Design Science

In search of the ideal design: systematic trade-off mitigation and constraint management in optimal design synthesis

Nökkvi S. Sigurdarson[1,2], Panos Y. Papalambros[3] and Tobias Eifler[1]

[1]Department of Civil and Mechanical Engineering, Technical University of Michigan Department of Mechanical Engineering, Denmark
[2]Delivery Solution Innovation, Novo Nordisk A/S, Hillerod, Denmark
[3]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract

Successful synthesis of a new design requires balancing of trade-offs that arise from multiple competing design objectives and constraints. Early-stage design synthesis typically does not consider detailed technical constraints; a task left to late-stage mathematical design optimisation to refine an already-determined configuration. The recently developed Multi-Objective Monotonicity Analysis (MOMA) has shown that design optimisation can be used successfully in configuration redesign. This article extends the MOMA approach to early-stage design. Synthesis of an aptly named *ideal design* is achieved by focusing on the avoidance or reduction of trade-offs and by managing active constraints across all stages of the design process. The ideal design meets a set of formal conditions, which provide the basis for a systematic collection of corresponding design principles that can be selectively combined to create new embodiments, avoiding overly restrictive trade-offs and constraints. These principles are consistent with the decision making of experienced mechanical designers, shown here in the industrial practice for designing drug delivery devices.

Keywords: Design Synthesis, Trade-offs, Optimal Design, Monotonicity Analysis, Design Principles, Mechanical Design

Received 18 January 2023
Revised 29 September 2025
Accepted 30 September 2025

Corresponding author
Nökkvi S. Sigurdarson
nssd@novonordisk.com

Des. Sci., vol. 11, e45
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2025.10034

1. Introduction

Design is the task of developing a well-defined solution to a poorly defined problem, and engineering design is no exception (Cross 2004; Design Council 2007; Pahl & Beitz 2007). In this article, we address problems in mechanical design, but we posit that the ideas presented are applicable to design problems in general. Initially facing a relatively undefined, perhaps even infinite, solution space (Cross 2004), the engineering design task starts with an abstract, conceptual idea and gradually embodies it, i.e., turns it into a concrete physical entity that fulfils the specified needs. In this context, conceptual design aims at the generation of novel solutions to incompletely defined problems. However, it is often overlooked that this ideation task is not restricted to early design but is equally relevant to all subsequent design stages (Daly *et al.* 2016). As the transition is made from abstract conceptual ideas to arranging physical components, a phase also referred to as

Embodiment Design (Pahl & Beitz 2007), the ill-structured nature of the task remains. In the often-called configuration problem, i.e., finding the optimal arrangement of components (Wielinga & Schreiber 1997), there is usually substantial freedom in designing components and their interfaces, with design iterations often leading to changes of part dependencies, the overall product layout or even a complete conceptual (re)design. Consequently, the number of potential design ideas and task complexity increase as solutions get increasingly detailed (Liu, Chakrabarti & Bligh 2003). Moreover, the design process is path-dependent, and upstream decisions will define the direction of subsequent design tasks. The ability to make changes thus decreases as the design matures, while early decisions must be made to reach the maturity necessary for a meaningful evaluation (Cross 2004; Weber 2014).

This situation presents a key challenge in design: objectively determine at an early stage of development how "good" the end product will be. At any stage of product development, the "best" design is usually only identifiable through comparison with others, with respect to (wrt) explicit or tacit criteria, whether the comparison is qualitative or quantitative (Pugh 1990; Simon 1996; Papalambros & Wilde 2017). As design details are added, the complexity of such comparison increases further. For example, the growing number of technical details and constraints will lead to dependencies and trade-offs, as we go from concept to initial embodiment, in particular when adding manufacturing, cost reduction and other life-cycle requirements (Arthur 1993; de Weck *et al.* 2011). In turn, these trade-offs have a drastic influence on end product performance (Altshuller 1984; Suh 1998), time to market (Wynn & Eckert 2017), robustness (Göhler 2017) and complexity (Frey *et al.* 2007).

Given the ill-structured and largely iterative nature of design, a rigorous approach for managing trade-offs across all stages of development is difficult. Instead, design synthesis and the realisation of abstract design functions into an initial physical solution oftentimes rely on heuristic approaches. These include qualitative design guidelines (Altshuller 1984; Suh 1998), heuristics aimed at specific, at times tacit, design objectives (French 1985; Pahl & Beitz 2007; Skakoon 2008), or advice on avoiding contradictions between initial design objectives to recognise essential needs and avoid compromising on 'pragmatic' solutions (Altshuller 1984). However, many of these approaches tend to focus on early design rather than on the detailed technical constraints that are relevant for the ill-structured configuration design task (Pahl & Beitz 2007; Sillitto 2009; Ullman 2017). Also, design guidelines that specifically address the embodiment stage were largely developed through analysis of existing designs (i.e., identifying "good" or "bad" solutions) or observations of design practices (Fu, Yang & Wood 2016; Reimlinger *et al.* 2020). Correspondingly, they tend to be contextual, given their extraction from specific development tasks and an often limited number of design objectives. And while an increased use of AI and digital twins is a promising direction to expand contexts, the question of how to balance different, also tacit, objectives when embodying a physical system is still an important challenge for many design engineers.

As the parametric exploration of complex design spaces remains computationally expensive (Georgiades *et al.* 2019), particularly in earlier, less-constrained design stages, most embodiment practices consequently rely on a limited number of candidate designs in order to then map and analyse potentially critical dependencies. Examples for corresponding approaches include the use of Design Structure Matrices (Eisenbart *et al.* 2017; Eppinger & Browning 2018; Chouinard, Achiche & Baron 2019) or enhanced Function-Means Tree representations (Mokhtarian, Coatanéa & Paris 2017; Müller *et al.* 2019).

At the other end of the design process, the design optimisation literature is rife with trade-off analysis techniques once a concept has been finalised and is being fine-tuned (Purshouse & Fleming 2003; Marler & Arora 2004; Papalambros & Wilde 2017). However, employing formal optimisation in early design is limited by our ability to model the relevant design decisions. In conceptual and early embodiment design, substantial design freedom and the ill-definiteness of the design problem generally drive our inability to model all potential configurations under a unified optimisation model (Papalambros & Shea 2001; Papalambros & Wilde 2017). While multiobjective design optimisation can help describe trade-offs quantitatively, the extant trade-off analysis techniques are, as a result, largely preoccupied with computation, visualisation and comparison of Pareto sets, rather than questioning *why* a trade-off exists in the first place. Instead, the focus lies on modelling of preferences (Das 1999; Purshouse & Fleming 2003; Kelly *et al.* 2011), measuring distances to a utopia point (Marler & Arora 2004), scaling methods for objective weighting (Athan & Papalambros 1996b; Kasprzak & Lewis 2001) and strategies for making trade-offs aggressively or conservatively (Otto & Antonsson 1991). Substantial work also exists for sensitivity, robustness (Gunawan & Azarm 2005), uncertainty (Mattson & Messac 2005), visualisation (Fonseca & Fleming 1998), dimensional reduction (Unal, Warn & Simpson 2016) and identification of competing objectives in a n-dimensional objective space (Purshouse & Fleming 2003).

Thus, there seems to be a gap between the creative mapping of ideas in conceptual and early embodiment design and the parametric exploration and optimisation of (usually few) candidate configurations by quantitative means. Progress in the field of computational synthesis and the use of machine learning and deep learning algorithms is promising towards bridging this gap, see, e.g., Regenwetter, Nobari & Ahmed (2022); Chakrabarti *et al.* (2011), or Cunningham *et al.* (2020); Regenwetter *et al.* (2022); Ghasemi *et al.* (2024); Wang *et al.* (2025). However, old challenges in exploring the design space (Cagan *et al.* 2005) seem to remain, e.g., for the exploration of layouts in load-carrying structures (Gamache *et al.* 2023).

With the exception of certain topology optimisation problems, e.g., in the field of lightweight structural design (Lyu & Saitou 2005; Regenwetter *et al.* 2022), the question of what a "good" design implies consequently remains a largely human, hence subjective, task. Experienced designers seem to rely more on their expertise rather than on systematic analysis (Kleinmuntz 1990; Ahmed, Wallace & Blessing 2003), producing better designs than novices (Cross 2004; Reimlinger *et al.* 2020). This fact has been attributed in part to a better understanding of and ability to manage trade-offs (Ahmed *et al.* 2003).

This article aims to provide a new perspective in using optimisation thinking for engineering design synthesis by extending recent work on Multi-Objective Monotonicity Analysis (MOMA) (Sigurdarson *et al.* 2022a,b). The presented approach does not address all challenges mentioned above and it requires engaging the human designer; it does show, though, how classical optimisation thinking can be used further upstream in the design process than we typically think.

In the remainder, a summary of the aims of the presented work and the relevant background is provided. Some theoretical background is included for completeness.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

The concept of an *ideal design* is introduced as a helpful guide to systematic synthesis tasks. Guidelines for synthesising an ideal design are then proposed. A demonstration example from practice related to drug delivery device design is presented. Some concluding remarks complete the article.

