Res Eng Design (2014) 25:351–373
DOI 10.1007/s00163-014-0182-8

ORIGINAL PAPER

Steepest-first exploration with learning-based path evaluation: uncovering the design strategy of parameter analysis with C–K theory

Ehud Kroll · Pascal Le Masson · Benoit Weil

Received: 18 January 2014 / Revised: 16 April 2014 / Accepted: 7 July 2014 / Published online: 18 July 2014

Abstract The parameter analysis method of conceptual design is studied in this paper with the help of C–K theory. Each of the fundamental design activities—idea generation, implementation of the idea as hardware and evaluation—is explained and defined as a specific sequence of C–K operators. A case study of designing airborne decelerators is used to demonstrate the modeling of the parameter analysis process in C–K terms. The theory is used to explain how recovery from an initial fixation took place, leading to a breakthrough in the design process. It is shown that the innovative power of parameter analysis is based on C-space "de-partitioning" and that the efficient strategy exhibited by parameter analysis can be interpreted as steepest-first, controlled by an evaluation function of the design path. This logic is explained as generalization of branch-and-bound algorithms by a learning-based, dynamically evolving evaluation function and exploration of a state space that keeps changing during the actual process of designing.

Keywords Design theory · Conceptual design · C–K theory · Parameter analysis

1 Introduction

The current study focuses on using C–K theory to clarify the (implicit) theoretical grounds and logic of the pragmatic design method called parameter analysis (PA), and helps explain some notions of C–K design theory. The general logic of the paper is as follows: PA is an intriguing design method based on years of practical application, but the rationale and causes behind it still need clarification. C–K helps build a conceptual model of PA, revealing its inner workings and pointing to future directions of improvement. In this section, we justify the research methodology, provide the background on PA, C–K theory and notions of search and outline the main results.

1.1 Methodology: theory-based study of design methods

Studying a specific method with the aid of a theory is common in design research. Reich et al. (2012) analyze ASIT, a derivative of TRIZ, using the C–K design theory, and also elaborate extensively on the validity of studying design methods with theories. They argue that in order to gain deep understanding of a single method and expose in detail the reasons for its performance, a "theory-driven analysis" should be applied. They claim that such theory-based investigations of methods allow furthering our understanding of how and why the methods work, identifying their limitations, areas of applicability and possible improvements, and comparing them to other methods using a common theoretical basis. At the same time, interpreting and demonstrating the methods from the theoretical perspective can provide empirical validation of the theory. Their choice of C–K theory is further explained as follows: "The selection of the theory is rather simple as there is only

---

E. Kroll (✉)
Faculty of Aerospace Engineering, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: kroll@technion.ac.il

P. Le Masson · B. Weil
CGS - Centre de gestion scientifique (Management Science Lab), MINES ParisTech, PSL Research University, Paris, France
e-mail: pascal.lemasson@mines-paristech.fr

B. Weil
e-mail: benoit.weil@mines-paristech.fr

one candidate theory that both offers a formal modelling and embeds creativity as an integral part of design, namely the C–K theory.''

Other researchers also used C–K theory to explain various design activities, phenomena and methods. For example, Eris (2006) analyzed the pedagogical use of student portfolios with two conceptual frameworks: C–K theory and divergent–convergent inquiry-based design thinking (DCIDT). Elmquist and Segrestin (2007) applied C–K theory to study methods used at the early stages of designing in the pharmaceutical industry. Gillier et al. (2010) investigated the application of a new project portfolio management method using C–K theory. Le Masson and Weil (2013) analyzed the German systematic design methods from a historical perspective with C–K theory, and Shai et al. (2013) conducted a similar study of the Infused Design method (Shai and Reich 2004a, b).

PA is a method to design innovative products (Kroll 2013). Contrary to systematic design methods that prescribe exhaustive listing of functions and their technological solution alternatives (Tomiyama et al. 2009; Smith et al. 2012), PA dictates focusing on the most critical ''conceptual design issues'' at any given time. And although the success of this logic has been demonstrated empirically (Kroll et al. 2001), there is still no clear theoretical explanation for it. Conventional intuition leads to designing by either extensively reviewing all the pertinent issues in order to avoid late discovery of fatal errors—this is the logic of systematic design, which is robust but time consuming and not completely adapted to certain design situations (Kroll 2013), or relying on a trial and error process—which is also time consuming and risky, unless the designer is very experienced and creative (Pahl et al. 1999). In contrast, PA emerges as a method that is neither a comprehensive overview nor a random walk. Therefore, we ask: what can explain the success of PA? One could attribute it to the experience of designers using PA, but the accumulated evidence (including the one reported here) shows that PA actually helps novice, inexperienced designers to find the way in complex situations requiring some extent of creativity. So the need to investigate the rationale behind PA still remains.

Casting PA in the C–K framework will help to uncover interesting facets of PA. In particular, we show that PA extends the search strategies used to solve complex optimization problems to the domain of design. To this end, the present work also draws upon methods used in artificial intelligence (AI) and operations research (OR), especially those based on branch-and-bound (B&B) algorithms for solving search and planning problems. Brief presentations of PA and some aspects of C–K theory and notions of search that will be useful in this paper follow.

1.2 The parameter analysis design method

Parameter analysis (Kroll et al. 2001; Kroll and Koskela 2012; Kroll 2013) is an empirically derived method for doing conceptual design. It was developed initially as a descriptive model after studying designers in action and observing that their thought process involved continuously alternating between conceptual-level issues (concept space) and descriptions of hardware[1] (configuration space). The result of any design process is certainly a member of configuration space, and so are all the elements of the design artifact that appear, and sometimes also disappear, as the design process unfolds. Movement from one point to another in configuration space represents a change in the evolving design's physical description, but requires conceptual reasoning, which is done in concept space. The concept space deals with ''parameters,'' which in this context are functions, ideas and other conceptual-level issues that provide the basis for anything that happens in configuration space. Moving from concept space to configuration space involves a realization of the idea in a particular hardware representation, and moving back, from configuration to concept space, is an abstraction or generalization, because a specific hardware serves to stimulate a new conceptual thought. As will be shown later, concept space in PA is fundamentally different from C-space in C–K theory.

To facilitate the movement between the two spaces, a prescriptive model was conceived, consisting of three distinct steps, as shown in Fig. 1. The first step, parameter identification (PI), consists primarily of the recognition of the most dominant issues at any given moment during the design process. These may include the dominant physics governing a problem, a new insight into critical relationships between some characteristics, an analogy that helps shed new light on the design task or an idea indicating the next best focus of the designer's attention. Parameters play an important role in developing an understanding of the problem and pointing to potential solutions.

The second step is creative synthesis (CS). This part of the process represents the generation of a physical configuration based on the issue recognized within the PI step. Since the process is iterative, it generates many physical realizations, not all of which will be very interesting. However, the configurations allow one to see new key parameters, which will again stimulate a new direction for the process. The third component of PA, the evaluation (E) step, facilitates the process of moving away from a

---

[1] Hardware descriptions or representations are used as generic terms for the designed artifact; however, nothing in the current work excludes software, services, user experience and similar products of the design process.

physical realization back to parameters or concepts. Evaluation is important because one must consider the degree to which a specific implementation represents a possible solution to the entire problem. Evaluation also uncovers the weaknesses of the configurations and points out possible areas of improvement for the next design cycle. The unique role played by the evaluation function is elaborated later.

PA's repetitive PI–CS–E cycles are preceded by a technology identification (TI) stage of determining the most challenging functional aspect of the task, and looking into fundamental technologies and physical principles that can be used, thus establishing several starting points or initial conditions for PA. A cursory listing of each candidate technology's pros and cons follows, leading the designer to pick the one that seems most likely to succeed. While this may seem to resemble the technique of functional decomposition (or analysis) and morphology, widely used in systematic design (e.g., Pahl et al. 2007), this is not really the case here. In TI, only the most difficult aspect(s) of the overall design task are addressed, as opposed to dealing concurrently with possibly many functions and sub-functions in the morphological approach. Figure 2 is a diagram depicting the place of TI and PA as the means for carrying out conceptual design within the design process. Because the logic of TI is quite similar to what follows in PA, we sometimes refer to their combination as the PA methodology.

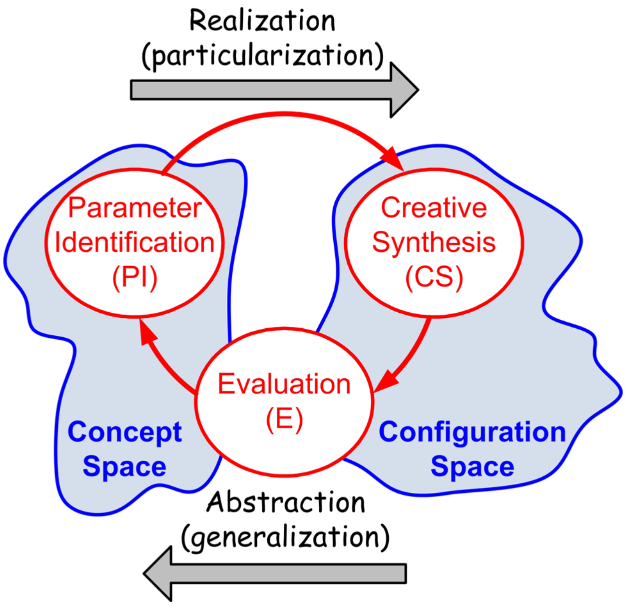


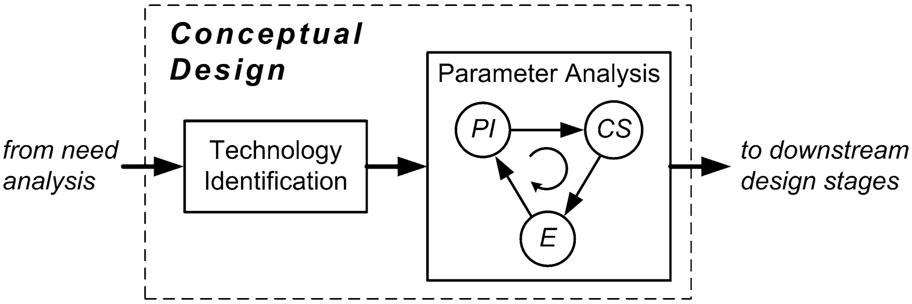
Fig. 1 The prescriptive model of PA consists of repeatedly applying parameter identification (PI), creative synthesis (CS) and evaluation (E). The descriptive model of moving back and forth between concept space and configuration space is also shown

The TI stage presents yet another enigmatic aspect: On the one hand, it avoids dealing with too many functions and their solution technologies by directing the designer to address only the core of the design task, for the sake of efficiency. On the other hand, we shall see that the method also enables recovery from a misled focus by a form of constructive backtracking: The user can at any point add new solution technologies, even revise the definition of the core task. This kind of recovery and backtracking processes has already been extensively studied in relation to search algorithms (Russell and Norvig 1995), so notions from that field will be used here to provide new insights on the design method.

1.3 Analogy to search

Design cannot be treated as a mere search problem (e.g., Hatchuel 2001) because the state space is not known, the goal state is not given, and often even the root state (the task) is ill-defined and evolves together with its solution (Dorst and Cross 2001; Maher and Tang 2003; Wiltschnig et al. 2013). However, search and design problems share a common theme of optimization in a broad sense. Design is not optimization in the "classic" computational problem-solving meaning, but it is concerned with finding good solutions, not just any solution. It also tries to reach the solution in an efficient manner, that is, with minimum resources such as time and knowledge acquisition effort. In order to better understand the observed efficiency of PA, some sort of optimization framework needs to be consulted.

Fig. 2 Technology identification is the first stage of conceptual design, wherein fundamental solution principles are proposed. It is followed by PA, the process of elaborating the solutions



One of the best known search methods, B&B, is a technique for finding optimal solutions to integer programming problems with a very large number of solutions (e.g., Hillier and Lieberman 2005, chapter 11). The basic idea is to divide and conquer so only a small fraction of the feasible solutions need to be examined. An original large problem is divided (the branching) into smaller and smaller subproblems that are more manageable. The conquering is done by bounding how good the best solution in the subset of feasible solutions can be, and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem. Many algorithms have been developed over the years, employing various search strategies such as breadth-first and depth-first, which differ in the order of expanding the nodes of the search graph to form subsets of the solution space.

