Research in Engineering Design (2023) 34:477–495
https://doi.org/10.1007/s00163-023-00420-8

ORIGINAL PAPER

Empirical studies on conceptual design synthesis of multiple-state mechanical devices

Anubhab Majumder[1] · Somasekhara Rao Todeti[2] · Amaresh Chakrabarti[1]

Received: 14 June 2022 / Revised: 28 March 2023 / Accepted: 10 June 2023 / Published online: 21 June 2023

Abstract
Conceptual design synthesis, which focuses on generating solution alternatives, has a significant impact on the cost and quality of the final product. The development of radically new and significantly better solutions requires generation and exploration of a large solution space. Most of the existing literature on conceptual design synthesis of mechanical devices: (1) is on synthesising device concepts for a single relation of 'converting one input set to another output set'—this is called single-state design synthesis (SSDS); and (2) primarily employs 'composition of building blocks' as the approach for the synthesis of concepts. Multiple-state design synthesis (MSDS), on the other hand, refers to synthesising device concepts for more than one relation between an input set and an output set. Since not much literature is available on studies of MSDS, it is essential to understand how and how well designers currently carry out MSDS. This knowledge can be used as a benchmark and a source of knowledge for developing a prescriptive support to improve MSDS. Therefore, the objective of the work presented in this paper is to obtain a better understanding of this process by carrying out empirical studies on multiple-state synthesis.

Keywords Multiple state · Mechanical devices · Conceptual design · Design synthesis · Empirical studies

1 Introduction

Conceptual design synthesis of mechanical devices can be considered as an activity of transforming a perceived need into a solution concept that utilises mechanical engineering principles to satisfy the need. Conceptual design synthesis, part of which is referred to as type synthesis in mechanisms (Johnson 1978; Yan 1998; Pozhbelko 2019), has a significant influence on the cost and quality of the final product (Hoover and Rinderle 1989; Zhang et al. 2020). To develop a radically better solution, it is very important to generate and explore a wide range of solutions (Liu et al. 2003). Conceptual synthesis of mechanical devices is a difficult task (Chakrabarti and Bligh 1996b; Pons and Raine 2005). The difficulties generally arise from a mechanism's intrinsically complicated geometrical and topological characteristics (Tian et al. 2005). Usually, the process of conceptual design depends on the designer's ingenuity, intuition, and experience (Tsai 2000). However, this approach often leads to a bias toward a limited set of solutions and cannot ensure the identification of an adequate set of feasible alternatives within the time constraint (Chakrabarti, and Bligh 1994). As per the survey conducted by Sacks and Joskowicz (2010), the majority of mechanisms are planar, having one or two degrees of freedom, and with motion along fixed axes. Therefore, the scope of this research is currently limited to such mechanical devices (i.e. mechanisms or machines).

Generating device concepts for converting one set of input (efforts or motions) to another set of output (efforts or motions) is called single-state design synthesis (SSDS) (Li et al. 1999b). In a single-state design task (SSDT), there is a single relation between input set and output set. In an SSDT, with 'm' inputs and 'n' outputs, if $m = 1$ and $n = 1$, it is a Single Input–Single Output SSDT; if $m = 1$ and $n > 1$, it is a Single Input–Multiple Output SSDT; if $m > 1$ and $n = 1$, it is a Multiple Input–Single Output SSDT; if $m > 1$ and $n > 1$, it is a Multiple Input–Multiple Output SSDT. For example, 'generate solutions for converting an input effort to two outputs: one of a linear motion and the other of an angular motion' (Chakrabarti and Bligh 1996b) is a Single Input–Multiple Output (SIMO) SSDT. Such a device may

✉ Anubhab Majumder
anubhabm@iisc.ac.in

[1] Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore, India

[2] Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, India

be useful, for instance, in taking a single input to create two outputs: one, e.g. to latch/unlatch a door, and the other to change the display to show that the door is now 'locked' or 'unlocked'. Zhang et al. (2015) presented such Multiple Input–Multiple Output (MIMO) mechanisms as hybrid mechanisms where multiple loop kinematic behaviours are captured by qualitative dual state vectors.

Generating solution concepts of a mechanical device intended to perform multiple operations, where an operating state can be described using one or more input set to output set relations, is called multiple-state design synthesis (MSDS) (Li et al. 1999a, b; Todeti and Chakrabarti 2009). In this kind of design task, the function (called as 'elemental function' or 'EF') of a mechanical device is defined by effort–motion relations between input components and output components. The term effort refers to a force or torque, and the motion can be either a rotation or a translation. Here, the concept of effort resembles the effort variable used in bond graph language and motion can be considered as the time integral of flow variable (Karnopp and Margolis 1979). The components on which efforts are applied are called input components. The components on which motions are desired are called output components. In this work, the input and output components set is termed input–output Pair or 'I–O pair'. An operating state can have one or more 'elemental functions' (EFs). Each EF has one effort–motion relation between a set of input components and a set of output components. Such a set of relations could, for instance, be useful in specifying four, inter-connected, intended tasks for a door lock, i.e. locked state, opening state, opened state, and closing state, respectively (and specifying all four, and not just latch/unlatch a door, as in the earlier, SIMO example). Liu et al. (2015) addressed such devices as 'multi-modal' systems where the word 'mode' referred to a certain functioning arrangement or condition. However, the definition of 'multiple state' given by Li et al. (1999a) is adopted for this work.

The present study considers specifically the type synthesis of multiple-state mechanical devices. Type synthesis is the process of identifying or developing potential mechanism structures to carry out a specific task or set of tasks without considering the component's dimensions (Olson et al. 1985). The overall aim of this research is to support designers synthesise a large solution space for design tasks comprising multiple operating states. Exploring a large solution space is important in conceptual design as it influences novelty (Srinivasan and Chakrabarti 2010a, b), which is a key indicator of creativity (Sarkar and Chakrabarti 2011). A review of the existing literature on conceptual design synthesis of mechanical devices, discussed in Sect. 2, revealed that most of the existing methods for conceptual design synthesis of mechanical devices largely employ compositions of kinematic building blocks or primitive structures. These methods primarily focused on the type synthesis of single-state design problems. In the case of MSDS, the compositional approach does not always ensure the exploration of multiple alternative design solutions. The behaviour of an existing kinematic building block can be altered, or multiple behaviours can be achieved by modifying the geometric features in the components of a building block. The modification of building blocks plays a key role in MSDS, which has not been considered by existing works (Li 1998). Therefore, new support systems or enhancements to current design synthesis tools are required to assist designers in generating a substantial variety of feasible alternative solutions for MSDS. In literature, it is found that type synthesis problems are difficult to describe algorithmically (Jimenez et al. 1997). The existing design support tools require an algorithm guided by a well-defined and systematic procedure, and this often require a descriptive understanding of the design synthesis process (Finger and Dixon 1989). In this paper, empirical studies on MSDS are reported with the purpose of developing a detailed understanding of the MSDS process as carried out by engineering designers. This should help in developing a descriptive model for MSDS. To adapt the descriptive model for future computer implementation, it is also important to understand which parts of the MSDS process are difficult or require domain knowledge or individual competency. As future implications, these understandings will help in providing appropriate prescriptive support to the designers, allowing them to explore a larger solution space by following a systematic MSDS process.

2 Literature review

2.1 Conceptual design synthesis of single-state mechanical devices

Most of the research on conceptual design synthesis of mechanical devices is limited to SSDS and can be broadly classified into case-based and process-based approaches (He et al. 2014). The case-based approach (Domeshek et al. 1994; Han and Lee 2006; Navinchandra et al. 1991; Prabhakar and Goel 1998; Ulrich and Seering 1988; Zhang et al. 2001) is a technique where past solutions are reused or adapted to solve new problems. Generally, the method begins with a knowledge base abstracted from design cases and then they are modified to meet the new specifications. For example, Han and Lee (2006) introduced virtual function generators (VFGs) to conceptualise and capture the underlying design principles of existing mechanical devices. A set of such VFGs is stored as a library of design cases. In response to a design specification, pertinent VFGs (a design case or portion of a design case) are retrieved from the case library and combined according to a synthesis strategy to match the specified motion requirements. This results in feasible design solution alternatives for a desired mechanism. In contrast, the process-based approach starts with the desired functionality of the device and synthesises a structure that satisfies it. This approach of synthesis usually generates intermediate behavioural specifications and then combines identified kinematic building blocks or primitive structures (e.g. levers, shafts, gears etc.) that generate those behaviours (Chakrabarti and Bligh 1994, 1996a, b; Chen et al. 2005; Chiou and Kota 1999; Ding et al. 2012; Finger and Rinderle 2002; Han et al. 2020; Hoover and Rinderle 1989; Kota and Chiou 1992; Li et al.1996; Li and Wang 2019; Moon and Kota 2002; Murakami and Nakajima 1997; Starling and Shea 2005; Subramanian and Wang 1995). For example, Kota and Chiou (1999) built a database of primitive structures, including rotational, translational, and helical motion, and developed a matrix-based approach to synthesising mechanisms. The desired functional requirements are given as a motion transformation matrix (MTM). This intended MTM for a design task is further decomposed into a product of MTMs by transformation rules; the decomposed matrices are searched against the MTMs of existing building blocks to find feasible combinations of building blocks to satisfy the design task. In the existing literature, different schemes for representing behavioural specifications have been proposed; some of the major schemes used by most researchers are: matrix-based representation (Chiou and Kota 1999; Kota and Chiou 1992), set of input–output vectors (Chakrabarti and Bligh 1996a, b; He et al. 2014; Moon and Kota 2002), parallel grammar (Li and Wang 2019; Starling and Shea 2005), topology graphs (Ding et al. 2012; Han et al. 2020) and configuration space (Murakami and Nakajima 1997; Subramanian and Wang 1995).

2.2 Conceptual design synthesis of multiple-state mechanical devices

Li et al. (1999a; b) developed ADCS (automatic design by configuration space), a computational tool for generating a solution for a given multiple-state design task (MSDT). ADCS uses the method of combinatorial retrieval of building blocks, which is simply a hierarchical search from requirement space to solution space, and the solution generated by ADCS is a network of building blocks. ADCS generates only one solution for a given MSDT. If one more solution is to be generated, some of the building blocks in the present solution need to be removed from the ADCS's database and run the program again. ADCS does not consider modification of building blocks. If some of the EFs of a given MSDT are not satisfied by a building block, ADCS searches in its database and retrieves another building block and adds to the existing building block(s), instead of modifying the existing building block. There is no guarantee that a compatible building block would exist in the ADCS database, in which case no solution could be generated. Multiple-state mechanical devices (MSMD) can also be considered as metamorphic mechanisms which have the ability to change their topology under different operating states (Pucheta et al. 2012; Zhang and Dai 2009; Zhang et al. 2011). Pucheta et al. (2012) proposed a combinatorial method for the topological synthesis of a circuit breaker mechanism by using graph enumeration; however, in their work, the existence of energy-storing components (such as springs) are ignored during the enumeration of alternative mechanisms. In contrast, Zhang et al. (2011) considered spring components and proposed a synthesis approach based on the morphological matrix (Pahl and Beitz 1996). This approach was illustrated by a mechanism design example, in which the input effort was fixed to a particular driving component in all operating states, which is not necessarily true for every MSDT. So, the composition of the building blocks approach that is widely used in SSDS does not work well for MSDS. The modification process, as observed from the empirical studies (presented in this work), is a core element of MSDS in generating a variety of solutions, which is highly limited in the compositional approach currently used in literature.