2. Aim of this work

In the above spirit, this article addresses the question of how to synthesise a "good" design, and suggests the notion of an *ideal design*. This notion is based on the aforementioned work on MOMA. Sigurdarson *et al.* (2022a) provide a theoretical framework for studying the causality of trade-offs in an existing design proposal quantitatively by focusing on the dependencies that exist among competing design objectives and the constraints that are active (bounding or tight) at the various optima. These dependencies, which in some cases are unique to the optimal set, create trade-offs and hence define the achievable performance of the final product. The resulting knowledge has applications in configuration redesign, i.e., for eliminating or reducing existing trade-offs, as shown in Sigurdarson *et al.* (2022b).

The present article posits that the idea of reactive configuration redesign principles transfers positively to earlier design phases, i.e., that the same logic can be applied proactively in product design synthesis, where conceptual and embodiment design are tightly linked. In line with the basic ideas developed in earlier work, e.g., Cagan & Agogino (1987); Jain & Agogino (1990); Deb & Srinivasan (2006); Curtis, Hancock & Mattson (2013), the underlying objective is to widen the applicability of formal optimisation knowledge beyond identifying the optimal solution to an already defined mathematical problem. In the sense of a procedural rationality in design (Simon 1996), we propose that the optimisation paradigm and trade-off knowledge can be used to inform the typically qualitative reasoning patterns employed by designers, and to identify design decisions that will yield a "better" optimisation problem – i.e., concepts and embodiments that will likely have a "good" optimum despite the uncertainty involved.

In the following, this approach to synthesis is achieved by:

(i) introducing the notion of the *ideal design*, and recognising that we can aim to approach this ideal but never actually reach it. The ideal design – a solution free of unintended trade-offs and unaffected by avoidable constraints – is a construct used to derive the mathematical foundation for describing what *good design* actually entails.
(ii) using this foundation to identify and classify design principles that can be selectively combined to guide the creation of new embodiment designs.
(iii) providing example cases that illustrate that the suggested principles are consistent with the decision-making of experienced designers.

We argue that the presented approach extends existing design synthesis methods and tools by relying on a mathematical foundation for seeking an ideal design, which we define formally using three conditions. Instead of relying on analysis and mitigation of trade-offs in an existing design, as in Sigurdarson *et al.* (2022a, 2022b), we use these conditions to derive strategies for the systematic avoidance, mitigation and reduction of trade-offs during design synthesis.

This work also helps to structure existing, largely context-specific design guidelines (e.g., specific to certain pre-defined objectives) and to bridge the gap

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

between early-stage exploration of conceptual solutions and their late-stage optimised embodiment. In doing so, we deliberately abstain from the idea that all trade-offs can, or should be, addressed computationally. Instead, the article adopts the usual understanding of ideation processes and employs the notion that *ideal design synthesis* is merely a means of supporting the continuous ideation of solutions that promise a "good" optimum based on the systematic consideration of trade-offs. While the product's evolution may make previous decisions and analyses obsolete, developing a set of concepts as well as configurations will still increase the likelihood of creating a better product.

3. Theoretical background

This section offers the theoretical background, terminology and symbols necessary for the later discussion on design optimisation. It includes an overview of monotonicity analysis (Papalambros & Wilde 2017), and its extension to multiobjective trade-offs in configuration design (Sigurdarson *et al.* 2022a,b). These are foundational to the proposed notion of an *ideal design* and the question of how trade-off mitigation strategies described in the earlier work extend to early-stage design synthesis.

3.1. Trade-offs and pareto sets in design

Multiobjective design optimisation problems are stated in negative-null form (Papalambros & Wilde 2017) as:

$$\min \quad \mathbf{f}(\mathbf{x}) \tag{1}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{h}(\mathbf{x}) = 0 \tag{3}$$

$$\mathbf{x} \in \mathcal{X} \tag{4}$$

where $\mathbf{f}(\mathbf{x})$ is a vector of k objective functions f_i, $i = [1, 2, \ldots, k]^T$ to be minimised, $\mathbf{x}$ is a vector of real-valued design variables, $\mathbf{h}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$ are the equality and inequality constraints respectively and $\mathcal{X}$ is the set constraint that may include additional restrictions besides those of Eq. (2) and (3). The attainable set $\mathcal{A}$ contains all feasible values of $\mathbf{f}(\mathbf{x})$. A point $\mathbf{f}(\mathbf{x}^*) \in \mathcal{A}$ is said to be Pareto-optimal if and only if there exists no other point $\mathbf{f}(\mathbf{x}) \in \mathcal{A}$ such that $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*) \wedge f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$.

The Pareto set C containing all Pareto-optimal points lies on the boundary of $\mathcal{A}$ facing the origin, hence it is also referred to as the Pareto frontier. The utopia point F^0 is a k-dimensional point consisting of all the single-objective minima and lying outside $\mathcal{A}$. The Pareto set contains an infinity of optimal designs corresponding to different trade-offs and proximity to the utopia point is often used as a preference criterion: the closer a Pareto point is to the utopia point, the "better" it is.

Proximity to the utopia point is central to our pursuance of an ideal design described in the next section. This proximity depends on the location and shape of the Pareto set, as noted in Sigurdarson *et al.* (2022a). Specifically,

1. A *trade-off variable* causes global dependencies, i.e., a variable x shared by two objectives, $f_1(x)$ and $f_2(x)$, causes a trade-off if $\arg\min f_1(x) \neq \arg\min f_2(x)$.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

This can occur only if the objectives are either oppositely monotonic, or when one or both are non-monotonic wrt x.
2. An *active constraint* causes regional or local dependencies, i.e., an active constraint (one that "hits" its bound at the optimum) reduces the degrees of freedom (DOF), affects the feasible domains for the remaining DOF, and changes the optimum.

These concepts are treated more formally below. Central to the arguments in the present work is the premise that knowledge of trade-off variables and constraint activity is just as important in early-stage design as it is in the usual late-stage (embodiment) optimisation. Such knowledge constitutes the basis for the notion of ideal design. Essentially, the premise is that all design problems have a Pareto set, and the better the Pareto set, the better the design, whether we can model it or not!

3.2. Multiobjective monotonicity analysis

In formal optimisation, Monotonicity Analysis (MA) leverages monotonic behaviour in objective and constraint functions to check for model boundedness and constraint activity prior to any computation (Papalambros & Wilde 2017). A scalar function is monotonically increasing with respect to a variable x, if it holds that $f(x_2) > f(x_1)$ for any $x_2 > x_1$, denoted as $f(x^+)$, and is said to be monotonically decreasing wrt x, if it holds that $f(x_2) < f(x_1)$, denoted as $f(x^-)$, resulting in the following MA principles:

First Monotonicity Principle (MP1): In a well-constrained minimisation problem, every increasing variable is bounded below by at least one non-increasing active constraint.

Second Monotonicity Principle (MP2): In a well-constrained minimisation problem, every nonobjective variable is bounded both below by at least one non-increasing semi-active constraint and above by at least one non-decreasing semi-active constraint.

Active constraints can be used to eliminate a design variable and thus reduce the problem's DOF. This process of model reduction reveals relationships that *necessarily* exist at the optimum as a consequence of the constraint activity. Thus, such constraint activity is a necessary but not sufficient condition for optimality.

Multiobjective Monotonicity Analysis (MOMA) (Sigurdarson *et al.* 2022a) extends MA to the multiobjective problems in Equation (5). It is convenient to employ the well-known *upper-bound formulation*, see, e.g., Marler & Arora (2004), to study the Pareto set (Papalambros & Wilde 1978; Sigurdarson *et al.* 2022a). This formulation creates a scalar substitute problem by selecting one objective function $f(\mathbf{x})$ from the vector $\mathbf{f}(\mathbf{x})$ as *the* objective and a vector of (upper) bounds ϵ to constrain the remaining objectives,

$$\min \quad f(\mathbf{x}) \tag{5}$$

$$\text{s.j.t} \quad \mathbf{c}(\mathbf{x};\epsilon) \leq 0 \tag{6}$$

$$\mathbf{g}(\mathbf{x}) \leq 0 \tag{7}$$

$$\mathbf{h}(\mathbf{x}) = 0 \tag{8}$$

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

$$\mathbf{x} \in \mathcal{X}, \epsilon \in \mathcal{R}^{k-1}, \tag{9}$$

where $\mathbf{c}$ is a $k-1$ dimensional vector of *(upper) bound objectives* expressed as $c_i(\mathbf{x};\epsilon_i) = f_{i+1}(\mathbf{x}) - \epsilon_i \leq 0$ or $c_i(\mathbf{x};\epsilon_i) = \epsilon_i - f_{i+1}(\mathbf{x}) \leq 0$, $i = [1,2,..,(k-1)]$; ϵ is a vector of parameters ϵ_i in the real space $\mathcal{R}^{k-1}$ for the bound objectives. When $f(\mathbf{x})$ is minimised for given values of ϵ_i, then the solution $\mathbf{x}^*$ is Pareto optimal if all of the bound objectives are active with non-zero Lagrange multipliers. Pareto points are thus identified by varying ϵ between lower and upper limits $\epsilon_{\mathbf{L}}$ and $\epsilon_{\mathbf{U}}$.