Pearl (1984) points to the fact that the emphasis of B&B methods in OR is on the split-and-prune paradigm that is effective in establishing completeness and optimality. In contrast, the AI approach is concerned with the generate-and-test viewpoint, which is more relevant to creating or constructing new objects while searching for solutions. Heuristic[2] search in the context of path-seeking problems has been studied both in OR and AI, with the purpose of increasing efficiency. The most common use of heuristic information has been the bounding functions which control the B&B search, as in AI's popular heuristic shortest-path algorithm called A^* (e.g., Russell and Norvig 1995). These kinds of algorithms might be useful in design since they could help in reducing the number of design alternatives to be explored.

Interestingly, PA appears to be an odd combination of design and search.[3] It is a design process in the sense that there is no target solution at the beginning (contrary to classical "problem solving" cases) and surprises and discoveries are expected at each step, particularly through the evaluation of configurations. But its reasoning process and strategy also share many features with B&B methods: PA incorporates opportunities and activities of diverging that seem similar to B&B's branching, and PA relies heavily on constantly evaluating the artifact, and this is analogous to B&B's bounding by a cost function. Hence, studying PA might help to understanding how B&B can be extended to design processes. To make this extension rigorous, we use a design theory, C–K, to better follow how PA actually helps to navigate strategically in the unknown (unknown state space, unknown goal state), just as B&B helps to traverse the space of a complex optimization problem (with complex but known state space and goal state).

1.4 The C–K theory of design

C–K theory (Hatchuel and Weil 2003, 2009; Le Masson et al. 2010) is a general descriptive model with a strong logical foundation (Kazakçi et al. 2008), resulting in powerful expressive capabilities. The theory models design as interplay between two spaces, the space of concepts (C-space) and the space of knowledge (K-space). Four operators allow moving between and within these spaces to facilitate a design process: C → K, K → C, C → C and K → K. Space K contains all established, or true,[4] propositions, which is all the knowledge available to the designer at any given moment during the design process. Space C contains "concepts," which are undecidable propositions (neither true nor false) relative to K, that is, partially unknown objects whose existence is not guaranteed in K. A concept is a hypothesis of the following form: "there exists an entity x, for which the attributes $A_1, A_2, \ldots, A_i$ are true in K." Design processes aim to transform undecidable propositions into true ones by jointly expanding spaces C and K through the action of the four operators. This expansion continues until a concept becomes an object that is well defined by a true proposition in K. Expansion of C yields a tree structure, while that of K produces a more richly networked pattern. This short introduction already shows that C–K theory provides a representation of the imaginable "states" in its C-space, and this representation happens to have a tree-shape, just like the structure of the state space in B&B. Moreover, C–K theory tracks in K-space the knowledge expansion, i.e., all the knowledge acquired and used during the design process. In particular, the evaluation criteria of the product to be designed are stored and enriched in K-space. Hence, C–K theory appears to be a powerful framework to interpret the design activities used when designing with PA.

1.5 The main results

Using C–K theory and search and graph traversal notions, the present paper draws an analogy between the PA design method and search algorithms to shed light on the reasoning behind the design activities and the overall design strategy of PA. It does not deal with computable cost functions as in OR and AI, but interprets the specific discovery and elaboration process of the design artifact as an extended search process. The paper derives two main results:

---

[2] 'Heuristic' here means an experience-based technique, rule of thumb, intuitive method, etc.

[3] Connecting design to search, which is the process of exploring a state space, has been studied quite intensively and many techniques are available. An overview can be found in Dym and Brown (2012).

[4] 'True' here does not imply absoluteness; rather, it means that something is considered correct or valid in the designer's mind.

1. The evaluation step built into each cycle of concept development with PA first assesses the evolving design configuration, and this is followed by implicitly assigning “values” to all pending concepts and making a decision as to the next move. Indeed, the original PA method never mentioned value assignment; the clarification of this implicit activity is an original contribution of this paper. The values are assigned subjectively, based on the designer’s judgment. Many decisions in design are subjective, and the PA method only provides the framework to make those decisions. A positive (high value) evaluation result will guide the development further down along the same path, while a negative evaluation will direct the process to another, more appropriate path or branch of the concept tree. *PA can therefore be regarded as a generalized B&B process*, guided by evaluation but with two main extensions: The evaluation function in PA evolves over time because it is *subject to learning,* and the “branching” that takes place in PA is actually a design step, since the parameters and configurations are not chosen from a closed list but rather *result from this learning*. In fact, branching can even take place to a new path, previously unknown, that is discovered and generated while designing.
2. The logic of PA provides *strategic guidance in the concept tree of C-space* toward the goal. We show that it can be characterized as a depth-first strategy, which is known in AI to provide quick results, and we show that this strategy is efficient, in the sense that it enables to minimize the exploration needed to reach an acceptable design. At the same time, *it allows backtracking to a higher level if necessary*, which corresponds to a C–K theory “de-partition” or “inclusion,” and thus supports innovation. Moreover, the depthwise exploration is controlled by the PI steps in what we call “steepest-first” manner, that is, addressing the more difficult and challenging issues first. These critical parameters, in PA terminology, are not fixed during the design process; rather, they keep changing.

1.6 Summary

To establish these results, a rigorous interpretation of each PA step in C–K terms had to be developed first. The exact meaning of the elements of C-space and K-space, the nature of the four operators and a consistent way of drawing C–K diagrams were all established. The structure of the paper is therefore as follows: The PA method is demonstrated in the next section by applying it to a conceptual design task and explaining the pertinent activities. Next, the PA steps are modeled with the spaces and operators of C–K theory based on the logic and reasoning of both the design method and the theory. This is followed by a step-by-step demonstration of the case study in C–K terms. The paper concludes with a discussion of the results of this study and their consequences in regard to both PA and C–K theory.

It should be noted that although a design method (PA) and a design theory (C–K) are used in this paper extensively, there are still activities and phenomena that are not covered by either of them. Design is a complex human cognitive activity that no single model can fully explain, nor can it be completely encompassed by computer algorithms such as B&B. The methods and theories of design can guide designers, but the quality of the designers’ knowledge and decisions still plays an important role in the success of the process. The subjectivity of the decisions and their limitations as related to the notion of “bounded rationality” (Simon 1972; Kahneman 2003) cannot be avoided and should not be regarded as a deficiency, but rather as an inseparable aspect of real design practice.

2 Parameter analysis case study

The following is a real design task that had originated in industry and was later changed slightly for confidentiality reasons. It was assigned to teams of students (3–4 members in each) in mechanical and aerospace engineering design classes, who were directed to use PA for its solution after receiving about 6 h of instruction and demonstration of the method. The design process presented here is based on one third-year mechanical engineering team’s written report. This was a semester-long project that started with identifying and analyzing the need, and ended with detail design. Only part of the students’ conceptual design process is used here.

The task was to design the means of deploying a large number ($\sim$500) of airborne sensors for monitoring air quality and composition, wind velocities, atmospheric pressure variations and so on. The sensors were to be released at altitudes of some 3,000 m from an under-wing container carried by a light aircraft and stay as long as possible in the air, with the descent rate not exceeding 3 m/s (corresponding to the sensor staying airborne for over 15 min). Each sensor contained a small battery, electronic circuitry and radio transmitter, and was packaged as a ϕ10 by 50-mm long cylinder weighing 10 g. It was necessary to design the aerodynamic decelerators to be attached to the payload (the sensors), and the method of their deployment from a minimum weight and size container. The following focuses on the decelerator design only.

The design team began with analyzing the need, carrying out some preliminary calculations that showed that at the relevant Reynolds number, the drag coefficient C_D of a

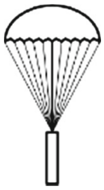
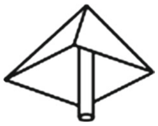
Parameter Analysis Terminology	***Reasoning Process***	***Outcome***
Design task statement	None (the "brief" is provided by the customer)	Design the decelerators for the given sensors and their means of packing and deployment in the air.
Need Analysis	Not described here. Can be done in several ways, such as QFD. It turns the "brief" into design requirements (specs.)	Design requirements: Sensors + low-cost decelerators packed compactly in container and staying in the air for 15 minutes, …
Technology Identification (TI)	What's the most difficult aspect of the task?	Deceleration of the sensors (and not packing, deployment, etc.).
	Which physical principles or technologies can be used to produce the deceleration?	Aerodynamic drag and buoyancy with the following technologies: 1. Flexible parachute 2. Rigid parachute 3. Balloon filled with light gas 4. Balloon filled with hot air.
	What's the behavior of each technology? Which is the best candidate? Pros and cons of each are listed and reviewed.	Flexible parachutes are most common in similar applications; rigid parachutes seem difficult to pack compactly; balloons may be much more complicated (inflation or heating needed). Hence, the technology of producing a large drag force by a flexible parachute seems most promising.
PI_1	The first conceptual issue (parameter) should be the chosen technology.	Parameter: "Produce a large enough drag force using a flexible parachute".
CS_1	Which particular physical structure would realize the flexible parachute concept?	Configuration: A 150-mm dia. hemispherical parachute, connected to the sensor with cords. 
E_1	What's the behavior of the hemispherical flexible parachute?	Drag force is ok and compact packing can be done by folding, but the parachute may not open because there isn't enough "pull" on it, and the cords may tangle.
	Shall we try to improve the last configuration or backtrack?	Try another technology from the TI stage.
PI_2	Use the next best technology for the decelerator design.	Parameter: "Use a rigid parachute to generate drag force".
CS_2	Which particular physical structure would realize the rigid parachute concept?	Configuration: A 150-mm diagonal square pyramid with the sensor rigidly attached. 
E_2	What's the behavior of the pyramidal rigid parachute?	Drag force is ok but compact packing is impossible because these configurations cannot nest inside each other.
	Shall we try to improve the last configuration or backtrack?	Try to improve the design by finding a way to pack it compactly.

Fig. 3 Description of the PA process used to design the airborne decelerators based on one team's written design report. The original presentation has been modified for brevity and clarity, but the content is preserved (continued on next page)

parachute-shaped decelerator is about 2, so to balance a total weight of 12–15 g (10 g sensor plus 2–5 g assumed for the decelerator itself), the parachute's diameter would be ~150 mm. If the decelerator is a flat disk perpendicular to the flow, the C_D reduces to ~1.2, and if it is a sphere, then $C_D \cong 0.5$, with the corresponding diameters being

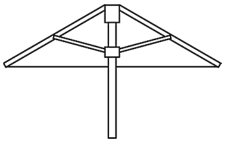
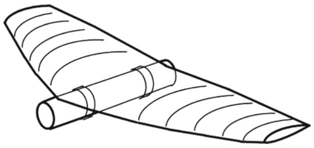
PI_3	How can the last configuration be improved? Combine the idea of flexible parachute that can be folded for packing with a rigid parachute that doesn't have cords and doesn't require a strong "pull" to open.	Parameter: "Use a frame + flexible sheet construction that can fold like an umbrella; use a spring for opening"
CS_3	Which particular physical structure would realize the "umbrella" concept?	Configuration: Lightweight skeleton made of plastic or composite with "Saran wrap" stretched and glued onto it. Hinges and slides allow folding. A spring will facilitate opening. 
E_3	What's the behavior of the "umbrella" parachute?	Drag force and compact packing are ok, but this structure is unreliable and expensive to manufacture because of the many moving parts.
	Shall we try to improve the last configuration or backtrack?	Parachutes (flexible and rigid) are problematic. Abandon this concept and try something else.
PI_4	Let's look at the problem differently, from an energy dissipation viewpoint instead of producing retarding force. Dissipation of the sensor's initial potential energy can be carried out by a long enough distance over which a smaller drag force can act.	Parameter: "Use a small aircraft that glides in spirals".
CS_4	Which particular physical structure would realize the glider concept?	Configuration: Wings with a span of 200 mm and a small twist to produce a 30-m diameter downward spiral. The wings can be made of Styrofoam and the sensor attached with plastic clips. 
E_4	What's the behavior of the spiraling glider?	This would work, seems cheap to make, and shouldn't have deployment problems. But how will the "gliders" be packed and released in the air?
	Shall we try to improve the last configuration or backtrack?	Continue with this configuration: design the container, packing arrangement, and method of deployment.