3 Research approach

The purpose of empirical studies is to prove or contradict existing proposals or to generate new knowledge about designing, with which better models or support for designing can be developed (Gero and Mc Neill 1998; Stauffer and Ullman 1988). Carrying out empirical studies should help understand the design process better, which in turn should assist in developing support, which includes computational methods and tools (Fricke 1996; Mulet and Vidal 2008). As discussed in Sect. 2.2, research into MSDS is relatively sparse; very few approaches for supporting the synthesis of MSMD have been developed; even these are inadequate in supporting the synthesis of a wide variety of solutions. Therefore, the objective of the work presented here is to obtain a better understanding of this process by carrying out empirical studies on MSDS. The empirical studies were conducted with 12 participants. The design task given to the participants is formulated from a door-latch device, which comprises two operating states (opening and closing) with five EFs. The participants were asked to develop a design solution in the form of a mechanism that should satisfy all five EFs of the door-latch device. No time limit was imposed for completing the synthesis task. The synthesis processes were video recorded, and the participants were asked to think aloud and draw schematic diagrams of the proposed solution concepts on paper. All recorded videos are analysed, and finally, as a research outcome, an empirically evaluated descriptive model of the design synthesis process for MSMD is developed.

4 Empirical studies

4.1 Design task used in the empirical studies

The design task used in the empirical studies is constructed from a door-latch device, which has a handle-block as the input–output components or I–O pair, and a two-state design task comprising five EFs ($f_i, i = 1 - 5$). The two operating states are the opening state and the closing state. Here, the EFs f_1, f_2 and f_3 correspond to the opening state, and the EFs f_4 and f_5 correspond to the closing state. The handle and block have two configuration parameters, the orientation of handle: θ and the position of block: x, respectively, with respect to the world coordinate system. The five EFs in the design task (see Fig. 1) to be performed by the mechanical device (having handle and block as I–O pair) are as follows:

- f_1: When the handle is at $\theta = \theta_1$, and the block is at $x = x_1$, if an effort is applied on the handle around its z-axis in the clockwise direction, it should rotate it from $\theta = \theta_1$ to $\theta = \theta_2$, and simultaneously the block should translate from $x = x_1$ to $x = x_2$ in the positive direction along its x-axis.
- f_2: When the handle is at $\theta = \theta_2$, and the block is at $x = x_2$, if an effort is applied to the handle around its z-axis in the clockwise direction, it should not move any further from $\theta = \theta_2$, and the block also should not move from $x = x_2$.
- f_3: When the handle is at $\theta = \theta_2$, and the block is at $x = x_2$, if the effort is released from the handle, it should rotate around its z-axis in the anti-clockwise direction from $\theta = \theta_2$ to $\theta = \theta_1$, and simultaneously the block should translate along its x-axis in the negative direction from $x = x_2$ to $x = x_1$.
- f_4: When the handle is at $\theta = \theta_1$, and the block is at $x = x_1$, if an effort is applied on the block along its x-axis in the positive direction, it should translate from $x = x_1$ to $x = x_3$ along its x-axis in the positive direction, but the handle should not move from $\theta = \theta_1$.
- f_5: When the handle is at $\theta = \theta_1$ and block is at $x = x_3$, if the effort is released from the block, it should translate along its x-axis in the negative direction, but the handle should not move from $\theta = \theta_1$.

4.2 Subjects and experimental procedure

A total number of 12 subjects participated in this study. All subjects were postgraduate students (Masters' or PhD) with

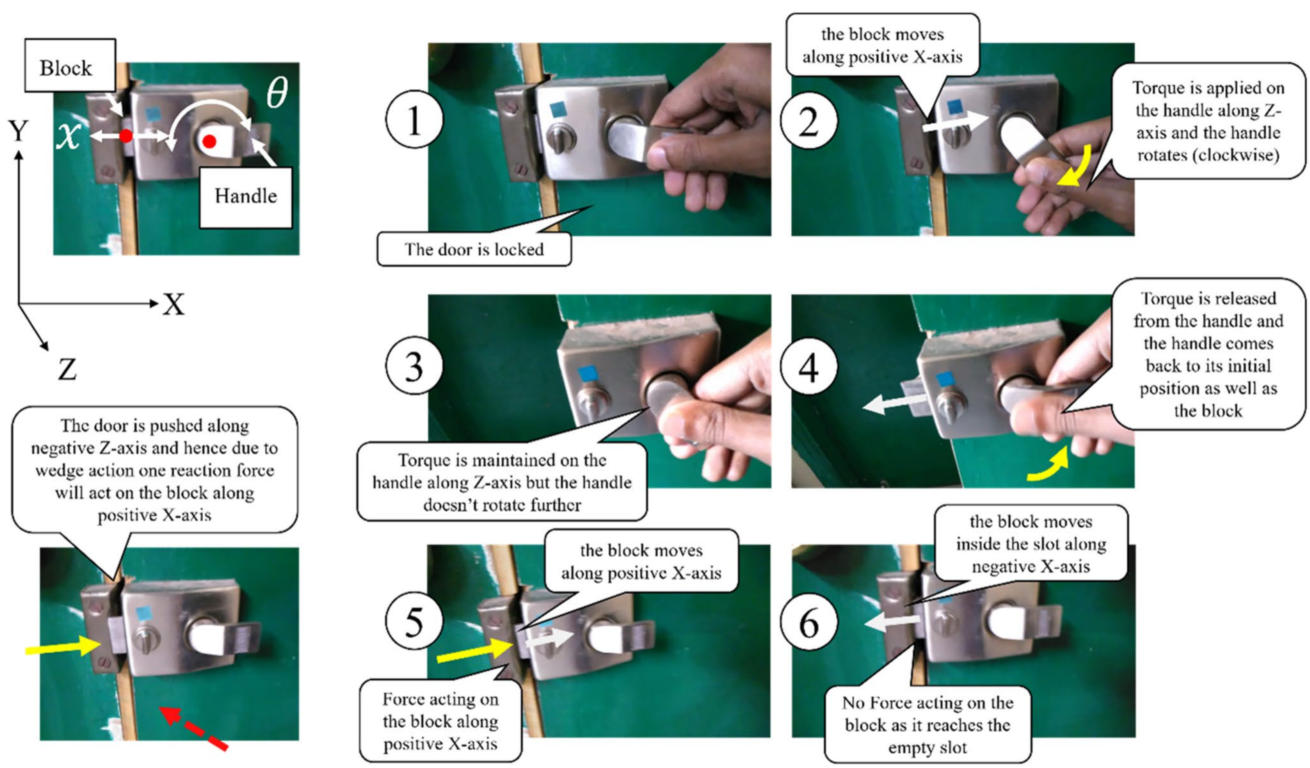


Fig. 1 Description of the MSDT provided to the subjects

a bachelor's degree in mechanical engineering and proficient with the theory of machines/ mechanisms. The experiments were carried out in an observatory (controlled environment) in the presence of an instructor. Each experiment was conducted with one subject at a time. The design task document was given to the subjects to provide the description of the MSDT with five EFs, as shown in Fig. 1. Apart from the MSDT document, a catalogue was given to the subjects for their reference, containing various known mechanisms from the theory of machines/mechanisms course. The subjects were allowed to search and choose mechanisms from the catalogue while generating solutions for the given MSDT; however, using the catalogue was not mandatory for the subjects during the synthesis activity. The subjects were asked to develop one design solution that must satisfy all five EFs in the MSDT document. In addition, they were asked to think aloud while they carried out their synthesis processes and to draw schematic diagrams/sketches of the intermediate steps of the design concept and outcome. The verbal data were captured through video recording, and sketches were captured using the camera. No time limit was imposed. The experiment was stopped once the subject reached one feasible solution (which should satisfy all five EFs). They were free to explore as many ideas/ sub-solutions as possible for a particular EF until they reached one feasible solution which satisfies all the EFs. One of the authors played the role of the instructor. During the experiment, the instructor was present to help participants understand the task, encouraged them to think aloud, and ensured they made sketches of all intermediate steps. The instructor's role was also to verify the feasibility of the solution(s) generated by the subjects. It is important to note that, in this study, the feasibility of a solution is verified only in terms of functionality, i.e. the solution should be able to perform all five EFs given in the MSDT. Other aspects of evaluating conceptual design solutions, such as manufacturability, reliability, and cost (Liu et al. 2000), have not been considered by the instructor for verifying feasibility.

5 Data analysis

5.1 Representing MSDT with elemental functions and specifications graph

EF of a mechanical device with 'n' components can be written as, $< (E_1, M_1), (E_2, M_2), ...(E_n, M_n) >$. Here, $E_i, M_i (i = 1{-}n)$ stands for the effort and motion of n^{th} component. Effort and motion can be represented using qualitative values of $+, -, 0$. E_i = '+' means effort is applied on the component in the positive direction (or anti-clockwise), '−' means effort is applied on the component in the negative direction (or clockwise), '0' means effort is not applied on the component. M_i = '+' means the component undergoes motion in the positive direction; '−' means the component undergoes motion in the negative direction; '0' means the component does not undergo any motion. For example, the EFs of the MSDT considered in the empirical studies (refer Sect. 4.1) can be written as: $f_1 :< (-,-), (0,+) > ; f_2 :< (-,0), (0,0) > ; f_3 :< (0,+), (0,-) > ; f_4 :< (0,0), (+,+) > ; f_4 :< (0,0), (0,-) >$.

The graph shown in Fig. 2 is termed as 'specifications graph' (Li et al. 1999a). This graph depicts the changes in the configuration of the door-latch system during its opening and closing states. Each node of the graph signifies a configuration state of the input/output components. The initial positions of the handle and the block with respect to the global coordinate system (X, Y, Z) can be labelled as $C_1(\theta_0, x_0)$. The EF f_1 leads to a change in configuration from $C_1(\theta_0, x_0)$ to $C_2(\theta_1, x_1)$, and during the closing state, arc f_4 ends at another configuration $C_3(\theta_{0,} x_1)$. However, the arc corresponds to the EF f_2, which starts and ends at the same node implying no change in configuration.