In well-bounded problems (where the feasible domain is a compact set), MOMA can reduce multiple objectives simultaneously to reveal the degrees of freedom remaining in the Pareto set and the constraints that shape it. Following Sigurdarson *et al.* (2022a), we classify *trade-off* or *harmonious* variables: If an objective pair f and c_i has a variable x_1 in common, but differ in monotonicity wrt x_1, e.g., $f(x_1^+)$ and $c_i(x_1^-)$, then x_1 is said to be a trade-off variable, denoted $\overline{\underline{x_1}}$. Correspondingly, an objective pair of like monotonicity wrt a common variable indicates that the variable is harmonious, either monotonically decreasing $\overline{\overline{x}}$ or monotonically increasing $\underline{\underline{x}}$, and can be used to partially minimise both objectives simultaneously. The basic insight in Sigurdarson *et al.* (2022a) is that in the presence of monotonic trade-off variables, no dominant minimum exists, resulting in a Pareto set.

As an illustration, consider a symbolic optimisation problem with three design variables, x_1, x_2 and x_3, which have monotonic relationships with two objective functions, $f_1(x_1^+, x_2^-, x_3^+)$ and $f_2(x_1^-, x_2^-, x_3^+)$, and three inequality constraint functions, $g_1(x_1^+, x_2^-)$, $g_2(x_1^-, x_2^+, x_3^+)$ and $g_3(x_1^-, x_2^+, x_3^-)$.

Inserting this into a symbolic Monotonicity Table (MT) (Papalambros & Wilde 2017) yields the initial problem overview shown in MT1 in Table 1. Following MP1, MP2 and the theorems developed in Sigurdarson *et al.* (2022a), it is clear that x_1 is a trade-off variable, as f_1 and f_2 are oppositely monotonic wrt x_1. This relationship means that we cannot minimise f_1 by reducing x_1, without increasing f_2. Further, both objectives are monotonically increasing wrt x_3. This implies that the optimal value, i.e., in standard negative-null form, the minimum, is determined by the greatest lower bound $x_3^* = \underline{x_3}$.

Given that a design point is feasible so long as all inequality constraints equations yield non-positive values $g_3(x_1^-, x_2^+, x_3^-) \leq 0$, we can, subsequently, use this knowledge to solve g_3 for x_3 and eliminate x_3 from the problem, i.e., by back-substituting a term that is monotonically decreasing wrt x_1 and increasing wrt x_2 into the remaining expressions. If we further assume that this term might be so strongly dependent on x_2 that its back-substitution into f_2 causes a change in the monotonicity of f_2 wrt x_2, this, in turn, makes x_2 a trade-off variable as f_1 and f_2 are now oppositely monotonic. Such situations are common in model reduction based on monotonicity analysis. In multiobjective problems, this means that trade-offs between design objectives can be introduced or made worse by active constraints, which create dependencies unique to the optimal set.

Beyond helping reveal dependencies between design objectives, MA consequently also reveals important knowledge about how constraints affect a given design problem. Implicitly, this means that a variable with an *increasing* influence on a constraint will restrict the design space as its smallest feasible value (lower bound) increases. Similarly, for the upper bound of decreasing variables. From the designers' perspective, we want to make design decisions that pose large upper

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

Table 1. How monotonicity analysis can reveal the shared design variables and active constraints that cause trade-offs between design objectives

MT1: *Initial Problem*

	f_1	f_2	g_1	g_2	g_3
x_1	+	-	+	-	-
x_2	-	-	-	+	+
x_3	+	+		+	-

$\longrightarrow$

MT2: *Reduced Problem*

	f_1	f_2	g_1	g_2
x_1	+	-	+	-
x_2	-	+	-	+

bounds on the decreasing variables that affect active constraints and small lower bounds on the increasing variables.

Following such arguments, MOMA helps to identify conditions under which the bound objectives are active, i.e., the values of ϵ that affect the feasible domain of $\mathbf{x}$. In turn, this allows reduction of multiobjective problems to reveal dependencies that create the Pareto set, providing valuable information for targeted design changes after an initially chosen design configuration has been optimised.

3.3. Theory of systematic design improvement

Pareto set dependency analysis can be used to identify configuration design improvements (Sigurdarson *et al.* 2022b). To define design improvement rigorously, we use the concept of the *meta-Pareto optimality* (Athan & Papalambros 1996a), which involves comparison of different solutions to a given design problem:

Meta-Pareto Set: Given Pareto sets $C_1, C_2, \ldots, C_p$ for p configuration solutions for a given design problem, the meta-Pareto set $\breve{C}$ consists of points within the union of these sets, $C_{\mathcal{U}} = C_1 \cup C_2, \cup \ldots \cup C_p$, that are Pareto-optimal with respect to the set $\breve{C}$. A point $\mathbf{f}_*$ is meta-Pareto-optimal if and only if there exists no point $\mathbf{f} \in C_U$ such that $f_i \le f_i*$ for all i and that $f_i < f_i*$ for at least one i.

Meta-Pareto sets allow for a comparison of Pareto points from different optimisation models, so long as they involve the same objectives. Assuming that the design changes made in an attempt to reach an improved configuration do not result in new or changed objectives, we introduce the following definition:

Design Improvement Criterion: If a configuration with Pareto set C_0 is redesigned, resulting in a new Pareto set C_1, the redesign is said to be an improvement, if and only if the meta-Pareto set of C_0 and C_1 is identical to C_1, namely, $\breve{C} = C_1$ irrespective of the weights assigned to the objectives, which implies that all of the Pareto points of the original design are at least weakly dominated by the Pareto points of the redesign.

The definition implies that the achievable performance in the new design is at least equal to or better than that of the previous design, wrt all criteria, exemplified in Figure 1. This formal definition is independent of the design context and the relative importance of the objectives, and it uses quantifiable properties we can employ in deriving mathematically based redesign principles, as presented in Sigurdarson *et al.* (2022b). If the meta-Pareto set consists of points from both designs, the redesign is potentially an improvement, depending on the relative weighting of the objectives involved. Since optimality is defined only in the context

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

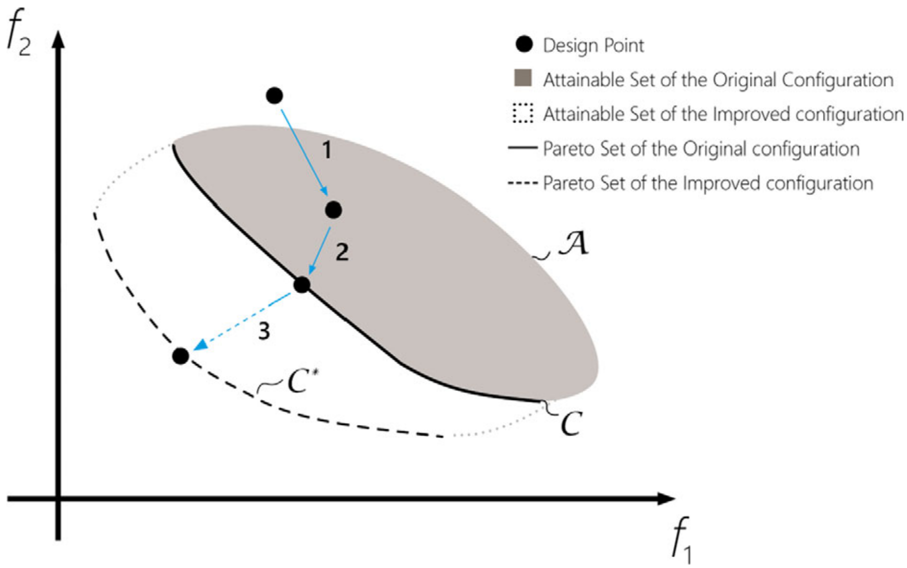

Figure 1. The embodiment design and initial dimensioning phase is characterised by inherently different activities (illustrated by the blue lines moving between design points) (1) identification of feasible design points, (2) optimisation of the initial configuration towards a Pareto-optimal design point and (3) changing the initial towards an improved configuration, a new Pareto-optimal design respectively (adapted from Sigurdarson *et al.* (2022b)).

of the particular optimisation model (Papalambros & Wilde 2017), such comparisons must use models of similar fidelity.

Under this definition, the redesigned configuration(s) can still involve changed objective functions and constraints, so long as the set of objectives itself remains the same, i.e., allowing for a comparison in the same k-dimensional objective space. Beyond this, two theorems are useful in the present discussion:

Theorem 1. Existence of the Pareto set: If no trade-off variables exist (globally or regionally) after back-substitution of active constraints, then the optimum is a point $F*$, rather than a set. Therefore, a Pareto set cannot exist without trade-off variables.