Fig. 3 continued

about 200 and 300 mm, respectively. It was also clear that such large decelerators would be difficult to pack compactly in large numbers, that they should be strong enough to sustain aerodynamic loads, particularly during their deployment, when the relative velocity between them and the surrounding air is high, and that being disposable, they should be relatively cheap to make and assemble. Further, the sturdier the decelerator is made, chances are that it will also be heavier. And the heavier it is, the larger it will have to be in order to provide enough area to generate the required drag force.

Figure 3 is a detailed description of the TI stage followed by the first portion of the PA process carried out by the design team. The distinct reasoning steps are listed alongside their respective outcomes. The wording and illustrations have been slightly modified for better clarity, but in essence, they follow the original students' work, which was a written report consisting of describing the TI stage as an essay and then listing of each PA step explicitly.

TI begins with the team specifying deceleration of the sensors as the most critical aspect of the design. For this task, they come up with the technologies of flexible parachute, rigid parachute, gas-filled balloon and hot-air balloon. Flexible parachutes can easily be folded for compact packing and represent a very common technological solution for slowing down the descent of airborne objects. Rigid parachutes can be made in various shapes, e.g., pyramids, cones or flat surfaces, and are also used in some similar applications. The balloons use both buoyancy and aerodynamic drag and can be packed compactly when deflated, but inflating or heating during or after deployment seems difficult. The concept chosen by the designers for further development is therefore the flexible parachute.

The first parameter identification step (PI_1) according to the PA method is simply to use the chosen technology as starting point. The concept ("parameter") is therefore to have a small conventional parachute provide the necessary drag force and allow compact packing in its folded state. The subsequent creative synthesis step (CS_1) realizes this idea in a specific hardware by sketching the configuration and sizing it with the help of some drag force calculations. Having a configuration at hand, evaluation can now take place (E_1), raising doubts about the operability of the solution: The 10-g weight of the payload may not exert a strong enough "pull" to open the parachute, and the cords may tangle during opening. Still within the evaluation step, the designers decide to abandon the flexible parachute concept and try another technology.

The next concept attempted (PI_2) is the rigid parachute from the TI stage, implemented as a square pyramid configuration (CS_2), but found to introduce a new problem—packing—when evaluated (E_2). Deciding to pursue this concept further, the designers propose a folding, semirigid parachute as the next concept (PI_3). It is implemented as an "umbrella" (a folding rigid skeleton with flexible canopy, CS_3) and evaluated (E_3), resulting in the conclusion that parachutes are not a good solution direction. This brings about a breakthrough in the design: Instead of thinking about producing a large retarding force to act over the vertical height of 3,000 m, which resulted in large structures that were unreliable and expensive, perhaps the problem should be considered from an energy viewpoint. Decelerating a falling object is concerned with dissipating potential energy by frictional work, and this can also be achieved by a smaller drag force over a larger distance, so instead of a vertical fall, the payload can be carried by a "glider" in a spiraling descent (PI_4). The resulting configuration (CS_4) shows an implementation of the last concept in words and a sketch, to be followed by an evaluation (E_4) and further development.

Several interesting points in this process are noteworthy. First, when the designers carried out preliminary calculations during the need analysis stage, they already had a vertical drag device in mind, exhibiting the sort of fixation in which a seemingly simple problem elicits the most straightforward solution. Second, TI yielded four concepts, all still relevant for vertical descent, and all quite "standard." A third point is that while the designers focused on synthesizing a device to slow down the descent, they constantly kept in mind the other required functionalities, such as compact packing, low cost and high reliability, as can be seen in the evaluation steps. Finally, it is interesting to note that when the "umbrella" concept failed (E_3), the designers chose not to attempt another technology identified at the outset (such as gas-filled balloon), but instead used the insights and understanding gained during the earlier steps to arrive at a totally new concept, that of a "glider" (PI_4). And while in hindsight this last concept may not seem that innovative, it actually represents a breakthrough in the design process because this concept was not apparent at all at the beginning.

We can conclude that PA seems to have allowed and supported a complex design process leading to a breakthrough when the known solutions were not sufficient and innovative alternatives became unavoidable. PA exhibited an interesting feature of recovery from a dead-end caused by a misled initial focus, and this recovery seems to have followed a form of constructive backtracking in the sense that the designers retreated from their initial focus but still kept in mind what had been learned during the initial exploration. This recovery and constructive backtracking can eventually lead to a breakthrough. Of course, this process depends on the designer's knowledge, experience and ability to use the method; however, it is interesting to clarify what in the method helps reach this "necessary breakthrough." To answer this, we need to interpret PA in terms of C–K theory.

3 Interpretation of parameter analysis activities in terms of C–K theory

Each of PA's reasoning steps described in the previous sections is broken down to elementary "moves" in order to formulate them as sequences of C–K operators. The basic premise for doing so is the epistemological difference in the meaning of "concept" between PA and C–K. Because knowledge is not represented explicitly in PA and because a design should be considered tentative (undecidable in C–K terms) until it is complete, both PA's parameters and configurations (i.e., the members of PA's concept space and configuration space, respectively) are entities of C–K's C-space. In other words, C–K theory does not distinguish between a concept's ideological foundation and its structural aspects while PA does. However, this is not meant to imply that no knowledge is used in PA's reasoning process; on the contrary, existing knowledge is extensively utilized in each PA step and new knowledge is constantly generated, so excursions to K-space should be incorporated in the interpretation.

3.1 Technology identification

Technology identification (TI) is a separate stage in the PA methodology that is done first. It involves three distinct reasoning steps:

1. *What is the most difficult aspect of the design task?* Here, the designer decomposes the overall task into sub-tasks and uses his/her knowledge and judgment to identify those sub-tasks that are relatively easy or have known solutions, and those that seem the most challenging, whose solution direction is not straightforward, or those requiring innovative approaches. Usually, only one or two such difficult tasks will be identified.
2. *Which physical principles or core technologies could be used to satisfy the difficult sub-task(s)?* Here the designer uses knowledge in the problem domain or looks externally (Internet, expert consultation, etc.) for similar problems and solutions. If none is found, or if some configurative solution is identified, the designer should abstract and generalize the sub-task at hand to the level of fundamental technological or physical principles.
3. *What is the behavior of each technology in the context of the task?* Cursory listing (and not a thorough selection process) of the pros and cons of each technology. Which one is the most promising candidate? It is implied here that some evaluation criteria can be found, perhaps among the design requirements, and that their application is analogous to assigning a “value” to each technology. A higher value implies that according to the designer’s judgment, the technology has better chances of resulting in a successful solution.

Interpreting these steps in C–K terms is shown in Fig. 4, with numbers attached to the arrows to denote the order of operations. K-space consists of existing knowledge items, marked by white background, and new knowledge that is shown with dark background. It begins with the known description of the overall design task (the “brief”) and the design requirements generated earlier. First, a K → K operator describes the isolation of the most difficult functional aspect of the task (step 1 above), followed by a K → C operator to establish the root concept, C_0. Core technologies for the main function are next generated by the designer based on existing knowledge and similar applications. This step (2 above) requires returning to K-space (a C → K operator), listing the possible technologies (K → K), and moving to C-space (K → C) to trigger the expansion of C_0 into C_1, C_2,…, C_i, which are concepts based on those technologies (a C → C operation). Finally, step 3 above calls for evaluating the candidate concepts and choosing among them. This is accomplished with a C → K operator that activates knowledge in K-space (K → K) to arrive at the desired outcome. A more rigorous explanation of how evaluation and selection work by assigning and maximizing a value is presented later in this section and in Sect. 4.

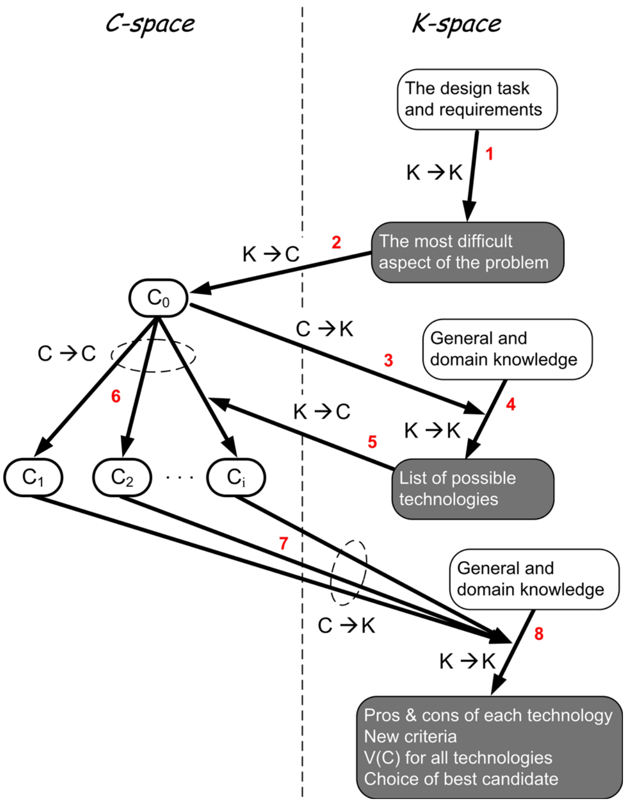


Fig. 4 Modeling the technology identification (TI) stage in C–K theory terms. The root concept C_0 is established, possible technologies identified and evaluated, yielding a value V(C) for all concepts. Finally, the best candidate is selected for further development

One point that may need clarification regarding this model is how the identified technologies can reside in both K-space and C-space at the same time. The answer lies in the different meaning (and therefore, logical status) of each occurrence: In K-space, the meaning is of technologies that are more or less known to be used in similar applications, and thus, they constitute knowledge items in the designer’s mind; in C-space, the meaning is of undecidable propositions, suggesting using these technologies to accomplish the specific task C_0. Note also that formally speaking, whenever a node of the concept tree in C-space is expanded (a “partition” in C–K terms), there is at least one more edge or path with the meaning of “other” that is not shown because it has not been explicitly used by the designer.

3.2 Parameters and configurations as attributes of C–K concepts

Following TI, the actual PA process consists of three steps (PI, CS and E) that are applied repeatedly and involves two types of fundamental entities: parameters (ideas, conceptual-level issues) and configurations (hardware representations, structure descriptions). To accommodate both entities in the C–K theory model, a refinement of the definition of a C–K concept as given in Sect. 1 is needed to distinguish between attributes that convey functional and behavioral purpose and meaning, and those that describe physical features. The former attributes are added at the PI step and correspond to PA's parameters. We shall call them "ideational" to emphasize that they contain the ideas that will eventually have led to the solution and denote them by P_1, P_2, etc. The latter attributes, on the other hand, are added at the CS step and correspond to PA's configuration items. We shall call them "structural" because they contain descriptions of hardware (see Footnote 2) and denote them as S_1, S_2, etc. That both types of attributes play a role in elaborating the design and therefore in describing a C–K concept, is a fundamental notion of PA that is also in line with Roozenburg's (1993) combinations of *mode-of-action* and *form* and Weber's (2005) combinations of (behavioral) *properties* and (structural) *characteristics*. The modified form of a C–K concept can now be written as "there is an object C_i, for which the ideational attributes $P_1, P_2,\ldots, P_m$ obtained with the structural attributes $S_1, S_2,\ldots, S_n$ are true in K." For brevity, we may also describe a concept as C_i $(P_1, P_2,\ldots, P_m, S_1, S_2,\ldots, S_n)$, preserving the original meaning.

Ideational and structural attributes differ not only in their meaning, but also in their role in the design process. Ideational attributes are used to define the evolving concept and represent the deep reasoning, the "ideology," behind the solution. They are explicitly integrated into the concept description in the PI steps as the "design path" and strongly and directly controlled during the design process by the results of the evaluation step. Structural attributes, on the other hand, are needed mainly to facilitate the evaluation and are more temporal in nature: They keep changing while developing the concept and may even be revised later, after completing the conceptual design phase and doing embodiment and detail design. In this sense, the structural attributes are not as significant as the ideational ones and only weakly and indirectly controlled through the CS steps; in other words, a change in the configuration is possible only by means of an ideational step (PI) and those changes usually are not unique.