5.2 Types of elemental functions

The EFs of a MSMD can be broadly characterised into four types as follows (see Fig. 3):

- *Type* − 1: When an I–O pair is at configuration (C_1), effort(s) is applied to some component(s) of the I–O pair, and the I–O pair moves to another configuration (C_2), see Fig. 3a.
- *Type* − 2: When an I–O pair is at configuration (C_1), effort(s) is applied to some component(s) of the I–O pair, but the I–O pair does not move, see Fig. 3b.
- *Type* − 3: When an I–O pair is at configuration (C_1), effort(s) is not applied on any component of the I–O pair, but the I–O pair moves to another configuration (C_2), see Fig. 3c.
- *Type* − 4: When an I–O pair is at configuration (C_1), the effort is not applied to any component of the I–O pair, and the I–O pair does not move to another configuration, see Fig. 3d.

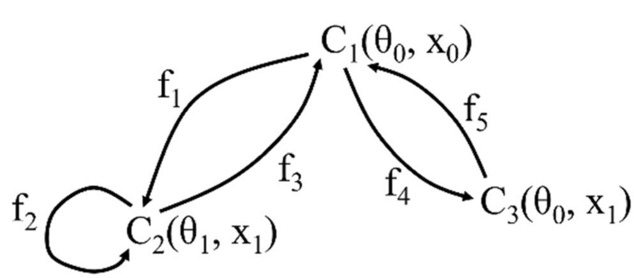


Fig. 2 The specifications graph of the given MSDT

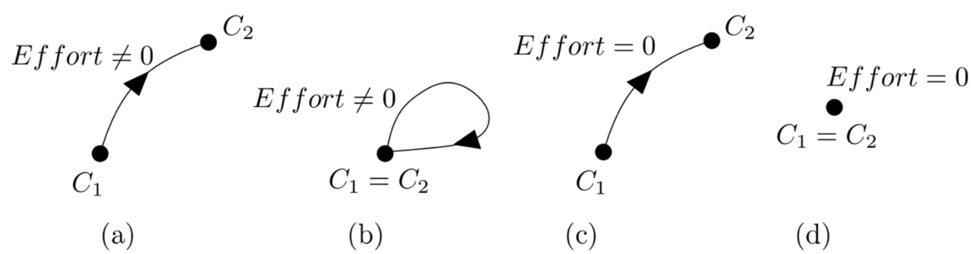


Fig. 3 Types of EFs

Type − 3 EF can be called a dependent EF because for *Type* − 3 to happen, there must exist a *Type* − 1 before. *Type* − 1, *Type* − 2 and *Type* − 4 can be called independent EFs. In the MSDT used for the empirical studies, f_1, and f_4 are *Type* − 1, f_2 is *Type* − 2 and, f_3 and f_5 are *Type* − 3 EFs.

5.3 Coding scheme

The captured videos of all twelve subjects were transcribed by the authors. The resulting protocol data were analysed by categorising each transcribed speech and action (used whenever the subject did some work but forgot to think aloud) with respect to the following categories:

Activities: An activity in designing can be defined as a deed of problem finding or problem-solving (Srinivasan and Chakrabarti 2010a, b). In this study, the primary level activities during the design synthesis process considered for coding were *Analyse*, *Generate*, *Evaluate*, and *Modify*. The activities are defined as follows:

- *Analyse* is to consider something in detail to discover its essential features. In this study, the instances of verbalisation of given task descriptions by the subjects were coded as *analyse* activities.
- *Generate* is to produce something. *Generate* has two secondary level activities: *Select* and *Retrieve*. *Select* is to choose something from several alternatives given in the mechanism catalogue [it refers to the information representation of common domain knowledge (Stauffer and Ulman 1991)]. *Retrieve* is to bring back something from memory.
- *Evaluate* is to check whether an EF is satisfied by the solution proposal.
- *Modify* is to change something. *Modify* has four secondary level activities: *Add, Replace, Remove,* and *Incorporate*. *Add* is to combine something with an existing solution. *Replace* is to substitute a thing for another. *Remove* is to delete something from the existing solution. *Incorporate* is to merge something with some other thing already in existence.

Requirements: Requirements can be defined as the intended technical characteristics of the design or constraints imposed on the design. In the literature, requirements are classified in several ways, such as functional, non-functional, solution-neutral, solution-specific, etc. Functional requirements specify the system's behaviour, i.e. what the system should do, and non-functional requirements stipulate quality constraints (such as performance, usability, etc.) on implementing these functional requirements (Sommerville and Sawyer 1997). Solution-neutral requirements are those that are not specific to any of the designer's solutions (Jagtap et al. 2014). Nidamarthi (1999) classified requirements into *given requirements* and *solution-specific* requirements. In the current study, the design task document provides the behavioural specifications of the door-latch device, and thus all the EFs can be considered as *given requirements* and can also be identified as functional and solution-neutral requirements. In contrast, the requirements specific to the subject's solution can be considered *solution-specific* requirements. Chakrabarti et al. (2004) stated that a *solution-specific* requirement is a contextualised version of the *given requirement* such that it retains those features of the existing design that already fulfil other requirements. Here, the concepts of EFs, and specifications graph are used for identifying and coding the *solution-specific requirements*. This helps to analyse the subjects' synthesis processes in more detail. With respect to the given EFs for the MSDT, the *solution-specific requirements* were assumed to be:

- *Req. 1:* Finding the initial solution for f_1 (a *Type-1* function) where a rotary motion needs to be converted to a translatory motion and the configuration changes from $C_1(\theta_0, x_0)$ to $C_2(\theta_1, x_1)$.
- *Req. 2:* Finding an intermediate solution that satisfies f_2 (a *Type-2* function) while maintaining the previously satisfied EF. Therefore, the initial solution which currently exhibits a *Type-1* function, needs to support a subsequent *Type-2* function at configuration $C_2(\theta_1, x_1)$.
- *Req. 3:* Finding an intermediate solution that satisfies f_3 (a *Type-3* function) while maintaining the previously satisfied EFs. In this case, both input and output components have to move from configuration $C_2(\theta_1, x_1)$ to $C_1(\theta_0, x_0)$ but no effort is applied from any external sources.
- *Req. 4:* Finding an intermediate solution that satisfies f_4 (a *Type-1* function) while maintaining the previously satisfied EFs. In this case, the current solution which already satisfies an existing *Type-1* function f_1, needs to

perform another *Type-1* function f_4 which moves the I–O pair from $C_1(\theta_0, x_0)$ to $C_3(\theta_0, x_1)$. This condition arises because the handle remains idle in during closing state.

- *Req. 5:* Finding a final design solution that satisfies f_5 (a *Type-3* function) while maintaining the previously satisfied EFs. This scenario is also similar to *Req. 3* where the I–O pair has to move from configuration $C_3(\theta_0, x_1)$ to $C_1(\theta_0, x_0)$ but no effort is applied from any external sources.

Based on the coding scheme discussed above, an activity-requirement sequence chart was prepared for each subject. For example, Fig. 4 shows the activity-requirement sequence chart of subject 3. The transcribed data for each subject was encoded in terms of the identified activities and the corresponding requirement under consideration during a time segment. In case of any absence of activities (where the subjects remained idle), those specific time segments were left blank in the sequence chart.

5.4 Understanding the subjects' synthesis strategy to approach the design task

It has been observed that the common practice followed by each subject is to select one of the EFs for which a fully or partially satisfying solution proposal is generated, and this proposal is kept on being modified until all the EFs in the design task are satisfied. An example of such synthesis process carried out by one of the subjects (Subject 7) is described as follows:

After analysing the given five EFs, the subject *generated* a rack-and-pinion mechanism as an initial solution proposal for *Req. 1* as shown in Fig. 5a where the handle was attached with the pinion with a fixed joint and the rack was acting as the block of the door-latch device. The subject realised that the rack was supposed to go towards positive x-direction with a clockwise rotation of the handle and thus modified the current solution by adding another gear as shown in Fig. 5b. To satisfy *Req. 2,* the motion of the rack was arrested after a certain limit by *adding* a stopper as shown in Fig. 5c. Next, the solution was further *modified* by *adding* a linear spring between the rack and ground to satisfy *Req. 3* as shown in Fig. 5d. While considering *Req. 4*, the subject observed a contradiction between EFs f_1 and f_4 where the effort applied on the rack on positive x-direction should not cause any rotation of the handle and thus the handle should remain detached from the rack during this operating state. The subject solved this contradiction by *replacing* the fixed joint between the handle and the gear with a variable constraint joint as shown in Fig. 5e where the joint allows the transfer of effort from handle to gear while the handle is rotated in a clockwise direction, but the joint disengages the gear and handle with the gear rotates in a clockwise direction causing by the motion of the rack in the positive x-direction. At this point, to stop the rotation of the handle due to its self-weight, the subject decided to *add* a torsion spring between the handle and the ground (see Fig. 5e). Finally, the subject analysed f_5 and realised that the current solution was already satisfying the desired requirement, i.e. *Req. 5,* and thus stopped the synthesis process with a final design solution shown in Fig. 5f. Overall, the subject took 38 min to arrive at the final design solution, with *Req. 4* taking the most time (29 min), followed by *Req. 1* (6 min),