Theorem 2. Position of the Pareto Set: Harmonious variables affect the position of the Pareto set C relative to the origin. Thus, design changes that widen their feasible domains in an improving direction, yield a new strongly dominant Pareto set, $C_{i+1} < C_i$.

In Sigurdarson *et al.* (2022b), these theorems, along with a set of proofs and corollaries, were used to identify different modes of reactive design change that can be implemented post-analysis, i.e., on the basis of the multiobjective monotonicity analysis. In short, it was found that from a mathematical perspective, an improved Pareto set can only be achieved through parametric design change (which in practice often involves improved materials or production processes), or through a specific set of model transformations. These model transformations were then translated into targeted design changes, which, when applied systematically, result in a new, strongly or weakly dominant Pareto Set $C_{i+1} \leq C_i$. The transformations

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

either involve the reduction/elimination of dependencies that drive trade-offs, reducing the impact of the constraints that shape the Pareto set the most, or eliminating regional/local dependencies that worsen the trade-off between objective pairs.

Modes of change relating to dependencies relevant to the present article include *separate trade-off variables* (e.g., resulting in the substitution of $\underline{\overline{x_i}}$ in one objective with an independent variable), *flip monotonicity* of $\underline{\overline{x_i}}$ in one objective and *scaling trade-offs* by introducing an independent variable to reduce the negative influence of $\underline{\overline{x_i}}$ in any of the objectives. Correspondingly, design changes relating to constraints include *leverage harmonious variables*, which involves introducing changes that relax the active constraints of shared variables that do not contribute to trade-offs in order to widen the feasible domain in an improving direction.

4. The ideal design

While Sigurdarson *et al.* (2022a,b) focus on configuration redesign, the present article examines how we can extend this work to conceptual design and initial embodiment synthesis. More specifically, we consider the question: What decisions can we make in early-stage synthesis to reduce the likelihood of trade-offs and to identify solutions that promise the best (proportional) optima in the end product? To answer, we will pose some conjectures and the conditions for an ideal design. In the subsequent section, we will provide guidelines for ideal design synthesis.

4.1. Conjectures and conditions of ideal design

To define decisions that yield the best optimal solution, we must first clarify what we strive for in design synthesis. The term “best optimal” here is not a casual pleonasm; multiple optima are compared based on how close they are to the ideal design. The properties of the ideal design and hence the means for comparing design options and selecting the best are the key topic of this section.

While methods discussed earlier prescribe various notions of “good” design, a description in a mathematical manner is difficult. From an optimisation perspective, what constitutes a “good” design is explicit, even if still subjective. Design is naturally multiobjective (Isaksson & Eckert 2020). Some objectives are well known, explicitly stated from an early stage, and are quantifiable. Others can be tacit, qualitative or subjective. While researchers typically leave it up to the designers’ intuition to decide the relative importance of objectives for selecting one Pareto point, designers in practice will struggle to do this, with biases, fixations and marketing directives that affect their decision-making.

Be that as it may, trade-offs exist, and some end products perform better than others, whether we are able to model the objective functions or not. It is well established that designers design with the optimum in mind: opportunism (French 1992; Cross 2004), trade-off knowledge (Ahmed *et al.* 2003) and *a priori* understanding of constraint activity (Onarheim 2012; Eckert & Stacey 2014) are key indicators of an experienced designer who will generally be more successful in early-stage design (Cross 2004).

In general terms, and drawing from the mathematical background, we, therefore, assert that the ideal design has the following attributes: (i) involves no trade-

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

offs, (ii) has the best performance with respect to the desired solution attributes and (iii) has low complexity. In optimisation terms, the ideal design has a single optimum rather than a set, positioned as close to utopia as possible, and defined by as few design variables as possible (as a measure of complexity).

We now formalise these concepts for an optimisation model stated in negative-null form. We introduce three conjectures that describe the hypothetical ideal designers strive for. From these conjectures, three corresponding conditions for ideal design arise. The conjectures are put forth in a hierarchy, in that the second conjecture is put forth assuming the first is true and the third is put forth assuming the two preceding conjectures are true.

Conjecture 1. First Conjecture of Ideal Design Synthesis.
In the ideal design, $\operatorname{argmin} f_i(\mathbf{x}) = \operatorname{argmin} f_j(\mathbf{x})$ *for any pair of design objectives, i and j,* $i \neq j$, *meaning no trade-offs exist.*

Recall from the Pareto set existence theorem that the Pareto set cannot exist if there are no trade-off variables in the problem (globally or regionally) after the back-substitution of all active constraints. Thus, we can state the following condition for the First Conjecture:

Condition 1. Avoidance of trade-off variables.
For a design to be ideal, it cannot contain trade-off variables, meaning $x_i \notin \underline{\overline{\mathbf{x}}}$, *for any variable i.*

By definition, trade-off variables have oppositely monotonic relationships with two or more objectives, either globally (meaning the variable is monotonic) or regionally (meaning the variable is non-monotonic). From Condition 1, it follows that the ideal design involves objectives that are dependent only on monotonic variables, or on non-monotonic variables that are not shared with other objectives, or on non-monotonic variables that by chance have the same value at the minimum of each objective. If no trade-offs exist, we can move on to the question of the optimum of each objective:

Conjecture 2. Second Conjecture of Ideal Design Synthesis.
In the ideal design, $f_i^* \to 0$ *or* $f_i^* \to -\infty$ *for any design objective i.*

Were it not for the First Conjecture and Condition 1, this conjecture would, on its own, simply imply that the ideal Pareto set is infinite. From the Pareto position theorem we know that harmonious variables and their bounds in part determine the position of the Pareto set. If Condition 1 is fulfilled, the location of the optimum is determined only by constraints and the existence of interior optima (which implies non-monotonicity). Hence, we can state the following condition for the Second Conjecture:

Condition 2. Boundedness in the Improving Direction.
For a design to be ideal, the bounds of its design variables must be infinite or asymptotic in the improving direction, meaning $\overline{x_i} \to \infty$ *for* $\mathbf{f}\left(x_i^-\right)$ *and* $\underline{x_j} \to 0 \vee -\infty$ *for* $\mathbf{f}\left(x_j^+\right)$ *for any i and j.*

As a consequence of the activity theorem of constrained optimisation (Papalambros & Wilde 2017), the optimum would never reach 0 or $-\infty$ if this

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

condition is not fulfilled. Assuming both conditions are fulfilled, we can state the third and final conjecture:

Conjecture 3. Third Conjecture of Ideal Design Synthesis.
The ideal design has as few design variables as possible, meaning $\dim \mathbf{x} \to 1$.

Given the stated prerequisite that all design problems are multiobjective, it follows that $\dim \mathbf{f} \geq 2$. From this, a condition arises, without which the Third Conjecture would result in trade-offs:

Condition 3. Simplicity through Harmonious Variables.
For a design to be ideal, all design variables must be harmonious, meaning $\operatorname{argmin} \mathbf{f}(\mathbf{x}) = \overline{\overline{\mathbf{x}}} \vee \underline{\underline{\mathbf{x}}}$.

With these conjectures and conditions, we have a mathematical basis for defining the ideal design as our interpretation of "good" design. This definition is consistent with the prior work on configuration redesign, but it goes beyond design improvement, as it does not involve comparison with preexisting designs and applies to synthesis rather than just redesign.

4.2. Seeking the ideal design

From a practical perspective, the above theoretical background and derived conjectures and conditions might seem overly formal. Obviously, no functional intent can be realised with a single design variable, just as trade-off variables can never be completely avoided. From a mathematical perspective, an optimal design problem with an infinite or asymptotically bounded feasible domain is poorly bounded with no convergent solution. The conjectures and conditions of ideal design are put forth to support the design synthesis process not obviate it. So, in real designs, we will never fulfil the conditions described: they are merely a construct we aspire to. The ideal design is literally *ideal.*

At the same time, the closer we can come to fulfilling the conjectures and conditions during synthesis, the closer we follow the underlying optimisation paradigm, which, in turn, will increase the likelihood of a good design outcome. We can summarise the synthesis implications of the three conditions as follows.

> *Condition 1 implies that dependencies are not a problem, so long as they do not introduce trade-offs between any of the context-relevant design objectives (recalling the importance of context for embodiment).*
>
> *Condition 2 implies that the bounds of harmonious and independent variables can have a substantial impact on the location of the optimum and, thus, how close the conceptual or embodiment design is to the ideal.*
>
> *Condition 3 implies that harmonious variables allow the realisation of more product functionality with less complexity. If we can avoid trade-offs and overly restrictive constraints without introducing new design variables, or when removing variables, the design will be closer to the ideal.*

In sum, in seeking the ideal design, we look for a few trade-off variables, a wide feasible domain, and a few design variables. We use a mathematical framework to collect guidelines for early-stage design. The more trade-offs we can avoid through targeted conceptual and embodiment decisions, the closer the design is to the ideal.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

Furthermore, the notion of an ideal design highlights the relevance of systematically considering engineering constraints when moving from conceptual to initial embodiment decisions. If we can arrange parts and features in a way that widens the feasible domain in the improving direction, or if we can integrate solutions based on monotonicity information, thus without affecting trade-offs, we can avoid following broad aphorisms such as the claim that all dependencies are inherently bad. From a pragmatic angle, some of the conditions might be better fulfilled through decisions beyond the narrow product, e.g., widening the feasible domain or reducing complexity through new materials, manufacturing processes or production facilities.