3.3 The creative synthesis step

Having established the nature of a C–K concept's attributes, it is now possible to elaborate each of PA's reasoning steps. The outcome of the design process is clearly a member of PA's configuration space, so the interpretation begins with the CS step being applied to a PA parameter and results in a new configuration. CS involves a realization of an idea in hardware representation by particularization or instantiation (the opposite of generalization). It usually requires some quantitative specification of dimensions, materials, etc., that are derived by calculation, but not more than is required to establish the behavior of the configuration. In terms of C–K theory, if PA's parameters and configurations are both elements of C-space, then the CS step should start and end in C-space. However, because knowledge is required to realize an idea in hardware and perform quantitative reasoning, a visit to K-space is also needed. The CS step therefore begins with a C → K operator for searching for the needed knowledge by triggering a K → K (deriving specific results from existing knowledge). The new results, in turn, are used by a K → C operator to activate a C → C that generates the new concept, which adds structural attributes to realize the latest ideational attribute. This interpretation of CS as a sequence of four C–K operators is depicted in Fig. 5, where $C_{i+1} = C_i + S_{n+1}$. C–K concepts generated by adding PA parameters (C–K ideational attributes) are denoted in the figures by round-cornered boxes, while those resulting from adding PA configurational elements (C–K structural attributes) are shown as regular boxes. C–K's root concept, C_0, does not have structural attributes, so it will always have rounded corners, as in Fig. 4.

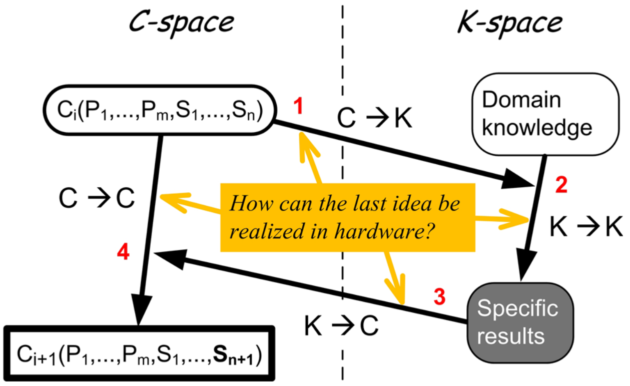


Fig. 5 Modeling the creative synthesis (CS) step in C–K theory terms. The latest ideational attribute P_m of concept C_i (which corresponds to a PA parameter) is implemented as structural attribute S_{n+1} of concept C_{i+1}

3.4 The evaluation step

One of the basic premises of PA is that parameters (concepts, ideas) cannot be directly evaluated in an effective manner; rather, they need to be implemented as configurations first and only then evaluated. This means that the evaluation (E) step begins with a C–K concept that includes structural attributes and attempts to deduce its specific behavior ("given structure, find behavior"), from which it will make a decision as to how to proceed. Reasoning from behavior to decision, however, includes two intermediary steps that are the key to understanding how the evaluation controls the design process so that it always moves in the most promising direction. First, the specific behavior of the configuration is used to establish possible new evaluation criteria, and those are applied (together with existing, older criteria) to all pending concepts to assign a value to them. Finally, a decision is made to move in the direction that maximizes this value.

The C–K interpretation is shown in Fig. 6: A C → K operator is used to initiate a K → K; the former being the operation of looking for the knowledge necessary for the evaluation, while the latter is the deductive reasoning that leads to deriving the specific behavior, new criteria and concept values, and making the decision as to how to proceed. The identification of new evaluation criteria is the actual *learning* done during the design process and is facilitated by having configurations to be evaluated. The combination of CS and E steps allows discovering unexpected behavioral aspects or revealing that some known functional issues have become more critical. New and critical issues in PA form the basis for the following PI step, as explained below.

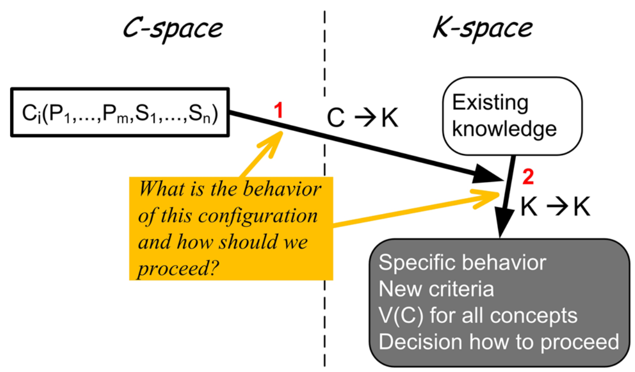


Fig. 6 Modeling the evaluation (E) step in C–K theory terms. Concept C_i corresponds to a PA configuration and existing knowledge is used to derive its behavior, deduce new evaluation criteria, calculate values V(C) for all pending concepts including C_i and make the decision as to how to proceed. The new criteria represent learning during design

The E step can be further described as activation of an evaluation function whose input arguments are the current concept C_i and all existing knowledge in K, including evaluation criteria learned in previous E steps. The function returns four arguments: First, the designer examines the configuration of C_i (its structural attributes $S_1,\ldots, S_n$) to see whether it works as it should, if it seems capable of satisfying the requirements, and if anything is still missing; this is the concept's specific behavior. Next, new evaluation criteria may be deduced from the behavior and added to the existing ones, to form a new set of criteria and a new ordering by importance within the set. Thirdly, all the concepts in the current C-space are evaluated with the updated criteria, and "values" V(C) are assigned to each concept. The values are not numerical, as B&B's costs, but rather a metric that represents the designer's judgment of the goodness and viability of the concept, its potential to lead to a conjunction for C_0, even its chances to materialize within given constraints of time and resources. Finally, a decision is made regarding the next move as one of the followings:

(1) *Termination* If the concept's behavior is as desired and nothing is missing (so no new evaluation criteria are added), and the value of the concept is higher than that of any other concept, then the design process is complete. All current attributes of the concept are accepted, and there is no subsequent PI step.

(2) *Following the current path* If an undesired behavior is detected, or something is missing in the concept, but its value is still the highest, then it should be improved by keeping its current attributes and adding a new ideational attribute in the next PI step (this is the most common occurrence).

(3) *Backtracking to a known but unexplored path* If the undesired behavior renders another existing concept more valuable, then the current development path should be stopped, and the next PI step will continue with the new highest value concept.

(4) *Backtracking to an unknown path* If the value of all existing concepts and technologies is very low, then all their attributes should be rejected and backtracking to C_0 will take place. The subsequent PI step will attempt to discover a new path.

3.5 The parameter identification step

The PI step begins with the results of the evaluation step in K-space, so it is a K → C operator that activates a C → C operator. The K → C operator carries the decision plus specific domain knowledge into C-space, while the C → C operator performs the actual derivation of the new concept.

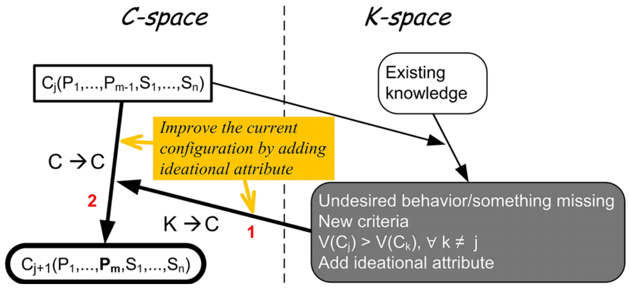


Fig. 7 Modeling the common occurrence of the parameter identification (PI) step in C–K theory terms following case (2) of evaluation (following the current path). Concept C_j has been evaluated (*thin arrows*) and weaknesses found. New criteria may be generated accordingly, but the value of C_j is still the highest, so ideational attribute P_m is added to form a new concept C_{j+1}

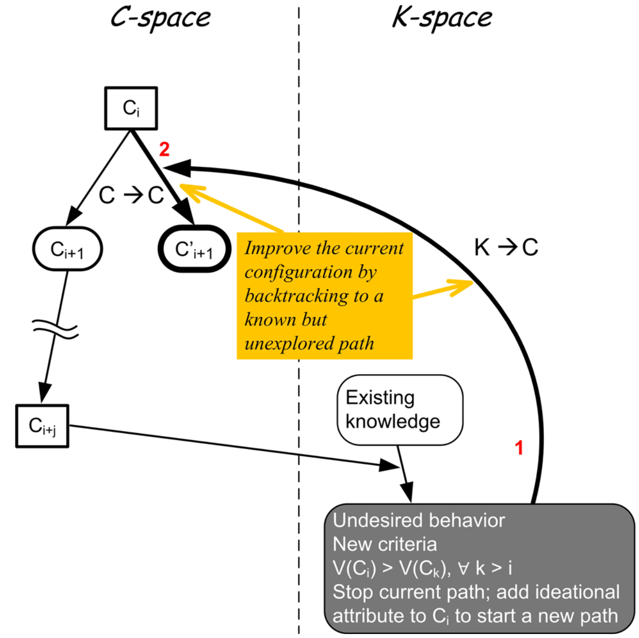


Fig. 8 Modeling the parameter identification (PI) step in C–K theory terms following case (3) of evaluation (backtracking to a known but unexplored path), with backtracking to a previous concept whose value suddenly becomes the highest. An ideational attribute P'_m is added to C_i and creates a path to C'_{i+1}, replacing the attribute P_m in C_{i+1}. If $C_i = C_0$ then C'_{i+1} represents a different technology from the TI stage that was known but not used so far

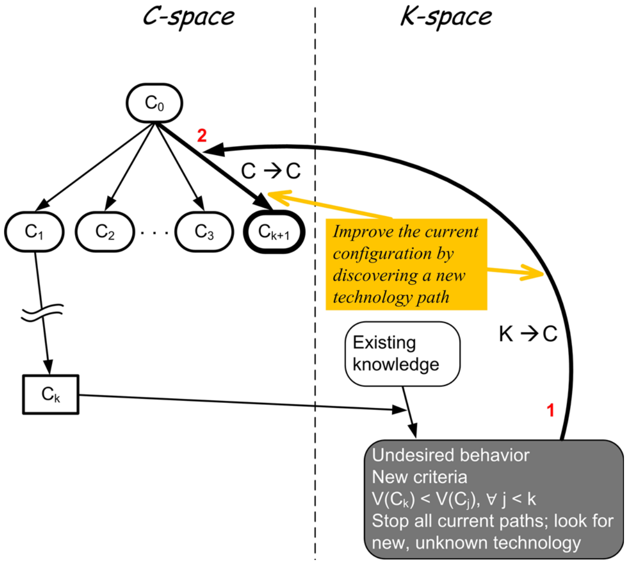


Fig. 9 Modeling the parameter identification (PI) step in C–K theory terms following case (4) of evaluation (backtracking to an unknown path), with backtracking to the root concept in order to *discover* a new technology. This implies discarding all the previous attributes and starting over

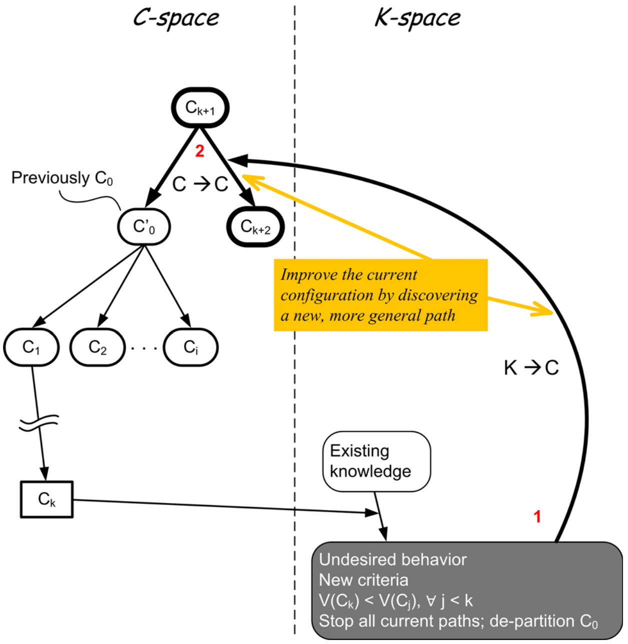


Fig. 10 Modeling the parameter identification (PI) step in C–K theory terms following case (4) of evaluation (backtracking to an unknown path), with backtracking to *higher* than the root concept in order to revise its identity. C_{k+1} becomes the new, more general root concept; C'_0 is a revised version of the previous root concept C_0; C_{k+2} is the beginning of a new, perhaps surprising path

Several cases can be distinguished based on evaluation results (2) to (4) above. The PI step can begin with a decision to improve the current design—case (2) above—as in Fig. 7, by adding an ideational attribute and staying on the current path. The PI step that follows case (3) above (backtracking to a known but unexplored path) is shown in Fig. 8, where a possibly long sequence of developing the concept along a path C_i, C_{i+1},…, C_{i+j} has already taken place. However, evaluating C_{i+j} reveals that a previous concept, C_i, now has a higher value, perhaps because the evaluation criteria have changed. Therefore, the current path is not continued, and a new path is developed from C_i