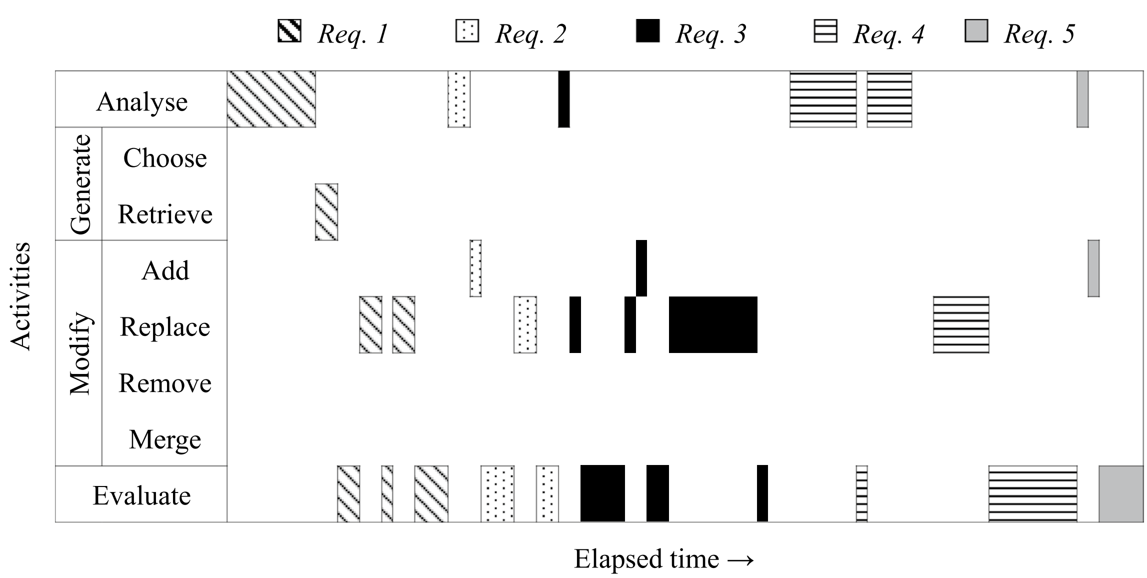


Fig. 4 The encoded data of Subject 3

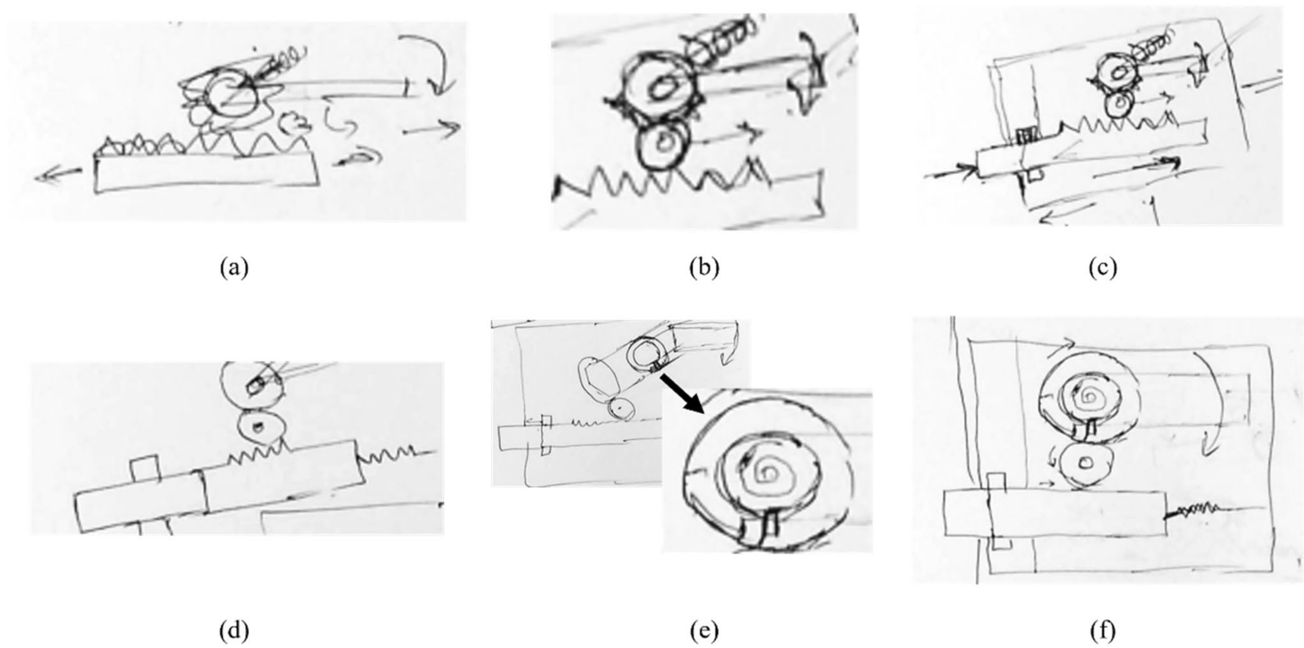


Fig. 5 Outcomes of various activities captured during the synthesis process carried out by Subject 7

and the rest of the time spent modifying the solution for the other three requirements.

Table 1 shows all the final design solution diagrams produced by the subjects along with the description of the initial mechanism *generated* for *Req. 1* and the further *modifications* performed on it to satisfy subsequent requirements. The discarded solutions or modifications are not shown in the table.

5.5 A 'preferred' synthesis approach for the given MSDT derived by analysing the final design solutions

Based on the above observation, one can understand that for the given MSDT, the selection of the initial solution for f_1 influences the final design solution. For example, if a mechanism (Slider-crank, rack & pinion, etc.) was chosen as an initial solution for starting the synthesis process, then the final design solution was a *modified* version of that mechanism, and no other mechanisms were *generated* and combined with the initial solution. It is important to note that the MSDT given to the subjects was described (as shown in Fig. 1) in a step-by-step manner and it was expected from the subjects to consider the functional requirements in the same sequence. Thus, a 'preferred' sequence of requirement consideration can be hypothesised (shown in Fig. 6) for the given MSDT which should start by considering *Req. 1* followed by *Req. 2, Req. 3, Req. 4* & *Req. 5* respectively. To satisfy each requirement, a sequence of *activities* needs to be performed. The sequence starts with an *analyse* activity where the subject tries to understand the requirement and ends with a successful *evaluate* activity where the proposed solution satisfies the concerned requirement. In between these two activities, the subject may perform one or more *generate/modify/evaluate* activities. For the 'preferred' approach, the *Generate* activity should only occur while solving *Req. 1*. The 'preferred' synthesis approach described in this section is built on the common understanding developed after analysing the final design solutions synthesised by the subjects (as described in Sect. 5.4) but doesn't claim to be the best approach for the given MSDT. The purpose of introducing this 'preferred' assumption is to assess the synthesis process efficiency of each subject's synthesis strategy which has been discussed in the following sub-section.

5.6 Synthesis process efficiency

In this empirical study, the subjects' actual synthesis processes are compared to the 'preferred' process discussed above, where the sequence of considering requirements was expected to run from *Req. 1* to *Req. 5* monotonically as a function of the elapsed time. This assumption implies that in a 'preferred' scenario, the subjects would have all the necessary knowledge to *generate* and *modify* a solution to satisfy the given requirements. As a result, solving for one requirement should not take longer than solving for another. In summary, the 'preferred' assumption is supported by the following two criteria: (1) Linearity, i.e. the sequence of considering and solving the requirements should follow a step-by-step manner as per the description

Table 1 Final design solutions created by the subjects

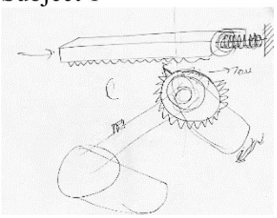
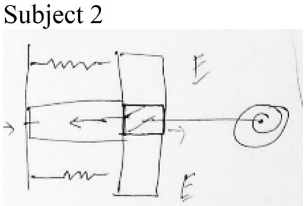
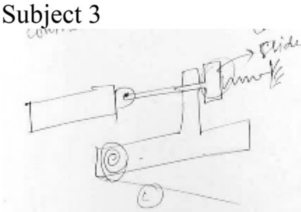
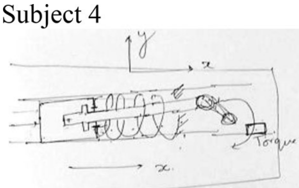
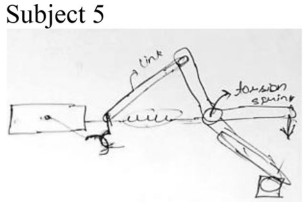
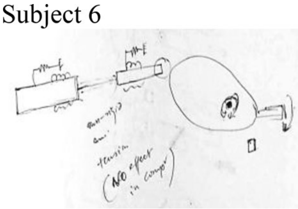
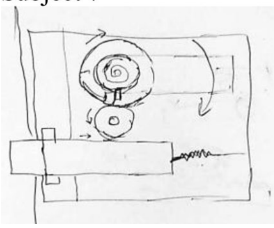
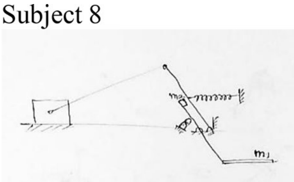
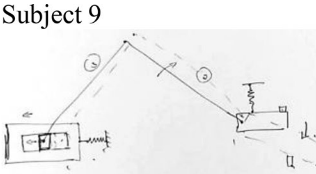
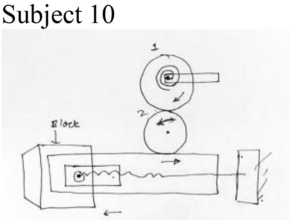
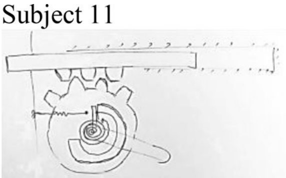
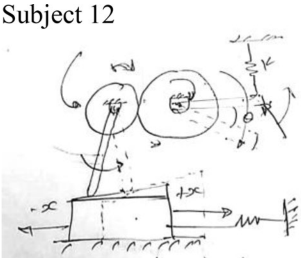
Final design solution	*Req. 1*	*Req. 2*	*Req. 3*	*Req. 4*	*Req. 5*
Subject 1 	Rack and pinion	Restricted rotation of the handle	Linear spring and torsional spring	Gear tooths in a certain portion of the pinion have been removed	Linear spring
Subject 2 	Scotch Yoke mechanism	Restricted movement of the sliding yoke	Torsional spring	The slot in the sliding yoke	Linear spring
Subject 3 	Inverted 'T' shaped handle, block and connecting lever	Restricted rotation of the handle	Torsional spring	The handle and connecting lever are connected with a 'pin in slot' joint	Linear Spring
Subject 4 	Slider-crank	Restricted rotation of the crank	Linear spring	The connecting rod and slider are connected with a 'pin in slot' joint	Linear spring
Subject 5 	Slider-crank	Restricted rotation of the handle	Linear spring	The connecting rod and slider are connected with hooks	Linear spring
Subject 6 	Cam and follower	Restricted rotation of the handle	Torsional spring	The follower and block are connected with a string	Linear spring

Table 1 (continued)

Final design solution	*Req. 1*	*Req. 2*	*Req. 3*	*Req. 4*	*Req. 5*
Subject 7 	Rack, pinion, and idler gear	Restricted movement of the rack	Torsional spring and Linear spring	Slot in the pinion	Linear spring
Subject 8 	Slider-crank	Restricted rotation of the crank	Linear spring	Double crank, one connected to the handle, another with slider and spring	Deadweight mass on the crank, which restricts the rotation of the crank
Subject 9 	Slider-crank	Restricted rotation of the handle	Linear spring	Slot in the slider	Linear spring
Subject 10 	Rack, pinion, and idler gear	Restricted movement of the rack	Torsional spring and Linear spring	Slot in the rack	Linear spring
Subject 11 	Rack and pinion	Restricted movement of the rack	Linear spring	Slot in the pinion	Torsional Spring
Subject 12 	Wedge cam-follower and a gear pair	Restricted rotation of the handle	Linear spring	Not required	Linear spring

of the MSDT provided to the subjects; and (2) Uniformity, i.e. the subjects should spend a uniform amount of time while solving for each of the five requirements. This 'preferred' assumption is similar to the algorithmic view of mechanical design discussed by Stauffer and Ullman (1988), where a specific sequence of steps needs to be followed to solve design problems. However, in actual scenarios, many subjects deviate from this 'preferred' or algorithmic assumption and follow a dynamic or iterative approach while solving the given requirements. As described by Radcliffe and Lee (1989), "Efficient designers are assumed to adopt a systematic approach and to follow a logical sequence of design processes". Thus, to develop a measure for calculating the synthesis process