5. Design guidelines for ideal design synthesis

There are several basic mathematical model transformations that can be applied to improve a configuration by identifying limitations through analysis. However, identifying trade-off variables, harmonious variables, trade-off inducing constraints and parasitic influences after detailed analysis is reactive, and we can only apply the methods introduced by Sigurdarson *et al.* (2022b) post-analysis of an existing design.

The question is then, how one can codify the modes of reasoning one could employ in early design synthesis to avoid dependencies that cause or worsen trade-offs, unnecessarily restrictive constraints and also typical mistakes. For this purpose, we rely on the conditions derived earlier to identify and classify the decisions one could make (opportunistically) in concept design and embodiment design to get as close to the ideal design as possible.

In this section, we offer design guidelines extracted from the ideal design conditions. The intention is not to present mathematics but to collect useful practices implied by or emerging from the mathematically defined conditions. Once the guidelines are identified, the mathematics are no longer necessary.

Drawing from the heuristics available in literature and the earlier discussion on systematic design improvement, we put forward a collection of design guidelines G corresponding to the three presented conditions:

Condition 1: $G^{\overline{x}}$ *guidelines for trade-off mitigation*

Condition 2: G^{χ} *guidelines to help improve the feasible domain*

Condition 3: $G^{\dim(\mathbf{x})}$ *guidelines for design integration*

The guidelines are classified depending on the design activity or mode of reasoning they relate to. Each class of guidelines include an introduction to contextualise them in the broader design process. Examples of classes under $\mathbf{G}^{\overline{x}}$ *guidelines for trade-off mitigation* include guidelines for the *selection of working principles* $\mathbf{G}_1^{\overline{x}}$ or for *synthesising embodiment structures with trade-off variables in mind* $\mathbf{G}_2^{\overline{x}}$. Novel guidelines that expand the ideal design construct and the redesign principles derived in Sigurdarson *et al.* (2022b) into design synthesis decisions are in some classes complemented by existing guidelines. Hence, this classification also underlines how existing heuristics are (contextually) consistent with the conditions of ideal design.

This is not an exhaustive set of guidelines, rather a set of opportunistic heuristics that translate the ideal design concept into specific design changes.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

In contrast to existing lists of largely context-specific design heuristics, the guidelines are not tied to specific design objectives. Instead, they shift the focus towards a generally applicable design strategy as we seek to reach the ideal design. The guidelines do assume the designer has taken some preliminary steps in order to apply them. These steps are as follows:

1. Define overall design objectives based on desired functionality.
2. Map out potential working principles and structures.
3. For each working principle, determine what constraints arise and if monotonic relationships exist between design objectives.
4. Identify potential trade-offs and active constraints caused by relationships.

Steps 1 and 2 are commonly taken in any design process. Step 3, less so, but most designers would intuitively identify basic monotonic relationships from very early on, even at their first sketches, and certainly at the selection of working principles. Design "variables" here do not necessarily reflect dimensions on a drawing or formal variables in an optimisation model; they might be more abstract and represent the designer's overall understanding of how the working principles and configuration of parts influence the objectives, something like "the larger the output from subsystem *A*, the less mechanical efficiency we can achieve using working principle *X* in subsystem *B*."

5.1. Condition 1 – Guidelines for trade-off avoidance

The mitigation of trade-offs through independence between design objectives is a common recommendation in the engineering design literature, e.g., Suh (1998), Pahl & Beitz (2007) and Skakoon (2008). Yet, as shown and discussed by Sigurdarson *et al.* (2022a) there are other routes towards avoiding trade-offs or reducing their influence, which might be preferable to independence in many contexts. If we expand the redesign principles derived in Sigurdarson *et al.* (2022b)), such as *separate*, *flip monotonicity* and *scale*, into decisions such as selection of working principles, synthesis of working structure and the resulting preliminary

$\mathbf{G}_1^{\overline{x}}$: Select working principles with trade-off variables in mind.

Early decisions can yield the most challenging trade-offs. With each of the different working principles that can be used to embody a function, different dependencies and corresponding trade-offs ensue. It follows that certain working principles are more suited to avoiding trade-offs between certain sets of design objectives. Considering this aspect while selecting working principles might lead to the avoidance of detrimental dependencies or aid in the invention or identification of new working principles:

- $\mathbf{G}_{1.1}^{\overline{x}}$ To inform synthesis, systematically identify and prioritise the key design objectives that emerge from business goals and user needs.
- $\mathbf{G}_{1.2}^{\overline{x}}$ Identify working principles that support the fulfilment of said objectives.
- $\mathbf{G}_{1.3}^{\overline{x}}$ Avoid working principles that do not allow for simultaneous improvement of all objectives they affect, i.e., that clearly introduce trade-off variables $\overline{x}$ into the design problem.
- $\mathbf{G}_{1.4}^{\overline{x}}$ Introduce additional, compensating functionality/working principles for reducing trade-offs between design objectives, in case that independence or like-monotonicity are not possible.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

$G^{\bar{x}}_{2}$: Synthesise embodiments with trade-off variables in mind.

In combining working principles into an overall solution, dependencies arise due to parts/variables contributing to several functionalities (and, therefore, objectives) simultaneously. Synthesis of this preliminary embodiment creates further trade-offs, as it gives rise to design constraints that may introduce trade-off variables when active. Even scant monotonicity information and identification of potential dependencies may guide this process towards avoiding trade-off variables:

- $\mathbf{G^{\bar{x}}_{2.1}}$ Systematically assess monotonicity while exploring combinations of working principles into different system layouts. Compare and select based on avoiding as many potential trade-off variables as possible.
- $\mathbf{G^{\bar{x}}_{2.2}}$ If a potential trade-off variable becomes evident, redistribute functionality among the parts, subsystems or functional elements in the system, or rearrange the parts themselves to achieve independence or a scaling of the trade-off variable.
- $\mathbf{G^{\bar{x}}_{2.3}}$ Avoid creating geometric dependencies between variables with oppositely monotonic influence on design objectives, e.g., positioning a geometric feature that needs to be as large as possible inside a feature that needs to be as small as possible.

$G^{\bar{x}}_{3}$: Avoid common drivers of trade-offs

Poor design decisions in combining working principles into preliminary embodiment can create avoidable trade-offs. Many context-specific examples of such can be found in existing heuristics in the engineering design literature; see, e.g., French (1985) and Skakoon (2008).

- $\mathbf{G^{\bar{x}}_{3.1}}$ Avoid making design objectives interdependent through equilibria.
- $\mathbf{G^{\bar{x}}_{3.2}}$ Avoid temporal conflicts, e.g., a part ideally being infinitely stiff for optimal performance in one system state and infinitely soft in another (Altshuller 1984). Redistribute functionality or introduce new parts to mitigate such scenarios.
- $\mathbf{G^{\bar{x}}_{3.2}}$ Avoid force loops that overlap unnecessarily, especially if these work in the opposite direction (French 1985).
- $\mathbf{G^{\bar{x}}_{3.4}}$ Avoid unbalanced and asymmetric loads, unless they are required to fulfil a given function (French 1985; Pahl & Beitz 2007)

$G^{\bar{x}}_{4}$: Be Pragmatic

Trade-offs are not always worthwhile avoiding – some will occur due to inherently conflicting objectives (e.g., low mass vs high stiffness), others exist between objectives on vastly different orders of importance, while other again will be better addressed through design beyond the product itself. This should be considered in synthesis and redesign, as accepting these situations might open new opportunities:

- $\mathbf{G^{\bar{x}}_{4.1}}$ Accept the existence of a trade-off variable if the relative importance of the two objectives is vastly different or if the loss in utility caused by the existence of the dependency is negligible.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

$\mathbf{G}^{\bar{\mathbf{x}}}_{4.2}$ The existence of a trade-off variable might be acceptable if most of the objectives involved are of like monotonicity w.r.t the variable.

$\mathbf{G}^{\bar{\mathbf{x}}}_{4.3}$ If a constraint seems to be difficult to fulfil, treat it as an objective in order to explore whether trade-off variables between it and the existing objectives can be avoided.

embodiment, a set of synthesis guidelines emerge. These apply to any trade-off variable and any set of objectives, whether they exhibit globally monotonic behaviour or not. Some might seem entirely obvious, but this merely reflects the importance of considering potential contributors to trade-offs upfront.