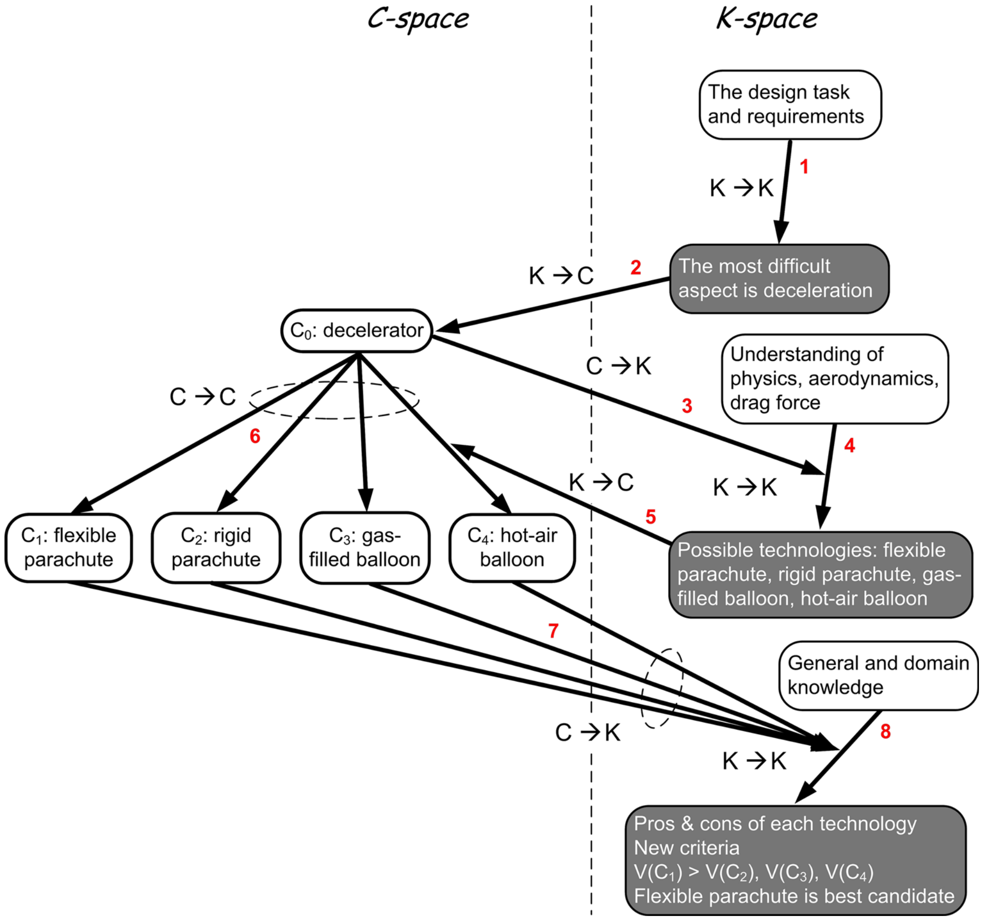


Fig. 11 C–K modeling of the technology identification (TI) stage of the decelerator design example. Producing drag force, simplicity and compact packing are used as evaluation criteria to assign the highest value to C_1, thus initiating a design path based on flexible parachute

instead. The latter path is not entirely new because it is the implicit "other" path that was known to exist when C_{i+1} was derived from C_i, but now it is made explicit. An ideational attribute P_m in C_{i+1} will be replaced by P'_m in C'_{i+1}. Sometimes, the backtracking required, as revealed by the evaluation, may be so substantial that it forces returning to the root concept and choosing another technology from those generated in the TI stage.

Case (4) of the evaluation step described above (backtracking to an unknown path) can be followed by any of the two possibilities described in Figs. 9 and 10. The designer may feel that the initial set of technologies identified earlier is not good enough, and look for new ones. He or she has by now gained some experience in working on the design task, including learning in K, so a new suitable technology, not considered earlier, may be discovered. This means that the concept development with PA will start over, and the ideational attribute added by the PI step is the technology to use in the new path (Fig. 9).

Finally, it may also happen that the learning during evaluation and the low values assigned to all existing concepts in case (4) of the evaluation ((backtracking to an unknown path) will lead the designer to re-examine the validity of the root concept itself. As shown in Fig. 10, this means that a C–K de-partition takes place, where a new, more general root concept emerges. The previously developed tree in C becomes one branch, while a totally new design path is created as another branch. The phenomenon of de-partition, or growing of the tree structure in C-space upward, at its root, has been demonstrated in (Le Masson et al. 2010, chapter 11).

4 Parameter analysis case study interpretation in C–K terms

A C–K-theoretical model of the decelerator design case study of Sect. 2 will now be elaborated to illustrate the results of the previous section. The design process began with the need, the problem to solve, as stated by the customer. A need analysis stage produced greater understanding of the task and the design requirements. This took place entirely in K-space and is not shown here. Next, TI focused the designers on the issue of deceleration (C_0), found possible core technologies, evaluated their pros and cons, and made a choice of the best candidate. As shown in

Fig. 11, this stage generated the root concept and four more concepts in C, thus establishing four possible design paths (note that for brevity, concepts in the diagrams list only the last attribute added to them; all other attributes are inherited from their ancestors and not shown):

$C_1 = C_0 + P_1 =$ decelerator based on (or having the ideational attribute of) flexible parachute,
$C_2 = C_0 + P_2 =$ decelerator based on rigid parachute,
$C_3 = C_0 + P_3 =$ decelerator based on gas-filled balloon,
$C_4 = C_0 + P_4 =$ decelerator based on hot-air balloon.

The evaluation of the four candidates at this stage is quite superficial: The designer imagines a decelerator based on that technology and uses some of the design requirements to judge the potential for success. Having only a general description of the technology in mind, the designers of the decelerators estimated that the two balloon technologies would be complicated, that the rigid parachute would be difficult to pack compactly, and so the common, straightforward solution of flexible parachute was valued highest; that is, $\mathrm{V}(C_1) > \mathrm{V}(C_2), \mathrm{V}(C_3), \mathrm{V}(C_4)$. Therefore, the evaluation criteria used were the capability to produce drag force (implicit), inherent simplicity (explicit) and potential for compact packing (explicit).

The following description of the PA process commences at this point. Figure 12 shows the first cycle of PI–CS–E as described in Fig. 3 and depicted with the formalism of Figs. 5, 6, 7, 8, 9 and 10. The result of the TI stage, to use a flexible parachute concept for the decelerator, is shown as the first PI step (for clarity, concepts C_2, C_3 and C_4 from TI are not shown now). This idea is next realized in hardware by a CS step, resulting in concept C_5 whose meaning is "a decelerator based on (or having the ideational attribute of) flexible parachute and the structural attribute of a 150-mm diameter hemispherical canopy with cords attached to the sensor." This last concept is evaluated by noting its behavior and generating two new criteria: opening in the air and tangling of the cords. These are added to the existing criteria, but their importance is high (these problems may render the concept useless), resulting in concept C_2 (see Fig. 11) becoming the highest valued. This corresponds to case (3) of the evaluation as in Fig. 8, so the decision is to abandon the flexible parachute design path and try the existing rigid parachute technology instead.

The second and third PA cycles are now added, as shown in Fig. 13, starting with the pruning of the flexible parachute branch and initiating a new branch based on the technology of rigid parachute (PI_2). This concept is realized as a 150 × 150 mm square pyramid (CS_2) and evaluated to discover a problem related to packing (an existing evaluation criterion), followed by a decision to improve this aspect of the design (E_2). This evaluation corresponds to case (2) of evaluation, so the process continues as in Fig. 7, with the improvement idea of using a folding frame with flexible skin, an "umbrella" (PI_3). This is implemented as a structure with rods, hinges, slides, "Saran wrap" and a spring (CS_3). Evaluation (E_3) of this last configuration produces its specific behavior as being so complicated that it would be costly and unreliable. Simplicity is an existing evaluation criterion used before, and low cost is one of the original requirements, although it is now used explicitly for the first time. Reliability, however, is a new criterion just found. All concepts associated with the rigid parachute technology are now valued low, joining the previously low-rated flexible parachutes. Moreover, the two remaining still untried balloon technologies are also assigned low values now, based on the updated set of criteria (ease of opening in the air and packing compactly, being low cost and reliable). This situation corresponds to case (4) of evaluation, where backtracking to the root concept or higher takes place, as in Figs. 9 and 10.

The fourth PI–CS–E cycle is depicted in Fig. 14. It begins with the evaluation result of step E_3 shown at the lower right corner. Having pruned the flexible parachute path earlier, the designers now prune rigid parachutes. They have two choices: either attempt to find a new, previously unknown technology for C_0, or revise the identity of C_0 by de-partitioning. Their accumulated experience, the learning, from the design process leads them to the understanding that they have so far considered only vertical drag devices and that the still unconsidered balloon technologies also belong to that category. So, they decide to take a fresh look at the problem (PI_4 in Fig. 14): From the energy dissipation viewpoint, a spiraling "glider" concept might work better. The C–K model of this step shows a de-partition, representing moving toward a more general concept, and in our case, redefining the identity of $C_0 = \textit{decelerator}$ to $C_0' = \textit{vertical drag decelerator}$ and partitioning C_9 to C_0' and C_{10}. This last concept is now implemented as the specific configuration C_{11} through the CS_4 step and evaluated, resulting in the conclusion that a conjunction for the new root concept has been reached. The design process may now proceed with the secondary issues (as identified in TI) of packing and deployment.

5 Discussion

A design theory used to study an empirically derived design method can provide explanation of the activities and phenomena, but also can be supported by the empirical data. The current study's main thrust was shedding light on PA using C–K theory, in particular the "recovery" logic in PA. On the way, some notions related to C–K theory have

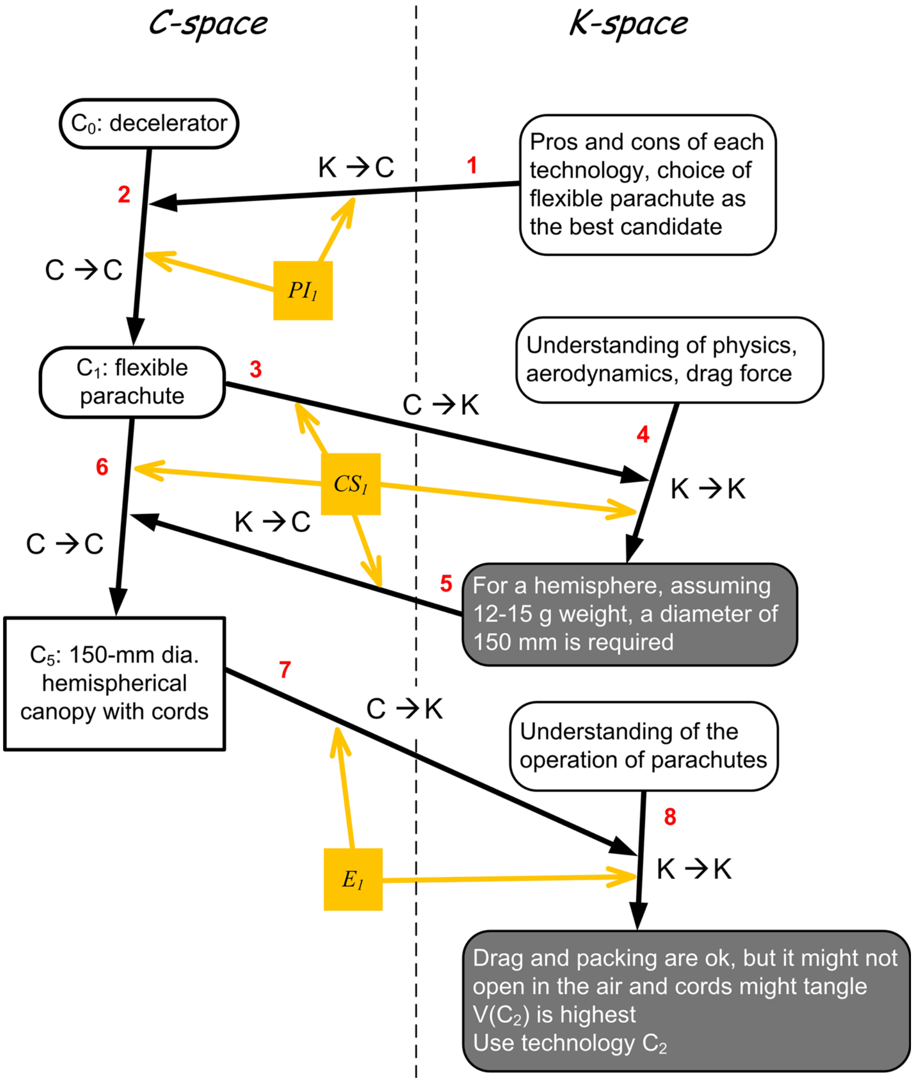


Fig. 12 C–K modeling of the first PI–CS–E cycle in the decelerator design example. The evaluation criteria are enriched thanks to analyzing the behavior of a configuration, by adding opening in the air and tangling of the cords. This results in the designers assigning the highest value to C_2*: rigid parachute* (not shown in the figure)

been clarified. The findings of this work—the interpretation of PA in terms of C–K theory and the inferences regarding the strategy of PA—are based on logical reasoning. The detailed case study is used only for demonstration purposes and is not the source of theoretical conclusions.