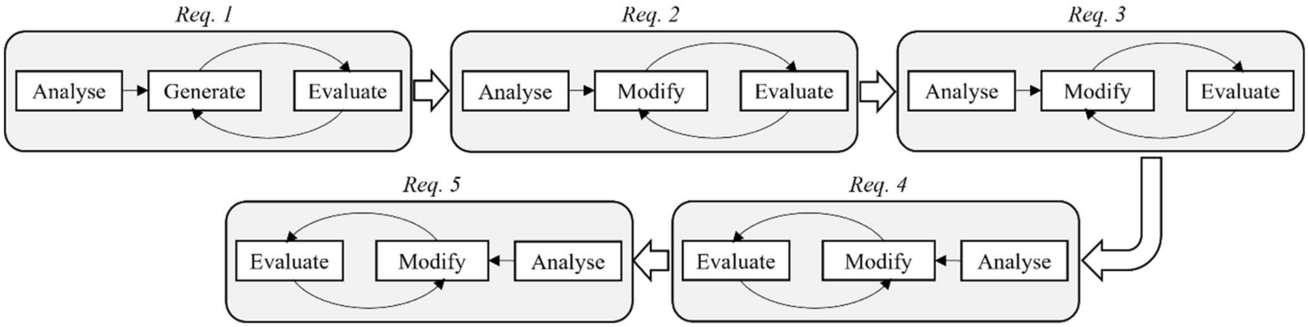


Fig. 6 The 'preferred' synthesis approach

efficiency, it has been assumed that the actual synthesis approaches that were closer to the 'preferred' (can also be considered systematic or algorithmic) synthesis approach were more efficient and should lead to a feasible design solution in less time.

To calculate the synthesis process efficiency, the actual time-requirement sequence followed by each subject was plotted and the extent to which the actual time-requirement sequence deviated (measured in terms of residual variance) from the 'preferred' sequence was taken as an indication of the synthesis process efficiency of each subject. Once the residual variances are calculated for all the subjects, the following research questions are investigated:

- *RQ1*: Did all the subjects spend a uniform percentage of total time on each requirement as per the 'preferred' assumption? If not, then on which requirement they have spent the maximum percentage of total time?
- *RQ2*: If the subjects' actual synthesis process deviates from the 'preferred' assumption, does it affect the design task completion time? (In other words, is there any correlation exist between the synthesis process efficiency and the design task completion time?)
- *RQ3*: The third question focuses on the *activities* associated with the requirements. The 'preferred' approach says that the subjects are expected to *generate* an initial solution for *Req. 1* and *modify* the same to satisfy the other requirements. In case a subject's *activity* sequence defers from the 'preferred' one, does it affect the synthesis process efficiency?

6 Results and discussion

6.1 Use of time

A considerable amount of time was spent on the given design task by all the subjects to arrive at a final feasible solution that satisfies all the EFs. The maximum and minimum amount of time spent by the subjects were 12 min and 45 min, respectively. The percentage distribution of total time spent among the requirements for all 12 subjects are shown in Fig. 7. As compared to the 'preferred' synthesis process assumption, none of the subjects spent a uniform amount of time across all the requirements. The

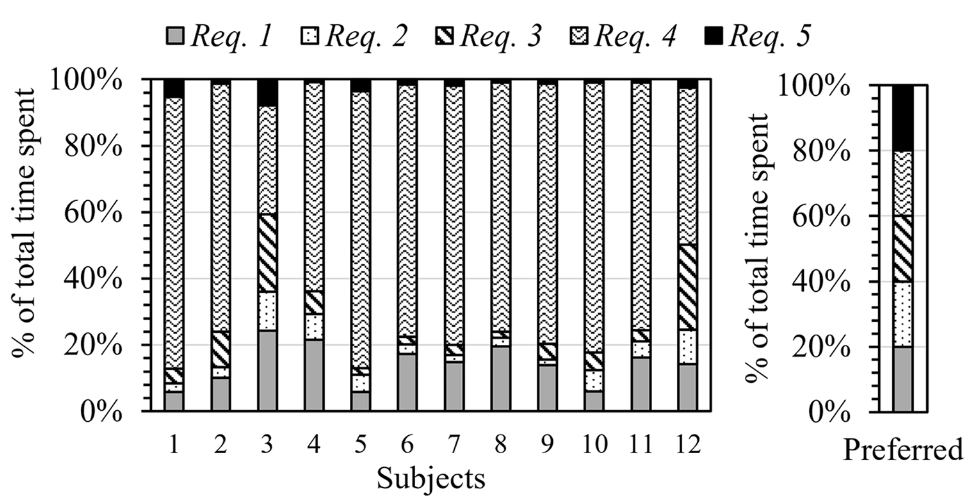


Fig. 7 The percentage distribution of total time spent among the requirements

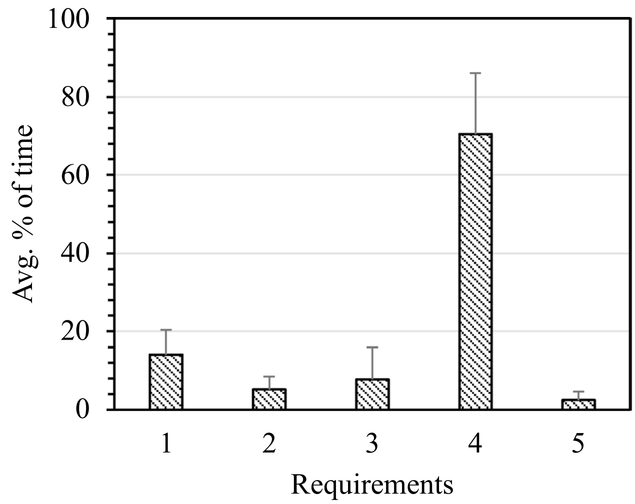


Fig. 8 The average percentage of total time spent on the requirements

average percentage of total time spent on *Req. 4* was found to be 70.5% which was much higher compared to that on any other requirement (Fig. 8). Thus, it can be concluded that *Req. 4* was the most difficult requirement, possibly because it demanded prior knowledge of metamorphic/variable constraint kinematic joints, which was lacking among most of the subjects. In contrast, the average percentage of total time spent on *Req. 5* was 2.5%, which was the least of all. It can be observed from Table 1 that while satisfying *Req. 3* and *Req. 5* the subjects had to modify the current solution such that it satisfied f_3 and f_5, respectively, where both were *Type-3* EFs. In order to satisfy *Req. 3*, the modifications performed by all the subjects consisted of adding a linear or torsional spring to their current solutions. Since, the nature of *Req. 5* was similar to that of *Req. 3*, in most cases, the prior modifications done by the subjects, while considering *Req. 3* helped them realise the *Req. 5* and thus it took the least amount of time compared to other requirements.

Overall, all the subjects were found to be proficient in modifying their proposed initial solution for *Req. 2*, *Req. 3*, and *Req. 5* compared to *Req. 1* and *Req. 4*. In these cases, the required modifications involve adding a spring or a stopper to the handle or block. According to the assumption of 'uniformity', if the subjects had the necessary knowledge to perform a modification, then solving one requirement should not take longer than solving another. A one-way ANOVA is performed to test the null hypothesis that there are no differences in the mean percentage of total time spent on *Req. 2*, *Req. 3*, and *Req. 5*. The test reveals that there is no statistically significant difference at the $p < 0.05$ level in the mean percentage of total time spent on *Req. 2*, *Req. 3*, and *Req. 5* $[F(2, 33) = 3.06; p = 0.06]$ and thus the notion of 'uniformity' is also justified.

6.2 Actual sequence of requirement consideration

Table 2 shows the sequence of requirement considerations for each subject without reference to the time spent on each requirement. It can be noted that except for Subject 3, Subject 5, and Subject 8, none of the other 9 subjects progressed through the requirements in the 'preferred' sequence. After analysing the videos, it was observed that although the EFs were described step by step in the design task document (Fig. 1), some of the subjects ignored intermediate requirements. As an example, after finding solutions for *Req. 1*, Subject 6, Subject 7, Subject 10, and Subject 11 jumped to *Req. 3* and came back to *Req. 2* later. When a subject failed to satisfy a particular requirement, they often decided to skip that requirement for the time being and progressed with the subsequent requirement and returned to that unsatisfied requirement in later stages. In some cases, after the subjects failed to satisfy a particular requirement, instead of trying to modify the current partial solution, they decided to discard the solution and started the synthesis process all over again with *Req. 1*.

Table 2 Requirement sequence of each subject

	Subject number											
	1	2	3	4	5	6	7	8	9	10	11	12
Sequence of requirements considered by each subject→	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	3	3	2	2	3	3	2
	3	3	3	3	3	2	4	3	3	2	4	3
	4	4	4	4	4	4	1	4	4	3	2	4
	5	3	5	1	5	1	2	5	5	2	4	5
	4	4		4		2	4		4	3	5	4
	5	5		1		3	5			4		2
	4			2		4				3		3
				3		5				4		4
				4						5		5
				5						4		
										5		

The above discussion also supports the observations made by Stauffer and Ullman (1988) that the designer's attention usually shifts towards critical parts of the given problem, i.e. designers become 'opportunistic' rather than 'systematic'. This 'opportunistic' nature of the design synthesis process results in deviation from the preferred 'linearity' and 'uniformity' assumptions. In regard to the amount of time spent on the task, it has been observed that, the average task completion time for Subject 3, Subject 5, and Subject 8 was 22 min, whereas for the rest of the subjects, the average task completion time was 34 min. This implies that, while designers' design synthesis approaches are generally dynamic and 'opportunistic,' in the context of MSDS, the systematic 'linear' approach may be useful for developing a design synthesis model for supporting the designers. However, according to the assumptions regarding synthesis process efficiency discussed in Sect. 5.6, 'linearity' is not the only factor determining efficiency; 'uniformity' should also be present to support that the subjects did not struggle while solving a particular requirement compared to another due to a lack of knowledge or individual competency. For example, Subject 5 and Subject 8 considered the requirements linearly, but the time spent across all the requirements was not uniform compared to Subject 3. Subject 5 and Subject 8 spent 83% and 75% of total time on *Req. 4* whereas Subject 3 spent 33% of total time on *Req. 4*.

To calculate the synthesis process efficiency of each subject, the sequence of considering requirements is plotted as a function of elapsed time. Two contrasting scenarios can be observed from the time-requirement sequence plots of Subject 3 (Fig. 9) and Subject 1 (Fig. 10). Subject 3 nearly resembles the 'preferred' time-requirement sequence with a residual variance of 0.263 (the lowest among all) and therefore can be considered an efficient synthesis process where both the assumptions of 'linearity' and 'uniformity' were nearly satisfied. In contrast, Subject 1 substantially deviated from the 'preferred' time-requirement sequence, resulting in a comparatively higher value of residual variance of 2.343.