5.2. Condition 2 – Guidelines to maximise the feasible domain

As the active constraints in a design problem can influence the location of the Pareto set, they can render trade-offs unimportant if the location of the Pareto set is such that the trade-offs lead to little loss of utility. Hence, synthesising mechanical systems with bounds in mind can have a substantial effect on the performance of the end product. Considering constraints while selecting working principles, and systematically arranging parts and geometric features based on the objectives to avoid creating overly restrictive constraints, may allow the widening of the feasible domain in an improving direction.

$\mathbf{G}^{\chi}_{1}$: Consider inherent constraints when selecting working principles

Some constraints are inherent to specific working principles. Rather than being caused by decisions made in regards to the configuration and shape of parts, they stem from the underlying physics involved or unavoidable practical limitations such as manufacture or assembly. For instance, in designing a suspension system, the constraints that arise from the selection of a pneumatic solution (e.g., seal integrity and radial piston fit) are vastly different from those involved in mechanical springs (e.g., shear stress, fatigue, spring index limits). Hence, selecting working principles can drastically influence feasible domains; both of the design variables that arise with the specific principle (e.g., a piston diameter in the suspension example) and those that exist in the system irrespective of what principle is selected (e.g., suspension mounting points).

$\mathbf{G}^{\chi}_{1.1}$ When possible, select working principles that avoid constraints that are not inherent to the design problem itself.

$\mathbf{G}^{\chi}_{1.2}$ When possible, rely on the principles of self-help (Pahl & Beitz 2007) to eliminate constraints related to mechanical failure.

$\mathbf{G}^{\chi}_{1.3}$ Consider introducing new functionality to eliminate active constraints, e.g., overload protection, active damping, designing to allow in-use adjustment or maintenance, etc.

$\mathbf{G}^{\chi}_{1.4}$ If possible, change designs to make active constraints dependent on additional *decreasing* variables (or to remove increasing variables), widening the feasible domain, e.g., re-arranging components to maximise load-bearing surface for a critical load case.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

G_2^χ: Widen the feasible domain through configuration

The relative arrangement of parts and geometric features in an assembly has a substantial effect on what constraints are imposed on the proportional optimisation problem. How the whole system fits together creates fit constraints (e.g., a part fitting inside another), tolerance chains and force paths/loops. Hence, knowledge of monotonic relationships between the design objectives and the key dimension(s) of a component should be used to support the identification of the ideal system layout.

- $\mathbf{G}_{2.1}^\chi$ Use the monotonicity of a system's harmonious variables to layer and spatially configure its parts, i.e., moving decreasing variables outward and increasing ones inward in the assembly.
- $\mathbf{G}_{2.2}^\chi$ Layer components from inside to out based on their influence on the objectives; the most influential decreasing variable furthest out, and the most influential increasing variable furthest in.
- $\mathbf{G}_{2.3}^\chi$ If a part contains increasing and decreasing variables that are geometrically interdependent, split it into two or re-allocate functionality to other parts.
- $\mathbf{G}_{2.4}^\chi$ Arrange components and interfaces to take advantage of scaling/gearing effects. For instance, a rule of thumb is to locate surfaces that control the position of parts or are loaded in the assembly location that allows the largest possible dimension, while locating rotating components as far inward as possible.

G_3^χ: Design towards hitting hard constraints.

While many constraints can be manipulated through design, e.g., eliminating contributions to a tolerance chain or increasing the achievable load-bearing area of a snap feature by moving it to another location in the assembly – other constraints are hard and unaffected by a change in configuration. Designing towards these constraints becoming active, rather than the feasible domain being defined by (ultimately) avoidable constraints, widens the feasible domain as much as possible in the improving direction.

- $\mathbf{G}_{3.1}^\chi$ In the ideal design, all harmonious variables are determined by their general limits (irrespective of context), rather than a specific limit determined by the manner in which the functional intent has been realised.
- $\mathbf{G}_{3.2}^\chi$ If a variable is bound by a hard constraint that cannot be manipulated through configuration design change, explore changes to the overall concept or potential for parametric change (e.g., a change in the production process, material selection, etc.).

Knowing a priori which constraints are active is challenging, especially when it comes to variables that are involved in several nonlinear phenomena. This should not prevent the designer from attempting to design around constraints that are *likely* going to be active. For instance, if we wish to minimise the mass of a system, we expect that stress constraints and manufacturing constraints on wall thickness will likely be active. If we are interested in minimising size, the geometric fits between components and the capabilities of the manufacturing processes will

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

G_4^χ: Manage parametric contributions caused by active constraints

Oftentimes, constraints will include parameters (e.g., properties related to the material and production process), which cannot directly be manipulated by the designer. Yet, their influence and importance can still be considered in the process of synthesis and redesign:

- **$G_{4.1}^\chi$** When possible, relax active constraints through design rather than parametric change. Parameters can almost never be adjusted freely and often indirectly represent design decisions beyond the designer's direct control (e.g., allowable cycle time in an assembly step, sourceable material grades, etc.). As a general rule, it is preferable to widen the feasible domain through design changes.
- **$G_{4.2}^\chi$** Avoid letting features necessitated by constraints affect the feasible domains of harmonious variables.

definitely come into play. The designer does not need to know which constraints are active if the design can be manipulated to affect several potentially active constraints at once. The following guidelines apply to all harmonious and independent monotonic variables and to non-monotonic variables that are bound at the optimum.

5.3. Condition 3 – Design integration to reduce complexity

The more harmonious variables we can achieve, the less complex the system will be. Achieving low complexity in synthesis or redesign involves avoiding redundant variables and increasing the number of objectives that the remaining variables contribute to. Such an increase in design integration implies that each component in the assembly affects more functionality (Matthiassen 1997). Design integration

$G_1^{\dim(\mathbf{x})}$: Integrate functionality with trade-offs/constraints in mind

From a synthesis perspective, integration may involve designing components that are involved in the embodiment of multiple working principles. In redesign, increasing integration might involve change such as combining parts, introducing new geometric features to existing parts and adding a state-change to the system. If care is not taken, one can easily end up making decisions that introduce new contributors to trade-offs or worsen the proportional optimum. Hence, the following design guidelines may apply:

- **$G_{1.1}^{\dim(\mathbf{x})}$** Whenever possible, integrate functionality that results in harmonious variables or the elimination of a constraint without the introduction of a trade-off variable.
- **$G_{1.2}^{\dim(\mathbf{x})}$** Integrate additional functionality as long as it does not shift bounds substantially in the non-improving direction.
- **$G_{1.3}^{\dim(\mathbf{x})}$** If variables/parts can be eliminated through the redistribution of functionality in a manner that does not introduce trade-off variables or new constraints, these variables/parts are redundant.
- **$G_{1.4}^{\dim(\mathbf{x})}$** Integrate whenever multiple functions can be performed over the same axis of operation (e.g., rotation around a given axis), so long as this does not introduce non-scalable trade-off variables, overly restrictive bounds, or result in an overconstrained mechanism.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

$\mathbf{G}_{1.5}^{\dim(\mathbf{x})}$ Design towards achieving *state changes.* As a rule of thumb, the more kinematic state changes (e.g., parts changing interfaces or kinematic degrees of freedom, and load paths being redirected) a designer is able to build into a mechanical system, the more functions and objectives each part can contribute to. This does not necessarily create trade-off variables or necessitate additional design variables, given that independence is achieved *in time* rather than geometry. Hence, this is somewhat analogous to the *Separate in Time* heuristic from TRIZ.

$G_2^{\dim(\mathbf{x})}$: Separate to avoid trade-off variables or inherent constraints

Oftentimes, separation becomes the only recourse, as some forms of functionality cannot be integrated without creating trade-off variables that cannot be scaled or inherent constraints that cannot be relaxed. TRIZ (Altshuller 1984) contains an expansive treatment on different approaches to separation, so the following guidelines are only stated in the specific context of a designer trying to get as close as possible to fulfilling the Conditions of Ideal Design:

$\mathbf{G}_{2.1}^{\dim(\mathbf{x})}$ Avoid integrating *physically contradicting* (Altshuller 1984) functionality in the same parts/subsystem – e.g., requiring a part to be stiff and compliant, insulating yet conductive, etc.

$\mathbf{G}_{2.2}^{\dim(\mathbf{x})}$ Split parts or introduce new ones and redistribute functionality, if the alternative is an active constraint or a trade-off variable that cannot be scaled.

$\mathbf{G}_{2.3}^{\dim(\mathbf{x})}$ Only modularise and parallelise the system when the alternative is a trade-off or a substantially narrowed feasible domain in the improving direction. This will often be the case in products that are maintenance-heavy or in architectures with a high degree of part re-use, where increased integration might lead to increased cost.

$G_3^{\dim(\mathbf{x})}$: Consider the hierarchy of trade-off avoidance

From Condition 3, it is preferable to avoid trade-offs through design decisions that do not introduce new design variables. Otherwise, we would be mitigating trade-offs by increasing complexity. Hence, to fulfil Conditions 1 and 3 simultaneously, there is an order of preference as to how to eliminate or reduce a trade-off:

$\mathbf{G}_{3.1}^{\dim(\mathbf{x})}$ Flip monotonicity over all else – attempt to achieve like monotonicity in the selection and combination of working principles into a system structure.