The decelerator design example is discussed first, followed by the interpretation of the pertinent entities (the elements of PA and C–K spaces) and design moves (steps and operators, respectively). A design method cannot be based on an "omniscient designer" hypothesis, nor can it be a purely random process; rather, it needs to have a strategy that guides the designer throughout the process. Many design methods appear as iterative processes with concept generation, concept selection and testing, and PA is no exception. Hence, the issue is rather to understand the kind of design strategies that are supported by these methods and that might be more specifically characterized by the methods. The design strategy supported by PA can be portrayed as focusing on one dominant issue at a time, examining known alternatives to address this issue, and, when necessary, looking for a breakthrough. We explain below how these specific features of the PA process can be related to two key aspects of its design strategy, namely the "steepest-first" ordering of the issues to be handled, and the continuous learning-based evaluation of the whole design path during concept development. Together, these aspects account for a certain form of efficiency and innovative capability of the PA methodology.

5.1 Recovery and constructive backtracking in the case study

The decelerator case study was chosen for this paper among many examples of using PA for conceptual design because it is relatively easy to follow in terms of the domain knowledge involved, and because it exhibits

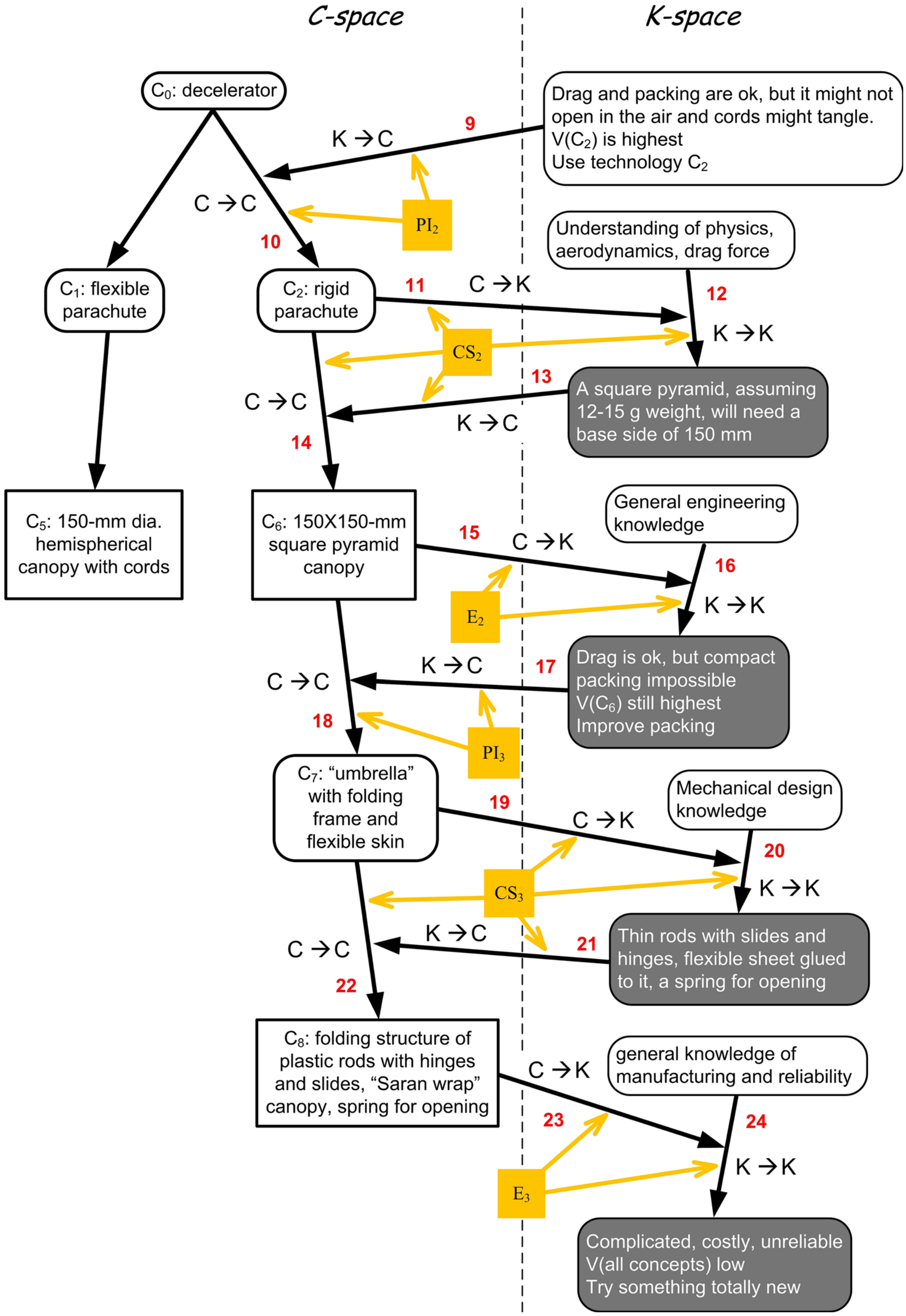


Fig. 13 The second and third PA cycles are added after pruning the flexible parachute branch. Both attempt to develop a concept with the rigid parachute technology. However, based on an updated set of evaluation criteria, the result is low values for all existing concepts and technologies

several interesting and relevant phenomena in a fairly short sequence of design activities. Other case studies of PA, as in Kroll et al. (2001), Condoor and Kroll (2008) and Kroll (2011), for example, tend to consist of much longer "chains" of PA cycles, sometimes requiring many background explanations to follow. And because the current work offers a rigorous translation of PA moves into C–K operators, a relatively short demonstrating example is just as good as a much more elaborate case study.

At the beginning of the decelerator design process, there was a TI stage of proposing several core technologies, listing their pros and cons, and selecting a best candidate for further development. Next, an attempt was made to pursue that design path, only to abandon it in the face of some difficulties. A complete backtracking took place next, and another design path initiated. This time, problems with the evolving artifact led to trying to improve it, but when more difficulties were encountered, the designers achieved

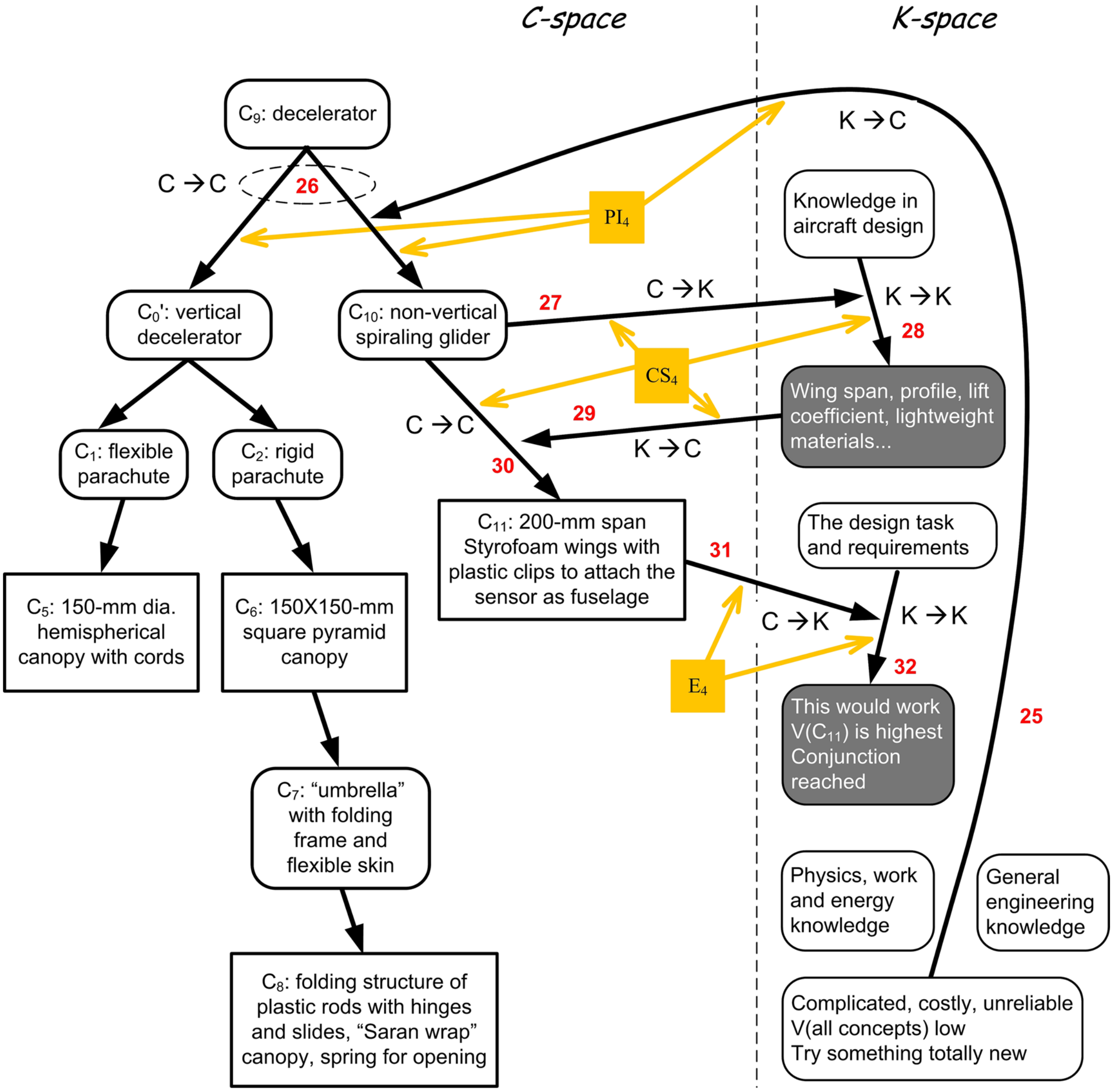


Fig. 14 C–K model of the fourth PI–CS–E cycle demonstrating a de-partition that leads to a conjunction for the root concept

a breakthrough by creating a totally new design path, and that terminated in success.

Can we consider this design process and its outcome to be optimal or exemplary? Certainly not: There might be even better solutions to this task, and other designers could perhaps have arrived at the same solution quicker. We cannot even say that each of the designers' decisions and choices was the best possible one. Nevertheless, we can observe many fundamental design activities that are not specific to using PA: looking for existing solutions to similar problems, selecting among alternatives, pursuing a concept through several iterations of refinement, reaching a dead-end, reasoning at the level of first principles, embodying ideas in hardware representations, evaluating the design artifact and learning while designing. This means that the modeling and interpretation proposed in this paper may be applicable also beyond the specific design method used here.

One aspect of the decelerator design task that deserves a short discussion is fixation. As many solution-driven engineers do (Lawson 2005, p. 182; Cross 2006, p. 7), the designers of the decelerator also began with straightforward, both well-known and less-known solutions for vertical descent (parachutes, balloons). They did not even consider non-vertical descents and certainly did not think of all the known solutions (e.g., spinning Samara seed-like devices, motorized mini "helicopters," and streamers, the kind of ribbons sometimes used in model rocketry instead of parachutes). The phenomenon of picking a limited number of known solutions and persevering with them is usually referred to as fixation and is often reported as limiting the designer's ability to innovate (Jansson and

Smith 1991; Linsey et al. 2010; Hatchuel et al. 2011). In this paper, we also refer to the sudden realization that vertical descent devices were not the only solution and to the subsequent creation of a new design path as recovering from fixation. However, it should be noted that most engineers rightly attempt to solve problems with known means first and only resort to innovative solutions when the conventional ones will not do. Furthermore, elaboration of an initial concept through cycles of evaluation and modification is PA's prescription for doing design and can also be viewed positively as exhibition of commitment.