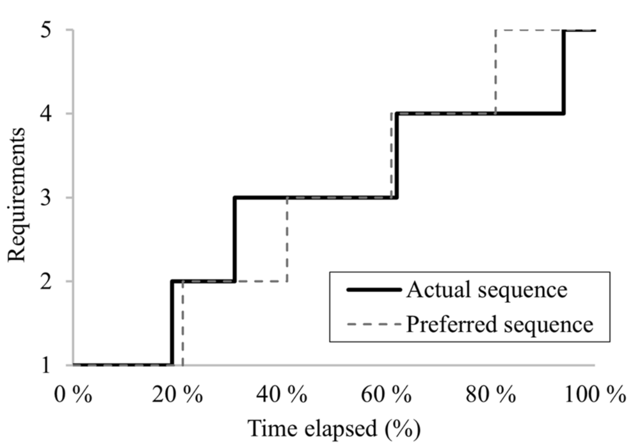


Fig. 9 The time-requirement sequence of Subject 3 (residual variance = 0.263)

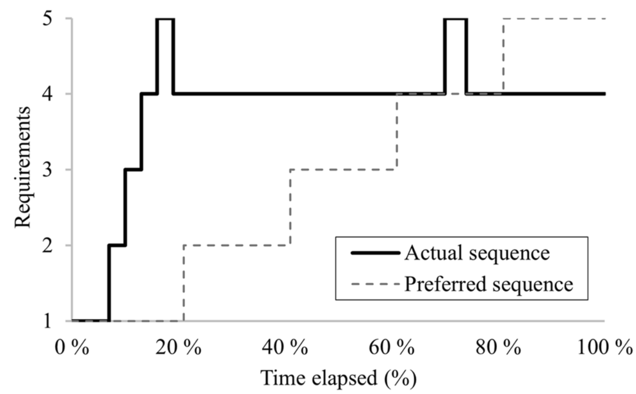


Fig. 10 The time-requirement sequence of Subject 1 (residual variance = 2.343)

Reasoning for the method of calculating synthesis process efficiency: In Sect. 5.5, a 'preferred' synthesis approach was discussed based on the observed synthesis strategy carried out by the subjects. As per the given definition of synthesis process efficiency in Sect. 5.6, those who closely followed the 'preferred' *time-requirement* sequence, resulted in lower residual variances than those who deviated more from the 'preferred' *time-requirement* sequence. In order to evaluate the 'preferred' assumption, it can be hypothesised that the subjects who were more efficient in their synthesis process, should take less time to complete the given design task. Thus, a statistical analysis is performed to test the correlation between the subjects' residual variances and task completion time. In Fig. 11, the task completion time values are plotted on the left *Y*-axis and the residual variances on the right *Y*-axis. The Kolmogorov–Smirnov test is performed to check for normality; it was found that the data did not differ

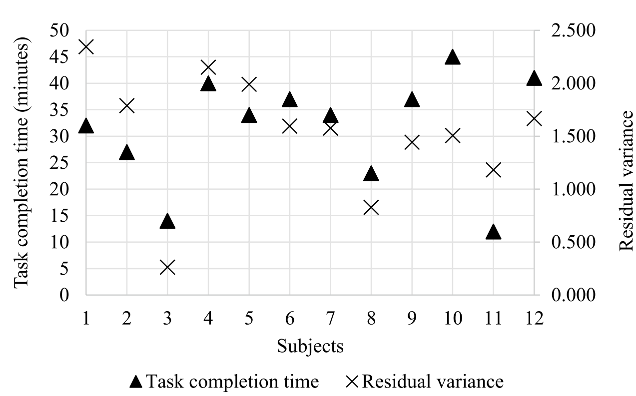


Fig. 11 Task completion time and residual variances of all the subjects

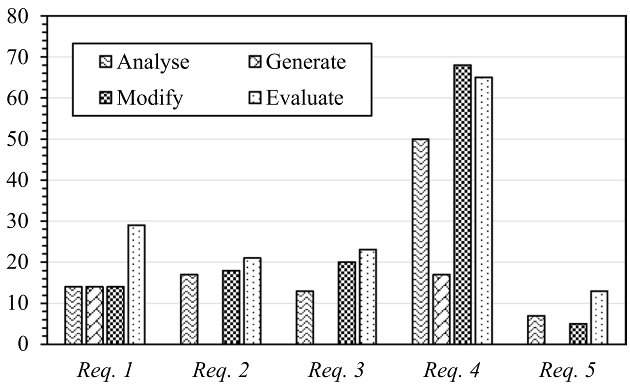


Fig. 12 The number of occurrences of activities corresponding to the requirements

Table 3 The synthesis process efficiency of all the subjects

Group A		Group B	
Sub. No.	Efficiency (res. var.)	Sub. No.	Efficiency (res. var.)
1	2.343	3	0.263
2	1.788	7	1.576
4	2.152	8	0.828
5	1.990	9	1.444
6	1.596	11	1.182
10	1.505		
12	1.677		
Mean	1.864	Mean	1.059
SD	0.308	SD	0.529

significantly (at $p > 0.7$) from that which is normally distributed. Further, a Pearson Correlation test (https://www.socscistatistics.com/tests/pearson/) shows that there is a moderate positive correlation ($r = .635; p = 0.026$) between residual variance and task completion time, that is, when the subject's synthesis process was closer to the 'preferred' time-requirement sequence, the subject was more likely to arrive at a feasible solution in less time.

6.3 Use of activities

Based on the coding scheme, various activities during the synthesis were identified. The number of occurrences of activities corresponding to the five requirements is shown in Fig. 12. The cumulative number of occurrences of activities in descending order are: *Evaluate*, *Modify*, *Analyse* and *Generate*. Although it was assumed in the 'preferred' process that a *generate* activity should only occur while considering *Req. 1* to create an initial solution, some of the subjects tried to *generate* new solutions to satisfy *Req.4* instead of *modifying* the existing solutions. In order to understand whether such strategy influences the efficiency of the synthesis process, the encoded dataset was divided into two groups: Group A and Group B. Group A consists of those subjects where at least one *Generate* activity corresponding to *Req. 4* was observed; whereas Group B consists of those subjects where no *Generate* activity corresponding to *Req. 4* was observed. In other words. the subjects of Group B preferred to progress with the semi-working initial solution and *modified* it further while considering *Req. 4*. Whereas the subjects of Group A preferred to generate a different primitive structure that satisfied f_4 and tried to combine it with the existing solution generated for f_1. For example, Subject 1 from Group A *generated* a rack-and-pinion as an initial solution, which satisfied f_1. While considering *Req. 4*, Subject 1 generated a ratchet and tried to combine it with the existing rack-and-pinion. After evaluating the combined solution, the subject realised that the proposed solution did not satisfy *Req. 4*. Then, the subject discarded the ratchet and tried to modify the gear tooth profile of the pinion (see Table 1 for reference) to satisfy *Req. 4*.

The synthesis process efficiencies calculated in terms of residual variances of all the subjects are listed in Table 3. A Mann–Whitney *U* test was performed to determine whether there are significant differences between the mean residual variances of Group A and Group B. The Mann–Whitney *U* test is a nonparametric test that allows two independent groups to be compared without assuming that values are normally distributed. The observed value of Mann–Whitney test-statistics (U_{obs}) was 1. The critical value of U at $p < 0.01$ for the combination of sample sizes ($n_1 = 7$ and $n_2 = 5$) was 3 for a one-tailed test. The result was significant at $p < 0.01$ and it was concluded that the mean residual variances of Group A was greater than Group B.

6.4 Discussion

This section summarises various aspects of the results and responds to the research questions. In this empirical study, first, the given MSDT was analysed using EFs and specifications graph, and the types of EFs associated with this task were identified. Then, a coding scheme was developed to code the video protocol data in terms of the *requirements* under consideration and the associated *activities*. There were mainly two kinds of *requirements*: identifying a solution for a particular EF (considered as a *given requirement*) and identifying a solution that resolves the conflict between two or more EFs (considered as a *solution-specific requirement*). Further, the subjects' synthesis approaches were analysed, and a common understanding was developed and described through a 'preferred' synthesis strategy that follows a logical sequence of *requirements* and corresponding *activities*. At this juncture, the 'preferred' approach was a hypothesised approach and to understand how the subjects' actual synthesis approaches deviated from the 'preferred' one, a measure of 'synthesis process efficiency' was introduced. The *RQ1* and *RQ3* (discussed in Sect. 5.6) primarily focus on what caused the subjects to deviate from the 'preferred' approach. Whereas *RQ2* investigates whether any correlation exists between the 'synthesis process efficiency' and the design task completion time. The results are summarised as follows:

- From Table 1, it can be observed that each designer, after analysing the given MSDT, selected a *Type-1* EF (f_1), and generated an initial solution proposal that can fully or partially satisfy this EF. This gives an implication that a *Type-1* EF can provide a starting point to approach this kind of synthesis task.
- In response to *RQ1*, Fig. 8 shows that the average percentage of total time spent on *Req. 4 was* found to be much higher compared to any other requirement. *Req. 4* pertained to resolving the conflict between two *Type-1* EFs (f_1 and f_4). It demanded prior knowledge of metamorphic/ variable constraint joints which was lacking among most of the subjects.
- In response to *RQ2*, the results from Sect. 6.2 indicates that when the subjects' synthesis approach deviated less from the 'preferred' synthesis approach, the subjects had spent less time to arrive at a feasible solution.
- In response to *RQ3*, the results from Sect. 6.3 shows that the subjects who preferred to progress with the semi-working initial solution (*generated* for *Req. 1*), deviated less from the 'preferred' approach compared to those who tried to *generate* new solutions for different *requirements*.

Overall, the empirical findings indicate that the 'preferred' synthesis strategy can be regarded as one of the potential models of the MSMD design synthesis process. It can assist in solving MSDS problems in a systematic and time-efficient manner. In the following section, the knowledge extracted from the 'preferred' synthesis strategy is presented in more detail as a descriptive model. The model intends to provide information about an ideal flow of work (or sequence of activities) which can be followed during the MSMD design synthesis process. The model supports 'systematic' or 'algorithmic' approach where, the requirements can be considered in a 'linear' step-by-step manner. This can also serve as a basis for future computer implementation where a design support tool can be developed to help designers explore a larger solution space by following the proposed model. Of course, the proposed model may not be effective and time-efficient in case of every MSDT. The 'opportunistic' nature of design, intuitive thoughts, and individual techniques of the designers may also lead to success. However, the method can be a useful addition to designer's repertoire of design synthesis methods.