$\mathbf{G}_{3.2}^{\dim(\mathbf{x})}$ Eliminate a trade-off variable by removing its influence on one objective entirely. This especially applies to *unnecessary* influences (see $G_4^{\dim(\mathbf{x})}$).

$\mathbf{G}_{3.3}^{\dim(\mathbf{x})}$ Separate the trade-off variable by redistributing functionality to existing geometry/design variables/parts.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

$\mathbf{G}_{3.4}^{\dim(\mathbf{x})}$ Separate the trade-off variable by introducing new design variables/features onto existing geometry/design variables/parts.
$\mathbf{G}_{3.5}^{\dim(\mathbf{x})}$ Scale the trade-off variable using existing variables. This can, for instance be achieved by relaxing the constraints on variables that act as a multiplier/divisor to the trade-off variable.
$\mathbf{G}_{3.6}^{\dim(\mathbf{x})}$ Separate the trade-off variable by introducing new parts/subsystems
$\mathbf{G}_{3.7}^{\dim(\mathbf{x})}$ Scale the trade-off variable by introducing new parts/subsystems

$\mathbf{G}_{4}^{\dim(\mathbf{x})}$: Avoid unnecessary influences

Trade-offs are caused by dependency. Some forms of dependency are not inherent to the concept or embodiment but arise unintentionally due to what can be viewed as noise. These situations should be avoided whenever possible:

$\mathbf{G}_{4.1}^{\dim(\mathbf{x})}$ Aim for kinematically correct designs, as static indeterminacy leads to non-linearities and dependencies (Ebro 2016).
$\mathbf{G}_{4.2}^{\dim(\mathbf{x})}$ Avoid the associated/parasitic loads that arise from asymmetric parts and load paths, or from unbalanced moments (French 1985).
$\mathbf{G}_{4.3}^{\dim(\mathbf{x})}$ Isolate negatively interacting subsystems from each other to avoid, e.g., unintended friction, heating, vibration or competing working directions (Torry-Smith, Mortensen & Achiche 2014; Chouinard *et al.* 2019).
$\mathbf{G}_{4.4}^{\dim(\mathbf{x})}$ To limit cost versus performance trade-offs, avoid designing geometric features required for manufacturing and assembly in a manner that influences functionality.

implies lower "structural" complexity (i.e., number of design variables), lower manufacturing cost (Suh 1998; Ulrich, Eppinger & Yang 2020) and increased robustness (Matthiassen 1997). In a study of part counts in jet engines, Frey *et al.* (2007) also found that complexity reduction can, in some cases, result in improved system performance, despite an increase in dependency.

Increasing integration, however, may create new problems through the introduction of trade-off variables or new or more restrictive constraints. Hence, the following design guidelines aim to support the convergence towards fulfilling Condition 3 without moving further away from fulfilling the other conditions.

6. Example: Design of a medical injection device

To illustrate the application of the guidelines and their consistency with (tacit) engineering design practices, we will use a medical device, the FlexTouch pen injector. It is designed for the injection of active pharmaceutical ingredients (APIs) in liquid formulation, such as insulin, human growth hormone and glucagon-like-peptide receptor agonists (GLP-1RA).

In the present study, the device is used to demonstrate how the prescribed guidelines can be used a-posteriori to explain the underlying reasoning behind decisions made in concept and embodiment design. Having been on the market for

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

more than a decade, the FlexTouch was designed long before this study, and is not used as an example to claim validation of the ideal design construct to support design thinking. Neither is the intent to claim that the FlexTouch is an ideal design or to assess how it approaches the ideal. Instead, the aim is to exemplify how important modes of reasoning that experienced mechanical designers (perhaps unconsciously) employ in synthesis of new concepts and embodiments correspond to the suggested conjectures and conditions. Being a highly integrated mechanical system, the device represents a good example of many trade-off decisions mechanical designers deal with across a range of mechanical systems.

6.1. Functionality and embodiment

The FlexTouch injection pen is a disposable needle-based injection device with a set amount of API designed to *autodose*. The user selects a certain dose by turning the dial that is rotationally coupled to the scale, and injects it via a sterile needle replaced by the user before each dose. The device is discarded once empty. In selecting the desired dose with the dial (Figure 2), the user winds up a torque spring inside the device, and the set dose is shown on the scale. The spring drives the dosing mechanism, rotating a lead screw through a stationary nut, thereby pushing a plunger through a cartridge filled with an API in liquid form (Figures 2 and 3). After inserting the needle and activating the dosing mechanism with the button at the end of the device, the user injects the API into their subcutis (a tissue layer under the surface of the skin). Upon pressing the button, the torque spring mechanism is released from a rotational lock, thereby turning the lead screw. Several ratchet mechanisms create a clicking sound and haptic feedback to indicate dose setting, dose progress and dose delivery completion.

Given the high accuracy requirements, substantial production volume and safety-critical nature of the product, the FlexTouch is embodied with as few components as possible, mostly made of polymers. This ensures low cost in high volume production, few tolerances that affect dose accuracy, and high reliability. However, this approach also means that each individual component contributes to numerous sub-functions, resulting in a highly interdependent design with many trade-offs to consider during development. The device took over 6 years to develop from initial sketch to running production, largely due to this interdependence. In retrospect, the final FlexTouch design reflects the many targeted decisions and iterations made in the development process to mitigate trade-offs, widen the feasible domain, and allow a low part count, relative to a set of key design objectives that are not specific to the embodiment design of FlexTouch, but rather to injection devices as a whole.

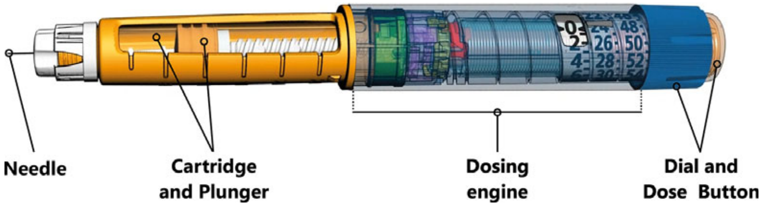

Figure 2. The core functional elements of the FlexTouch injection device design.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

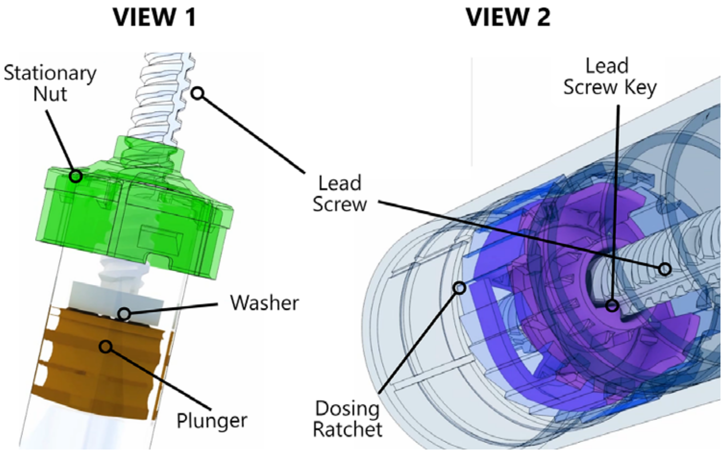

Figure 3. The selection of a lead screw rather than, e.g., a rack and pinion allows low friction without elongating or widening the device. The introduction of the purple component in View 2, with the torque transferring key on the smallest possible diameter, and the positional control on the largest diameter (via the dosing ratchet), almost eliminates the trade-off between mechanical efficiency and dosing accuracy.

Relevant Design Objectives: Minimise device diameter and length, maximise mechanical efficiency (in turn increasing dosing speed), maximise dose accuracy.

Design Guidelines Involved: $\mathrm{G}^{\overline{\mathrm{x}}}_{1.1}$, $\mathrm{G}^{\overline{\mathrm{x}}}_{1.2}$, $\mathrm{G}^{\overline{\mathrm{x}}}_{1.3}$, $\mathrm{G}^{\overline{\mathrm{x}}}_{1.4}$, $\mathrm{G}^{\overline{\mathrm{x}}}_{2.1}$, $\mathrm{G}^{\overline{\mathrm{x}}}_{4.2}$, $\mathrm{G}^{\chi}_{2.1}$ and$\mathrm{G}^{\chi}_{2.2}$.

6.2. Seeking Ideality in Synthesis – Working Principle Selection Towards Condition 1

A key decision in the development of the FlexTouch is that the dosing engine works in rotation and converts it into a linear movement, relying on a lead screw mechanism to push the piston in the drug cartridge (Figure 3). The FlexTouch is an integrated product, and the selection of a working principle has implications for all functionality in the device. As such, this decision was made at the very beginning of the conceptual design iteration that ultimately led to the final product.

As it is difficult to qualify the characteristics of *good* design decisions without a comparator, we can look to helical rack and pinion mechanisms, which are a typical alternative to lead screws when converting rotation to linear movement with high accuracy, while allowing for gearing. As opposed to other mechanisms, e.g., cams, crank and sliders, or scotch yoke mechanisms, a rack & pinion allows for a similar form factor to a lead screw, and is hence practical for comparison.