5.2 Using C–K theory to interpret PA design "moves"

C–K theory has been clarified by this study with regard to its spaces and operators. By letting the elements of C-space correspond to both PA's parameters (concepts) and configurations (structures), a rigorous and consistent model of PA in terms of C–K theory has been derived. The following structure of a C–K concept makes a distinction between two types of attributes: "there exists an object *Name*, for which the group of ideational attributes P_1, P_2,… can be made with the group of structural attributes S_1, S_2,…". The ideational attributes correspond to PA's parameters and the structural ones to PA's configuration items. For example, concept C_8 in Fig. 14 can be described as:

> There exists an object C_8, for which the group of ideational attributes
> P_1 = produces vertical drag (inherited from C_0')
> P_2 = based on rigid parachute (inherited from C_2)
> P_3 = built as an umbrella, i.e., folding frame and flexible skin (inherited from C_7)
> can be made with the group of structural attributes
> S_1 = 150 × 150-mm square pyramid canopy (inherited from C_6)
> S_2 = constructed of plastic rods, hinges, slides, Saran-wrap and spring.

The last attribute, S_2, is the configuration item added to C_7 in response to the parameter P_3 to form concept C_8. The interesting thing to note is that except for the root concept in C–K (which is not defined as a PA entity), all other concepts have some attributes. But because a C–K concept can be either a PA parameter or configuration, and as PA excludes the possibility of having configurations without parameters to support them, the concepts in C–K sometimes have only ideational attributes, and sometimes ideational plus structural attributes; however, a concept cannot have structural attributes and no ideational ones.

All three PA design moves have been modeled in terms of sequences of the four C–K operators: PI corresponds to the pair [K → C, C → C], CS is the quartet [C → K, K → K, K → C, C → C], and E is the pair [C → K, K → K]. It can be seen that although PA's fundamental entities, concepts and configurations, belong in C–K's C-space, all three PA moves require a visit to K-space. K-space contains existing knowledge in the problem domain and related areas, and also meta-knowledge—knowledge about the design process itself—although this last item was not shown in the diagrams of this paper. More importantly, K-space is where learning is carried out during the design process by evaluating the evolving artifact, deducing its behavior, assigning values to all pending concepts and generalizing this new knowledge to form a decision as to how to proceed.

The role of PI, parameter identification, as the most important step in PA has also been clarified. PI consists of identifying, through the learning facilitated by successively evaluating configurations, what the relevant new parameters to be kept are, i.e., to be considered as the defining ideas for the concept. Note that "identification" in PI carries the meaning of a *design* action, and not just a selection in a decision making process, since the concept keeps changing. Some attributes are identified and selected in K-space when forming a configuration (in the CS step), but the most influential step on the final outcome is adding ideational attributes in C-space to generate new concepts.

Some basic notions of C–K theory have also been clarified by this study. It has been shown that K → K operators represent deductive reasoning, generating new knowledge from existing one, but their action needs to be triggered by a reason, a purpose, and this is represented by a C → K operator. Such activation of a K → K operator takes place in two cases: first, as part of a CS step, where the meaning is searching for the knowledge needed to implement an idea as a configuration, for example, using the drag force formula to calculate the parachute diameter given the weight and desired rate of descent. The second case is during an E step, meaning looking for the knowledge needed to deduce the behavior of a configuration. (An exception to this triggering of K → K is the steps marked with a "1" in Figs. 4 and 11, denoting the transition from the preceding need analysis or task clarification stage to conceptual design.) Likewise, a K → C operator uses knowledge for initiating a C → C operator. As demonstrated in this study, C → C operators do exist, representing the derivation of a new C–K concept from another while inheriting its attributes. However, this operation does not happen by itself in C-space, only if activated by a K → C operator, as part of a PI or CS step. This validates C–K theory's premise of mutual expansion: K-space is responsible for the expansion in C-space, but perhaps somewhat surprising, C-space drives the generation of new knowledge—the learning—in K-space.

Another issue clarified is that the tree structure of C-space is not chronological, as demonstrated by the de-partition that took place. To capture the time-dependence of the design process, C–K's concepts were labeled with a running index and the operator arrows numbered. One of the fundamental notions of C–K theory is that everything in C-space represents "undecidable" entities, but once a "true" or "false" logical status is assigned to it, this entity becomes knowledge and "moves" to K-space. The interpretation of this notion in the current paper is that concepts in C remain undecidable even when the designer finds them deficient and abandons their further development in favor of pursuing other paths. For example, concepts C_5 and C_8 of Fig. 14 are still present although their development was stopped due to their low value, as determined by the corresponding evaluation steps. This means that the designers could return to these concepts at a later stage, if their value increased through learning new knowledge.

5.3 Steepest-first exploration

At two distinct steps of the design process, the designer is required to make a choice or selection among issues at the functional or conceptual level. First, during TI, the designer examines the design task with the aid of added understanding gained during need analysis, to identify the most difficult aspect of the task. The methodology directs the designer to begin the design process with that issue, as demonstrated by choosing "deceleration" for the root concept. The second step requiring such selection is PI, activated at every cycle of PA by the preceding evaluation. Here, the designer should consider the "most critical conceptual-level issue" of the moment.

At both instances, the selection represents an efficient strategy of depth-first that is quite unique: Instead of getting the easier aspects out of the way first and handling the more difficult issues later, as might seem reasonable in general problem solving, or perhaps addressing *all* the issues simultaneously, as in systematic design, the PA methodology sends the designer in the "steepest" direction. This heuristic rule is based on two insights. First, there is the recognition of the function–form dependence in design, which means that a structure created to provide some function usually results in new behaviors, themselves requiring structural modifications, and so on (Gero and Kannengiesser 2004). To make this potentially endless cycle more manageable and efficient, it makes more sense to address the higher-difficulty aspects first, assuming that the easier needs will be satisfied later in a way that complies with the already-solved problems.

The second insight inspiring the "steepest-first" heuristics is the fact that most designers form quite early an underlying core concept and keep pursuing it even when faced with implementation difficulties. This realization was central to forming the original PA methodology by observing designers (Li et al. 1980) and has been confirmed by both anecdotal evidence and empirical studies of practicing designers. For example, Cross (2004) calls this central idea the "principal solution concept" and Lawson (2005) names it the "primary generator idea." This fundamental design idea dominates the rest of the functional aspects and therefore needs to be addressed early. Most of the critical issues with the evolving design cannot be identified upfront, but rather arise as the design unfolds according to the main idea.

In compliance with the "steepest-first" strategy, issues of packing, deployment, etc. were put off during the TI stage of the decelerator design example. Clearly, if the decelerator itself is still undefined, one cannot design its means of packing and deployment; nevertheless, these secondary issues were not completely ignored when designing the decelerators themselves. The initial "central idea" was using flexible parachutes, but it was abandoned quite early, perhaps indicating that the student designers were not experts. A more experienced designer might have addressed the new critical issues of opening the parachute and tangling of the cords while keeping the original concept. He or she could, for example, introduce means of forcing the parachutes to open using the airflow created by the airplane's movement, or mechanically pulling on the canopies with static lines.

The most critical aspect identified with the next central idea (rigid parachute) was the packing of relatively large, non-nesting structures. The decision to opt for an umbrella-like foldable configuration could not have been made earlier, when thinking of flexible parachutes. Furthermore, the implementation with plastic rods, hinges, etc. facilitated the identification of cost and reliability as key drawbacks. Here, again the designers could have chosen to modify the current concept by thinking of ways to simplify the structure, perhaps looking at cocktail umbrellas or the art of origami. Instead, they generated another central idea, that of a glider.

The steepest-first strategy is an inherent part of the PA method, constituting meta-knowledge that resides in K-space and originates from training and practicing the method. The current interpretation through C–K theory and the analogy to B&B, however, allow us to suggest that this strategy is in fact carried out through the repeated application of evaluation steps. When faced with a need to pick the "most critical issue" among several choices, the selection will be of the issue that could potentially reduce the uncertainty most steeply and therefore generate more value for the resulting concept.

5.4 Design path evaluation

A significant result of this study is that the PA design process is controlled by a learning-based state and path evaluation function that is responsible for both the efficiency and innovative capability of the inherent strategy. For evaluation to be credible and useful, PA encourages the designer to quickly implement ideas as hardware representations and not rely on assessing abstract ideas. In this sense, the strategy resembles the use of (virtual) rapid prototypes as an aid to the design ideation process. Such rough sketches of prototypes with initial sizing and perhaps other specified properties represent the current state of the solution and can readily be evaluated. In some cases, simulations and physical models are needed for testing and experimentation. Even more important, the design path that has led to the current state can also be assessed, with the robustness of the evaluation results constantly increasing by learning. Comparing PA to OR's and AI's B&B family of search algorithms, the former exhibits a more general strategy wherein the evaluation function is not fixed a priori, nor does it change algorithmically, but rather, it is based on a process of learning during design and can be modified accordingly at any time.

At the beginning of the process, during the TI stage, technologies for the core task are proposed, their advantages and drawbacks listed, and a selection of the best candidate is made. Although this is clearly an activity of evaluation, there is still no learning involved, and it only serves to tentatively point in the general direction or path of the design development to initiate the PA process. In fact, PA's depth-first with backtracking allows changing the initial choice quite easily, as demonstrated in the decelerator example. Moreover, the final design does not necessarily have to be based on one of the core technologies identified at the outset. In the decelerator example, the designers listed parachutes and balloons and ended up with an original concept of a spiraling glider. In general, if we use the term "innovative" to describe solutions that are not based on the core technologies known at the beginning of the design process, two mechanisms for innovation have been revealed through the C–K interpretation: (1) looking for a new technology (this has not been demonstrated by the decelerators example but is depicted in Fig. 9) and (2) re-examining the root concept and de-partitioning C-space.

C–K modeling, however, reveals much more about the E step. In addition to looking at the latest version of the evolving design and judging the extent to which it works properly and satisfies the design task requirements, it also examines the whole design path which is included in the concept description. The ideational attributes of the evaluated C–K concept constitute a trace of the stream of consciousness, the flow of thoughts, from the root concept to the present state, while the structural attributes form the description of the physical artifact. The designer can conclude that the current configuration represents a conjunction for the root concept, and then the design is complete, or that there is a disjunction and the process should continue. In the latter case, the exact reason can be identified: It may be a specific S_i (a structural attribute) that needs to be modified or a P_j (ideational attribute) that now turns out to be problematic. Accordingly, the decision about how to proceed will address the pertinent issues.

Learning-based evaluation has been demonstrated through the case study of this paper. Choosing the flexible parachute concept (C_1 in Fig. 12) was equivalent to forming a hypothesis that a solution based on this technology was feasible. To be tested, that hypothesis needed to be refined by embodying the idea in specific hardware (C_5). The evaluation at that moment addressed two issues: (1) did the specific hardware represent a good solution and (2) was a solution based on flexible parachutes reachable? The designers' conclusion, that the 150-mm diameter hemispherical parachute presented significant shortcomings, was translated into a low value for the whole design path of flexible parachutes and a corresponding decision to attempt another technology whose value was higher.

In the second evaluation, that of rigid parachutes, drawbacks of the configuration were initially addressed by keeping the design path and attempting to modify the concept. Only during the next evaluation step, E_3, the designers had already learned enough to assign a low value to both the flexible parachute and rigid parachute paths and conclude that they should take a fresh look at the underlying physics. Moreover, the two untried design paths of using balloons were also put aside (again, through assignment of low values) in light of the newly learned insight regarding vertical versus non-vertical descents.