7 A descriptive model of the MSMD design synthesis process

Descriptive models are representations of strategies proposed to show how design is carried out or what is involved in designing (Evbuomwan et al. 1996). These models are mainly concerned with designers' actions and activities during the design process. The model provides a structured approach to design synthesis, helping designers to move from the MSDT to the final design solution in a systematic way. By following the model, designers can ensure that they generate feasible design solutions quickly and effectively that meet the design requirements. In this section, a generic descriptive model of the synthesis process for a MSDT have been discussed.

7.1 The descriptive model

The descriptive model, given in the form of a flow chart of activities, input to these activities and output from these activities, is shown in Fig. 13. The process involves systematically drawing and modifying a solution concept until the solution meets all the EFs. Each step of the descriptive model is explained below:

Step 1: Develop the specifications graph (from the given MSDT).
Step 2: Analyse all the EFs in the specifications graph to identify the *types* of EFs.
Step 3: Select one EF (which has initial configuration C_i and final configuration, C_{i+1}) of $Type-1$.
Step 4: Generate an initial solution proposal for the selected $Type-1$ EF.
Step 5: Modify the proposal for the selected EF if it does not completely satisfy the EF.
Step 6: Select the next EF, which starts at configuration C_{i+1}, as specified by the path (i.e. arcs representing the EFs) in the specifications graph.
Step 7: Modify the solution proposal if it does not satisfy the current EF, till the current EF is completely satisfied.
Step 8: Repeat Step 6 followed by Step 7 with other remaining EFs (if any).

To produce multiple feasible alternative design solutions, designers can generate a different initial solution at Step 4

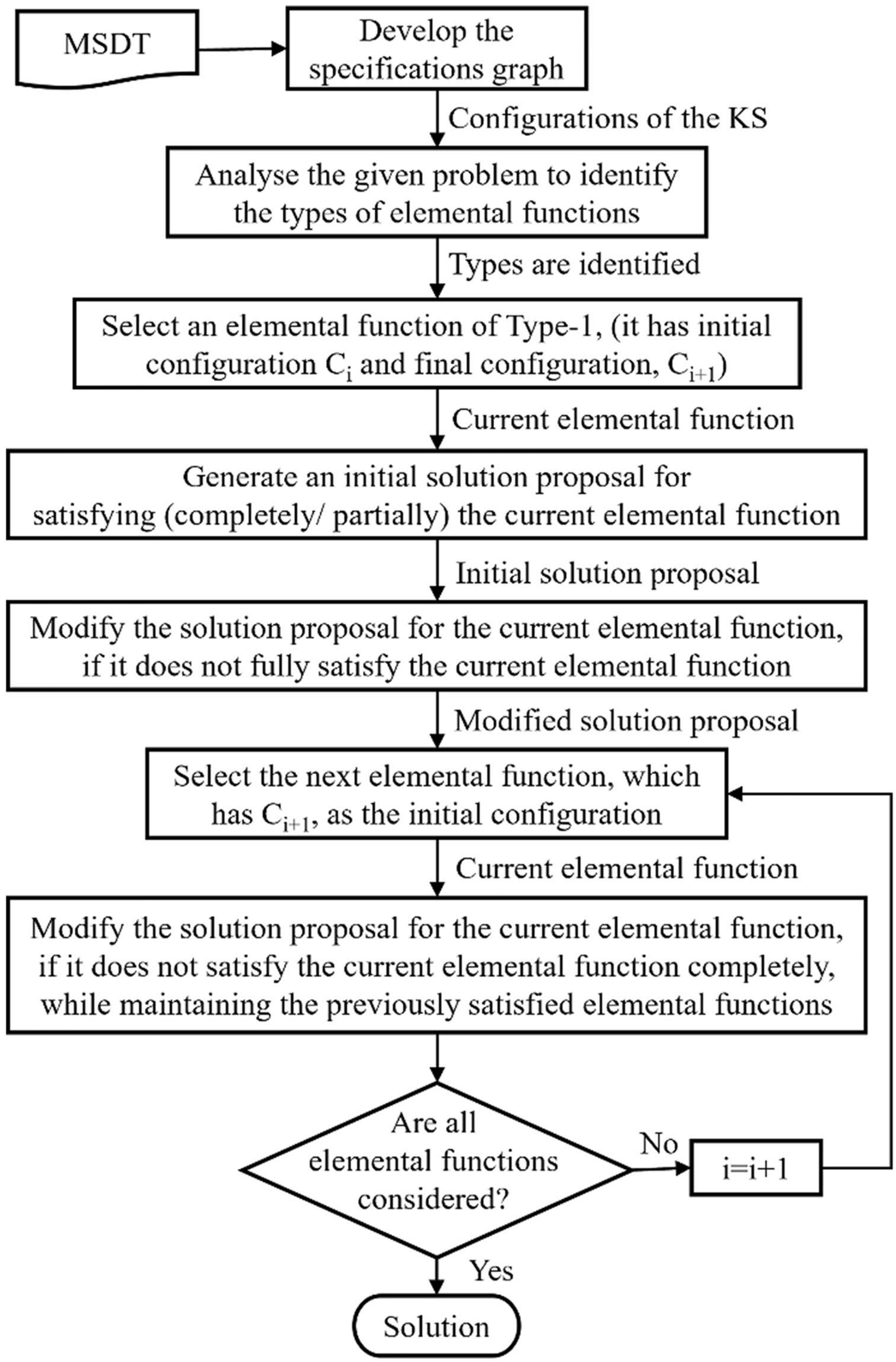


Fig. 13 Flowchart of the descriptive model

and then proceed with the remaining steps mentioned above to modify the initial solution till it meets all the EFs.

The descriptive model promotes 'systematic' nature of the design synthesis process addressed in Sect. 6.2. However, the model doesn't ensure 'uniformity' as discussed in Sect. 5.6. As observed in the empirical studies, some of the required modifications may require domain knowledge and individual competency. The subjects spent the maximum amount of time solving mismatches between two Type-1 EFs. This kind of modification involves metamorphic kinematic joints, which is an essential feature of any MSMD. In general, a MSMD needs to achieve different EFs under different operating states through its variable topological characteristics. Therefore, in addition to the descriptive model, appropriate prescriptive knowledge needs to be provided to the designers to ensure 'uniformity'. The prescriptive knowledge should support designers in modifying the semi-working solution to eliminate a mismatch between the required EF and the existing EF exhibited by the semi-working solution at a particular configuration state of the given MSDT. Different kinds of modifications observed in the empirical studies (see Table 1) can be used to develop prescriptive knowledge. Structural characteristics of different existing MSMDs, such as circuit breakers, electrical switches, bicycle gear shifters, automotive transmissions, etc., can also be used to develop a design case library, and appropriate knowledge regarding modifications can be abstracted from these design cases. However, when it comes to generating an initial solution for a particular Type-1 EF (at Step 4 of the descriptive model), the existing process-based and case-based SSDS methods discussed in Sect. 2.1 can be useful.

7.2 Comparison with other existing models on MSMD synthesis

In the context of MSMD synthesis, Zhang et al. (2011) employed morphological matrix as a method of mechanism synthesis to *generate* the sub-mechanisms corresponding to the subfunctions. In their case, the subfunctions were similar to the EFs, which transform the system from one configuration to another. An appropriate mechanism was selected for each sub-function from a library of conventional mechanisms realising different motion transformations. Once the morphological chart was created, a final working solution was achieved by combining the sub-mechanisms. Even if this approach technically gives a solution to a MSDT, the solution obtained as a combination of sub-mechanisms may become cumbersome and not worthy of consideration for downstream activities like embodiment or detailed design. If a morphological chart contains a semi-working or partially working solution, it leads to rejection of the concept. A similar approach can also be observed in the design synthesis process proposed by Li et al. (1999a, b), where the computational tool called ADCS uses a recursive algorithm to generate a design tree which contains the design specifications (i.e. the EFs). Then, it retrieves primitive structures from the database according to the design specifications in the design tree nodes. Finally, the design solution is generated as a network of primitive structures obtained by traversing the design tree. While generating the design solution, ADCS does not consider modification of the retrieved primitive structures apart from combining a new primitive structure to the current semi-working solution proposal to eliminate the mismatches between existing and required EFs. In contrast, the proposed descriptive model intends to start with a semi-working initial solution and encourage the designer to modify it until it becomes a fully working solution. It is important to note that the proposed descriptive model is analogous to the 'Backwards Design Method' proposed by Burgess (2012). This method also involves starting with an idealistic semi-working solution and then systematically solving the unworkable parts of the solution until a complete, feasible solution is found. However, the 'Backwards Design Method' is not specifically developed for an MSDT and does not provide any direction to produce the initial semi-working solution concept.

8 Conclusions

It has been found from the literature that existing methods for conceptual design synthesis of both single-state mechanical devices and multiple-state mechanical devices predominantly employ the method of composition of building blocks, where a solution is a network of building blocks. As the literature on multiple-state design synthesis is sparse, empirical studies are undertaken in the research reported in this paper, to (1) understand how and how well designers currently synthesise solutions to multiple-state design tasks; and (2) to comprehend the knowledge involved in carrying out the multiple-state design synthesis processes. It has been observed from these empirical studies that the common practice followed by all the designers, during the synthesis process for multiple-state design tasks, is to select one of the *Type-1* EFs for which a fully or partially satisfying solution proposal is *generated*; and this proposal is kept on being *modified* until all the EFs in the design task are satisfied. From the video recordings, data were encoded in terms of activities and requirements. With help of that, it was possible to acquire a better understanding of a subject's synthesis strategy. The actual synthesis strategies taken by the subjects were compared with a 'preferred' synthesis strategy that follows a logical sequence of requirements and corresponding activities. The findings indicated that when the subject's synthesis process was closer to the 'preferred' assumption, the subject was more likely to arrive at a feasible solution in less time. The subjects spent the maximum amount of time resolving mismatches between two Type-1 EFs. This kind of modification involves metamorphic kinematic joints, which is an essential feature of any MSMD. Most of the subjects lacked prior knowledge of metamorphic/ variable constraint kinematic joints, which was likely to have been the main reason as to why the 'uniformity' assumption was not met. Based on the empirical studies, a descriptive model of the synthesis process for MSDT is developed. In the future, the descriptive model will be used in addition to appropriate prescriptive support to help designers generate a large solution space for a given MSDT.

Acknowledgements The authors wish to express their gratitude to all the students for voluntarily taking part in the experiments.

Author contributions AM (first author) carried out the work and prepared the manuscript under the supervision of Prof. AC. Dr SRT (second author) helped conduct the experiments and assisted in manuscript drafting.

Data availability Not applicable for our work.

Declarations

Conflict of interest The authors declare no competing interests.

