.1273–1278.
[12] Nielsen P. *A qualitative approach to rigid body mechanics. Technical Report No.UIUCDCS-R88 -1469,* 1988(Department of Computer Science, University of Illinois at Urbana-Champaign).
[13] Brown J.S. and de Kleer J. A qualitative physics based on confluences. *Artificial Intelligence*, 1984, 24 (1-3), 7-83.
[14] Somasekhara Rao Todeti and Amresh Chakrabarti. An empirical model of process of synthesis of multiple state mechanical devices, Vol.4, Pages. 239-250, *Proceedings of the 17th International Conference on Engineering Design (ICED'09)*, Stanford, CA, August 24-27, 2009.
[15] Elisha Sacks and Leo Joskowicz, *The Configuration Space Method for Kinematic Design of Mechanisms* 2010, (The MIT Press)
[16] Alex C. Starling and Kristina Shea. Virtual synthesiers for mechanical gear systems. *In International Conference on Engineering Design*, *ICED’05*, Melbourne, August 15-18,2005.
[17] Kota S. and Chiou S. J. Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. *Research in Engineering Design*, 1992, 4, 75-87.
[18] Murakami T. and Nakajima N. Mechanism concept retrieval using configuration space. *Research in Engineering Design*, 1997, 9, 99- 111.
[19] Chakrabarti A. and Bligh T. P. Functional Synthesis of Solution-Concepts in Mechanical Conceptual Design. Part II: Kind Synthesis, *Research in Engineering Design*, Vol. 8, No. 1, pp-52-62, 1996.
[20] Chakrabarti A. and Bligh T. P. An approach to functional synthesis of solutions in mechanical conceptual design. Part III: Spatial configuration. *Research in Engineering Design*, 1996, 2, 116-124.
[21] Sun K. and Faltings B. FAMING: Supporting innovative mechanism shape design. *Computer Aided Design*, 1996, 28(3), 207 – 216.
[22] Subramanian D. and Wang C. S. Kinematic synthesis with configuration spaces. *Research in Engineering Design*, 1995, 7, 193-213.
[23] Hoover S. P. and Rinderle J. R. A synthesis strategy for mechanical devices. *Research in Engineering Design*, 1989, 1, 87-103.

Contact: Prof. Amaresh Chakrabarti
Indian Institute of Science (IISc), Bangalore
Centre for Product Design and Manufacturing (CPDM)
Bangalore
India
Tel: +91 80 2293 2922
Fax: +91 80 2360 1975
E-mail: ac123@cpdm.iisc.ernet.in
URL: http://www.cpdm.iisc.ernet.in/people/ac/ac.htm

Amaresh Chakrabarti is Professor in the Department of Centre for Product Design and Manufacturing (CPDM) at Indian Institute of Science, Bangalore, India. His research interests are functional

synthesis, design Creativity, design Methodology, collaborative design, eco-design, engineering design, design synthesis, requirements management, knowledge management, computer aided design, and design for variety.

Somasekhara Rao Todeti is Research Scholar in the Department of Centre for Product Design and Manufacturing (CPDM) at Indian Institute of Science, Bangalore, India. His research interests are engineering design, functional synthesis, mechanisms and machines, geometric modeling, and computer-aided design.