Evaluation in PA can therefore be generalized as follows. A configuration that consists of a C–K concept of the form C_i (P_1, P_2,…, P_m, S_1, S_2,…, S_n) is given. The hardware description (S_1, S_2,…, S_n) is examined to reveal whether it would work properly and satisfy the design requirements. If the answer is "yes," then the design is complete. Otherwise, some undesired behavior has been detected because something is still missing or a problem is discovered. If the value of the current concept is still higher than all other concepts, the design process should continue by modifying the set (P_1, P_2,…, P_m), which is the ideation sequence in the design path. If the evaluation shows that the design path as a whole is good, then it is kept and the design process continues along it. A relatively minor modification would be an addition of a new ideational attribute P_{m+1}, followed by implementing it as a new structural attribute S_{n+1}. Or perhaps the current problematic aspect can be resolved by backtracking to a previous decision point, changing the path slightly from P_m to P'_m, and realizing it as S'_n instead of S_n.

However, it may well happen that examination of (P_1, P_2,…, P_m) will trace the current problematic situation to as early as P_1, meaning that the whole design path is undesirable. Clearly, this can happen by the designer making a mistake when generating P_1 in the first place, or it can represent a learning process: an original thought that was correct at an earlier time turns out later to be wrong, after acquiring new knowledge by means of the actual activity of designing. Backtracking to the beginning of the design path is a major shift in the design process and is carried out through reasoning about the whole concept space and at the ideation level (PA's parameters). It can lead to choosing another technology already listed as a possible candidate or to searching for a yet-unknown technology, or even to re-examining the validity of the root concept and attempting a de-partition.

The innovative capability of PA's strategy has been attributed to de-partitioning in C-space, facilitated by the extensive learning during the concept development process, which in turn refined the evaluation function. PA allowed recovery from the effect of the initial fixation by learning accomplished through the repeated generation and evaluation of "standard" configurations during the design process. This learning manifested itself in the production of new knowledge, or K-expansions in C–K terms, and discovery of a final solution that was not included in the fixation-affected initial set of technologies. Moreover, the important attribute responsible for the de-partitioning was the vertical descent, and this was implicit—either ignored or unrevealed—at the beginning, when proposing concepts C_1 to C_4. Only evaluation based on learning helped discover the criticality of this attribute, which was subsequently subtracted from the properties of the emerging concepts. This generalization in the definition of the root concept—de-partitioning or inclusion in C–K terms—has been identified as the exact mechanism though which innovation was achieved.

The learning process and the way the design progression is controlled by the evaluation, as described above, are similar to the more rigorous presentation in Ullah et al. (2012). They attribute the learning in design modeled with C–K theory to an increase in epistemic information content due to the presence of undecidable concepts. When the designer is unable to reduce the information content in the current path, a different path is attempted.

It is also interesting to compare PA's strategy to classical systematic design methods. In the latter, extensive design work at the functional, conceptual and more detailed levels would have taken place before carrying out an evaluation that could lead to a similar de-partitioning. PA, on the other hand, does not postpone the evaluation; rather, it is incorporated in every step—including evaluation of the design path—and becomes more robust as the design unfolds due to the built-in learning.

5.5 Practical implications for PA

Studying PA with C–K theory helps to answer some common practical questions regarding this design method: How can one prioritize the unknown issues? How efficient is PA? When is PA applicable? What are its limitations? We briefly address these issues below.

As elaborated in Sect. 5.3, prioritization to determine the present most critical issue depends on the designer's knowledge, experience and skill. There is no one "correct" way to prioritize, and different designers may derive different results. However, the learning process embedded in PA helps to re-discuss the initial choices and change them as needed and as might become apparent to the designer at later stages of the process.

The claim that PA incorporates an efficient strategy is clarified by the analogy to B&B. Just as the latter helps to avoid exhaustive explorations of complete search spaces, PA guides the designer to move in the most promising direction, and this is explained as the logic of implicit value assignment. We can therefore see this as a form of B&B extended to design processes. Because it appears that the efficiency and exploration capacity of the PA method depend on the value assignment logic, a possible improvement of PA may be to ask its practitioners to try to explicate the value assignment, or it may be possible to clarify different PA strategies associated with different value assignment logics. For example, an approach similar to "General-Opinion and Desire" (GD) proposed in Ullah (2005) may help assign values to alternative concepts in a structured way. GD provides means to encode the extent to which a concept is both known and desirable using several criteria and linguistic input information provided by the designer.

We can now begin to specify some features of PA's domain of relevance and limitations. PA is neither specifically adapted to situations where the goal of the design process is to use only known solutions (i.e., routine design tasks) nor to generating intentionally many breakthroughs purely for the sake of innovation. Rather, PA is oriented toward efficiently and quickly finding a good solution. If known technologies suffice, PA will support a design using them. If known solutions are unsatisfactory, PA will allow discovering other technologies and possibly new perspectives on the design task, leading to a breakthrough.

One possible limitation of PA stems from its depth-first strategy: If a good solution is reached, the designer will probably stop with that and not explore other options. Clearly, the PA process may be deliberately applied to other technologies to generate alternative solutions, but it would never be as exhaustive as morphological approaches, for example. Moreover, PA seems to require more skill and ability from the user than systematic design methods such as in Pahl et al. (2007). As we have seen, the judgment needed to continually prioritize critical issues and evaluate partial solutions plays a significant role in PA, and may be more demanding than systematically addressing all pertinent functional issues, creating numerous combinations of solution concepts, and finally selecting among them.

6 Conclusion

C–K theory was shown to be able to model PA's steps, which are fundamental design "moves": generating an idea, implementing the idea in hardware representation and evaluating the configuration. It also showed that PA supports innovative design by providing a means for recovering from fixation effects. Conversely, PA helped to clarify the structure of C–K's concepts, operators and C-space itself, and to emphasize the importance of K-space expansions.

C–K theory is, by definition, a descriptive model of design and does not contain a strategy for designing. However, it is capable of providing explanations to what happens during design and interpreting the strategy of specific design methods. The main results of this study are the explanation of PA's strategy as steepest-first exploration, controlled by a learning-based design path evaluation. These have been clarified by applying C–K theory and some search-related notions from OR and AI, and demonstrated with the decelerator design case study.

Several interesting issues remain for future research. We have not touched in the present work the cognitive aspects of identifying critical conceptual design parameters and the taxonomy of the knowledge involved. In other words, what particular knowledge and capabilities are required of the designer when making the various decisions, and what exactly happens in K-space during PA as related to the structures of knowledge items and their role as drivers of the design process? In addition, it might be useful to try to identify additional innovation mechanisms in PA that can be explained with C–K theory, and compare PA to other design methods with the tools of C–K theory. An interesting future direction might be the integration of creativity methods, such as TRIZ, in the framework of PA to provide even more innovation capabilities.

Acknowledgments The first author is grateful to the chair of "Design Theory and Methods for Innovation" at Mines ParisTech for hosting him for furthering this research. This work was supported by the Israel Science Foundation under Grant No. 546/12.

References

Condoor S, Kroll E (2008) Parameter analysis for the application of the principle of direct and short transmission path: a valve-actuator design case study. J Eng Des 19:337–357

Cross N (2004) Expertise in design: an overview. Des Stud 25:427–441

Cross N (2006) Designerly ways of knowing. Springer, London

Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem–solution. Des Stud 22:425–437

Dym CL, Brown DC (2012) Engineering design: representation and reasoning, 2nd edn. Cambridge University Press, Cambridge

Elmquist M, Segrestin B (2007) Towards a new logic for front end management: from drug discovery to drug design in pharmaceutical R&D. Creat Innov Manag 16:106–120

Eris O (2006) Insisting on truth at the expense of conceptualization: can engineering portfolios help? Int J Eng Ed 22:551–559

Gero J, Kannengiesser U (2004) The situated function–behaviour–structure framework. Des Stud 25:373–391

Gillier T, Piat G, Roussel B, Truchot P (2010) Managing innovation fields in a cross-industry exploratory partnership with C–K design theory. J Prod Innov Manag 27:883–896

Hatchuel A (2001) Towards design theory and expandable rationality: the unfinished program of Herbert Simon. J Manag Gov 5:260–273

Hatchuel A, Weil B (2003) A new approach of innovative design: an introduction to C–K theory. In: Proc 14th Int Conf Eng Des (ICED'03), Stockholm, pp 109–124

Hatchuel A, Weil B (2009) C–K design theory: an advanced formulation. Res Eng Des 19:181–192

Hatchuel A, Le Masson P, Weil B (2011) Teaching innovative design reasoning: how concept–knowledge theory can help overcome fixation effects. Artif Intell Eng Des Anal Manuf 25:77–92

Hillier FS, Lieberman GJ (2005) Introduction to operations research, 8th edn. McGraw-Hill, New York

Jansson DG, Smith SM (1991) Design fixation. Des Stud 12:3–11

Kahneman D (2003) A perspective on judgment and choice: mapping bounded rationality. Am Psychol 58:697–720

Kazakçi AO, Hatchuel A, Weil B (2008) A model of CK design theory based on term logic: a formal CK background for a class of design assistants. In: Proc 10th Int Des Conf (Design 2008), Dubrovnik

Kroll E (2011) Structured concept development with parameter analysis. In: Proc Int Conf Eng Des (ICED'11), Copenhagen

Kroll E (2013) Design theory and conceptual design: contrasting functional decomposition and morphology with parameter analysis. Res Eng Des 24:165–183

Kroll E, Koskela L (2012) Interpreting parameter analysis through the proto-theory of design. In: Proc 12th Int Des Conf (Design 2010), Dubrovnik

Kroll E, Condoor SS, Jansson DG (2001) Innovative conceptual design: theory and application of parameter analysis. Cambridge University Press, Cambridge

Lawson B (2005) How designers think: the design process demystified, 4th edn. Architectural Press, Oxford

Le Masson P, Weil B (2013) Design theories as languages of the unknown: insights from the German roots of systematic design (1840–1960). Res Eng Des 24:105–126

Le Masson P, Weil B, Hatchuel A (2010) Strategic management of innovation and design. Cambridge University Press, Cambridge

Li YT, Jansson DG, Cravalho EG (1980) Technological innovation in education and industry. Van Nostrand Reinhold, New York

Linsey JS, Tseng I, Fu K, Cagan J, Wood KL, Schunn C (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Mech Des 132:041003

Maher ML, Tang HH (2003) Co-evolution as a computational and cognitive model of design. Res Eng Des 14:47–63

Pahl G, Badke-Schaub P, Frankenberger E (1999) Résumé of 12 years interdisciplinary empirical studies of engineering design in Germany. Des Stud 20:481–494

Pahl G, Beitz W, Feldhunsen J, Grote KH (2007) Engineering design: a systematic approach, 3rd edn. Springer, London

Pearl J (1984) Heuristics: intelligent search strategies for computer problem solving. Addison-Wesley, Reading

Reich Y, Hatchuel A, Shai O, Subrahmanian E (2012) A theoretical analysis of creativity methods in engineering design: casting and improving ASIT within C–K theory. J Eng Des 23:137–158

Roozenburg NFM (1993) On the pattern of reasoning in innovative design. Des Stud 14:4–18

Russell SJ, Norvig P (1995) Artificial intelligence: a modern approach. Prentice Hall, Upper Saddle River

Shai O, Reich Y (2004a) Infused design: I theory. Res Eng Des 15:93–107

Shai O, Reich Y (2004b) Infused design: II practice. Res Eng Des 15:108–121

Shai O, Reich Y, Hatchuel A, Subrahmanian E (2013) Creativity and scientific discovery with infused design and its analysis with C–K theory. Res Eng Des 24:201–214

Simon HA (1972) Theories of bounded rationality. In: McGuire CB, Radner R (eds) Decision and organization. North-Holland, Amsterdam, pp 161–176

Smith G, Richardson J, Summers J, Mocko GM (2012) Concept exploration through morphological charts: an experimental study. ASME J Mech Des 134:051004

Tomiyama T, Gu P, Jin Y, Lutters D, Kind Ch, Kimura F (2009) Design methodologies: industrial and educational applications. CIRP Ann Manuf Technol 58:543–565

Ullah AMMS (2005) A fuzzy decision model for conceptual design. Syst Eng 8:296–308

Ullah AMMS, Rashid MM, Tamaki J (2012) On some unique features of C–K theory of design. CIRP J Manuf Sci Technol 5:55–66

Weber C (2005) CPM/PDD—an extended theoretical approach to modelling products and product development processes. In: Proc 2nd German-Israeli symp advances in methods and systems for development of products and processes, TU Berlin/Fraunhofer IPK, Berlin, pp 159–179

Wiltschnig S, Christensen BT, Ball LJ (2013) Collaborative problem–solution co-evolution in creative design. Des Stud 34:515–542