References

Burgess SC (2012) A backwards design method for mechanical conceptual design. J Mech Des. https://doi.org/10.1115/1.4005620

Chakrabarti A, Bligh TP (1994) An approach to functional synthesis of solutions in mechanical conceptual design. Part I: Introduction and knowledge representation. Res Eng Des 6(3):127–141. https://doi.org/10.1007/bf01607275

Chakrabarti A, Bligh TP (1996a) An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues. Artif Intell Eng Des Anal Manuf 10(4):313–331. https://doi.org/10.1017/s0890060400001645

Chakrabarti A, Bligh TP (1996b) An approach to functional synthesis of solutions in mechanical conceptual design. Part II: kind synthesis. Res Eng Des 8(1):52–62. https://doi.org/10.1007/bf01616556

Chakrabarti A, Morgenstern S, Knaab H (2004) Identification and application of requirements and their impact on the design process: a protocol study. Res Eng Des 15(1):22–39

Chen Y et al (2005) Automated conceptual design of mechanisms using improved morphological matrix. J Mech Des 128(3):516–526. https://doi.org/10.1115/1.2180807

Chiou S-J, Kota S (1999) Automated conceptual design of mechanisms. Mech Mach Theory 34(3):467–495. https://doi.org/10.1016/s0094-114x(98)00037-8

Ding H et al (2012) Complete atlas database of 2-DOF kinematic chains and creative design of mechanisms. J Mech Des. https://doi.org/10.1115/1.4005866

Domeshek EA et al. (1994) A case-based design aid for conceptual design of aircraft subsystems. In: Proceedings of the Tenth Conference on Artificial Intelligence for Applications, pp 63–69. https://doi.org/10.1109/CAIA.1994.323691

Evbuomwan NFO, Sivaloganathan S, Jebb A (1996) A survey of design philosophies, models, methods and systems. Proc Inst Mech Eng, Part B: J Eng Manuf 210(4):301–320. https://doi.org/10.1243/PIME_PROC_1996_210_123_02

Finger S, Dixon JR (1989) A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes. Res Eng Des 1(1):51–67

Finger S, Rinderle JR (2002) Transforming behavioural and physical representations of mechanical designs. Engineering design synthesis. Springer, London, pp 303–317. https://doi.org/10.1007/978-1-4471-3717-7_18

Fricke G (1996) Successful individual approaches in engineering design. Res Eng Des 8(3):151–165. https://doi.org/10.1007/bf01608350

Gero J, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19(1):21–61

Han Y-H, Lee K (2006) A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms. Comput Ind 57(4):305–318. https://doi.org/10.1016/j.compind.2005.09.005

Han L et al (2020) A computational synthesis approach of mechanical conceptual design based on graph theory and polynomial operation. Chin J Mech Eng. https://doi.org/10.1186/s10033-019-0424-9

He B, Zhang P, Liu L (2014) Simultaneous functional synthesis of mechanisms with mechanical efficiency and cost. Int J Adv Manuf Technol 75(5–8):659–665. https://doi.org/10.1007/s00170-014-6167-y

Hoover SP, Rinderle JR (1989) A synthesis strategy for mechanical devices. Res Eng Des 1(2):87–103. https://doi.org/10.1007/bf01580203

Jagtap S et al (2014) How design process for the base of the pyramid differs from that for the top of the pyramid. Des Stud 35(5):527–558

Jimenez JM et al (1997) A simple and general method for kinematic synthesis of spatial mechanisms. Mech Mach Theory 32(3):323–341. https://doi.org/10.1016/s0094-114x(96)00017-1

Johnson RC (1978) Mechanical design synthesis: creative design and optimization. Krieger Pub Co. ISBN: 978-0882756127

Karnopp D, Margolis D (1979) Analysis and simulation of planar mechanism systems using bond graphs. ASME J Mech Des 101(2):187–191. https://doi.org/10.1115/1.3454038

Kota S, Chiou S-J (1992) Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. Res Eng Des 4(2):75–87. https://doi.org/10.1007/bf01580146

Li CL (1998) Conceptual design of single and multiple-state mechanical devices: an intelligent CAD approach. Thesis, University of Hong Kong, Pokfulam, Hong Kong SAR. Retrieved from https://doi.org/10.5353/th_b3123733

Li Y-T, Wang Y-X (2019) A bottom–up function-based synthesis approach for complex mechanisms based on clustering rules. Mech Based Des Struct Mach 47(5):555–567. https://doi.org/10.1080/15397734.2019.1578666

Li CL, Tan ST, Chan KW (1996) A qualitative and heuristic approach to the conceptual design of mechanisms. Eng Appl Artif Intell 9(1):17–32. https://doi.org/10.1016/0952-1976(95)00060-7

Li CL, Chan KW, Tan ST (1999a) A configuration space approach to the automatic design of multiple-state mechanical devices. Comput-Aided Des 31(10):621–653. https://doi.org/10.1016/s0010-4485(99)00058-5

Li CL, Chan KW, Tan ST (1999b) Automatic design by configuration space: an automatic design system for kinematic devices. Eng Appl Artif Intell 12(5):613–628. https://doi.org/10.1016/s0952-1976(99)00029-9

Liu YC, Chakrabarti A, Bligh T (2000) A computational framework for concept generation and exploration in mechanical design: further developments of FuncSION. Artif Intell Des. https://doi.org/10.1007/978-94-011-4154-3_25

Liu Y-C, Chakrabarti A, Bligh T (2003) Towards an 'ideal' approach for concept generation. Des Stud 24(4):341–355. https://doi.org/10.1016/s0142-694x(03)00003-6

Liu C et al (2015) Conceptual design of multi-modal products. Res Eng Des 26(3):219–234. https://doi.org/10.1007/s00163-015-0193-0

Moon Y-M, Kota S (2002) Automated synthesis of mechanisms using dual-vector algebra. Mech Mach Theory 37(2):143–166. https://doi.org/10.1016/s0094-114x(01)00073-8

Mulet E, Vidal R (2008) Heuristic guidelines to support conceptual design. Res Eng Design 19(2–3):101

Murakami T, Nakajima N (1997) Mechanism concept retrieval using configuration space. Res Eng Des 9(2):99–111. https://doi.org/10.1007/bf01596485

Navinchandra D, Sycara KP, Narasimhan S (1991) A transformational approach to case-based synthesis. Artif Intell Eng Des, Anal Manuf 5(1):31–45. https://doi.org/10.1017/s0890060400002523
Nidamarthi S (1999) Understanding and supporting requirement satisfaction in the design process. Doctoral dissertation, University of Cambridge
Olson DG, Erdman AG, Riley DR (1985) A systematic procedure for type synthesis of mechanisms with literature review. Mech Mach Theory 20(4):285–295
Pahl G, Beitz W (1996) Engineering design: a systematic approach. Springer Verlag, London
Pons DJ, Raine JK (2005) Design mechanisms and constraints. Res Eng Des 16(1–2):73–85. https://doi.org/10.1007/s00163-005-0008-9
Pozhbelko V (2019) Type synthesis method of planar and spherical mechanisms using the universal structural table with all possible link assortments. Advances in mechanism and machine science. Springer International Publishing, pp 1517–1526. https://doi.org/10.1007/978-3-030-20131-9_150
Prabhakar S, Goel AK (1998) Functional modeling for enabling adaptive design of devices for new environments. Artif Intell Eng 12(4):417–444. https://doi.org/10.1016/s0954-1810(98)00003-x
Pucheta MA et al (2012) Topological synthesis of planar metamorphic mechanisms for low-voltage circuit breakers. Mech Based Des Struct Mach 40(4):453–468. https://doi.org/10.1080/15397734.2012.687296
Radcliffe DF, Lee TY (1989) Design methods used by undergraduate engineering students. Des Stud 10(4):199–207
Sacks E, Joskowicz L (2010) The configuration space method for kinematic design of mechanisms (The MIT Press). The MIT Press. ISBN: 9780262013895. URL: https://mitpress.mit.edu/books/configurationspace-method-kinematic-design-mechanisms
Sarkar P, Chakrabarti A (2011) Assessing design creativity. Des Stud 32(4):348–383
Sommerville I, Sawyer P (1997) RE: a good practice guide. Wiley, pp 113–114
Srinivasan V, Chakrabarti A (2010a) An integrated model of designing. J Comput Inform Sci Eng 10(1115/1):3467011
Srinivasan V, Chakrabarti A (2010b) Investigating novelty–outcome relationships in engineering design. AI EDAM 24(2):161–178
Starling AC, Shea K (2005) A parallel grammar for simulation-driven mechanical design synthesis. In: Volume 2: 31st Design Automation Conference, Parts A and B. ASMEDC. https://doi.org/10.1115/detc2005-85414
Stauffer LA, Ullman DG (1988) A comparison of the results of empirical studies into the mechanical design process. Des Stud 9(2):107–114
Stauffer LA, Ullman DG (1991) Fundamental processes of mechanical designers based on empirical data. J Eng Des 2(2):113–125
Subramanian D, Wang C-S (1995) Kinematic synthesis with configuration spaces. Res Eng Des 7(3):193–213. https://doi.org/10.1007/bf01638099
Tian YL, Zou HJ, Guo WZ (2005) An integrated knowledge representation model for the computer-aided conceptual design of mechanisms. Int J Adv Manuf Technol 28(5–6):435–444. https://doi.org/10.1007/s00170-004-2399-6
Todeti SR, Chakrabarti A (2009) An empirical model of the process of synthesis of multiple state mechanical devices. In: DS 58-4: Proceedings of ICED 09, the 17th International Conference on Engineering Design
Tsai LW (2000) Mechanism design: enumeration of kinematic structures according to function. CRC Press
Ulrich K, Seering W (1988) Computation and conceptual design. Robot Comput-Integr Manuf 4(3–4):309–315. https://doi.org/10.1016/0736-5845(88)90002-6
Yan H-S (1998) Creative design of mechanical devices. Springer Singapore. 256 pp. ISBN: 9813083573. URL: https://www.ebook.de/de/product/5667683/hong_sen_yan_creative_design_of_mechanical_devices.html
Zhang L, Dai JS (2009) An overview of the development on reconfiguration of metamorphic mechanisms. In: 2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots, pp. 8–12
Zhang WY, Tor SB, Britton GA (2001) A prototype knowledge-based system for conceptual synthesis of the design process. Int J Adv Manuf Technol 17(8):549–557. https://doi.org/10.1007/s001700170137
Zhang WX, Ding XL, Dai JS (2011) Morphological synthesis of metamorphic mechanisms based on constraint variation. Proc Inst Mech Eng, Part c: J Mech Eng Sci 225(12):2997–3010. https://doi.org/10.1177/0954406211408953
Zhang L et al (2015) Characterization based design under dual state vectors. J Mech Robot. https://doi.org/10.1115/1.4030464
Zhang M, Sui F, Liu A, Tao F, Nee AYC (2020) Digital twin driven smart product design framework. Digital twin driven smart design. Academic Press, pp 3–32. https://doi.org/10.1016/B978-0-12-818918-4.00001-4