In the following, we combine knowledge from basic machine elements theory with symbolic monotonicity analysis to demonstrate why a lead screw is indeed closer to being ideal, and to exemplify the likely reasoning behind this decision. In doing so, we see how designers employ contextual knowledge to (tacitly) apply a logic similar to that of the guidelines prescribed in this article, and how this actually results in solutions that are closer to fulfilling the conditions of ideal design. For the sake of readability, we will use a design-focused variable and objective syntax, without deviating from the negative-null form, throughout the remainder of this example.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

6.2.1. Avoiding contributors to a size versus efficiency trade-off

Compared to rack and pinion, the use of a lead screw driven by a torque spring avoids several trade-off variables between objectives that are universal to injection devices, such as max. mechanical efficiency (η, treated as $-\eta$ in negative-null form), min. dosing inaccuracy (ϵ_d) and min. device size (diameter, D_D, and length, L_D), while also avoiding multiple otherwise restrictive constraints.

If we first look at the question of efficiency versus size, we see that the pitch diameter, d_p is of significant importance (see Figure 4 for a visualisation). Evidently, $D_D\left(d_p^+\right)$, as the larger the pitch diameter, d_p, the larger the lead screw mechanism, and the wider the device diameter, D_D. A textbook screw-equation analysis reveals that d_p has a non-monotonic influence on mechanical efficiency. In practice, though, this relationship is *regionally* monotonically increasing, i.e., $-\eta\left(d_p^+\right)$ for designs with low friction material pairs and for pitch angles (a.k.a. helix angles) $\alpha \leq 45\,^\circ$. This bond is a practical one for most material pairs due to the risk of frictional self-locking for high angles, which is common for polymers. Thus, for lead screws, the lower bound $\underline{d_p}$ simultaneously optimises D_D and $-\eta$.

For a rack and pinion meanwhile, the corresponding pitch diameter of the pinion, d_p has a monotonically decreasing influence $-\eta\left(d_p^-\right)$; the larger the contact diameter d_p between pinion and rack, the smaller the tooth force, F_T, and hence the smaller the frictional loss, given that $F_T = T_{in}/d_p$. Yet, as the outer diameter of the rack and pinion mechanism still increases monotonically with the pitch diameter $D_D\left(d_p^+\right)$, the consequence is that we cannot simultaneously reduce device diameter and maximise efficiency by adjusting d_p. Hence, the selection of a

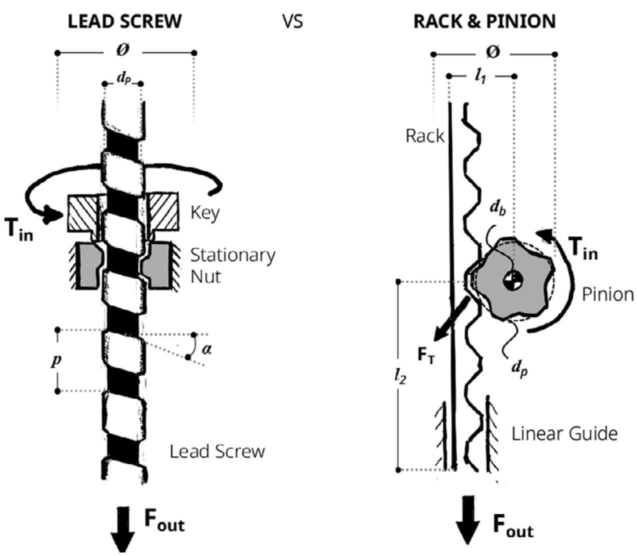

Figure 4. The key design variables in the dimensioning of two ubiquitous working principles for accurate conversion of rotation into linear motion: lead screw and rack and pinion mechanisms.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

lead screw over a rack and pinion avoids the pitch diameter becoming a trade-off variable between an objective pair, consistent with guidelines $G^{\overline{x}}_{1,1}$, $G^{\overline{x}}_{1,2}$ and $G^{\overline{x}}_{1,3}$, contributing to approaching fulfilment of Condition 1.

6.2.2. Avoidable dependencies between efficiency, size and accuracy

The aforementioned trade-off between mechanical efficiency and device diameter is worsened even further by the skew angle of F_T. In gear design, to avoid undercutting teeth, the minimum pressure angle α_{min} that determines the angle of F_T increases as the number of teeth z_n is reduced. This can be expressed via an inequality constraint:

$$g_{zmin} = 2/sin\left(\alpha_{min}\right)^2 - z_n = 2/sin\left(\alpha_{min}\right)^2 - d_p/m \leq 0 \tag{10}$$

As we wish to minimise the device diameter, and by extension d_p, we can infer that this constraint will be active, causing the number of teeth to decrease. The underlying reasoning is that the only other variable, the gear module, m (determining the tooth *overlap* between rack and pinion), is usually dimensioned to ensure the system can withstand operational loads. In effect, reducing the diameter of the pinion increases the pressure angle of F_T, as g_{zmin} becomes active, which ultimately increases friction and, in turn, worsens the trade-off between efficiency and device size. With this in mind, the selection of the lead screw over the rack and pinion is also consistent with most of the *Avoid common drivers of trade-offs* guidelines, $G^{\overline{x}}_3$.

In fact, as the pressure angle of the pinion can never be 0 (due to g_{zmin} this would require an infinite number of teeth), F_T of the rack and pinion mechanism will always have a skew angle relative to the axis of operation. If there is no geometry to balance this, the rack would slide away from the pinion with higher input torque, causing a positional error that increases dosing inaccuracy, ϵ_d. The mitigation is a linear guide positioning the rack, to balance the skew angle of F_T at the distance l_2 from the axis of rotation of the pinion (shown in Figure 4). As increasing this length helps reduce sliding friction, mechanical efficiency increases monotonically with l_2, thus $-\eta\left(l_2^-\right)$. Yet inevitably, so does the length of the device $L_D\left(l_2^+\right)$. This also contributes to rack and pinions being further from fulfilling Condition 3, than lead screws, as the necessity of the linear guide increases the number of design variables.

6.2.3. Scaling the unavoidable accuracy versus size trade-off

As we reduce the screw diameter, the accuracy of its rotational position becomes more sensitive to unavoidable noise factors, affecting dosing accuracy. This is a general problem in reducing the size of rotating components; geometric variation, e.g., due to production tolerances, affects the fit between the screw and the key, resulting in an increasingly larger angular error, the smaller the pitch diameter. Hence, as we improve the size and efficiency of the lead screw mechanism by reducing the pitch diameter d_p, we increase the dosing inaccuracy, meaning $\epsilon_d\left(d_p^-\right)$. In other words, the pitch diameter is still a trade-off variable $\overline{\underline{d_p}}$.

Yet, looking at the design, we see the designer's clear awareness of the trade-off between size and efficiency on one side, and accuracy on the other, given the

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

presence of compensating functionality, as prescribed by $G^{\bar{x}}_{1.4}$. Specifically, the rotational position of the key component (the purple geometry in Figure 3, View 2) and by extension of the Screw, is controlled through the addition of elastic ratchet arms onto the Key, with stiffness k_{arms}. These arms interface with the housing (the transparent component in Figure 3, View 2) in a dosing ratchet interface, with a diameter d_{arms}. Located on the inside of the housing, this rotationally locating interface is positioned on the largest feasible diameter on the entire device, reducing the absolute errors caused by geometric variation in the ratcheted surface, while ensuring a high dosing resolution (the number of discrete positions the mechanism can start or end at) without requiring infeasibly small geometric features.

At the same time, the ratchet arms introduce frictional losses as the arms are bent passing over each ratcheted surface and creating *click* sounds), resulting in a reaction force on a large diameter. Yet, the loss involved is significantly less than the sliding friction that would have occurred if the screw diameter corresponded to the ratcheted interface diameter, especially because the stiffness of the ratchet arms can be minimised without negatively affecting the stated objectives. Correspondingly, the ratchet diameter does not contribute to increasing the device diameter D_D, given that other components that fit around the lead screw determine said dimension.

The trade-off compensating functionality of the ratchet arms effectively allows torque transfer with low frictional loss via the lead screw key at a small diameter without sacrificing positional accuracy. This downscaling of the contribution of d_p to the trade-off increases conformance with Condition 1 and contributes to approaching Condition 2– an aspect we will return to further below. This design reasoning exemplifies the guideline $G^{\bar{x}}_{1.4}$, and also illustrates $G^{\bar{x}}_{4}$, as the designers have clearly employed a degree of pragmatism in the embodiment, accepting the slight frictional loss due to the elastic arms.

To summarise, the lead screw presents a working principle that arguably approaches the ideal, for the objectives at hand, at least when considering Condition 1 of ideal design and the associated guidelines for converging to it. The lead screw allows simultaneous optimisation of efficiency and size ($G^{\bar{x}}_{1.1}$-$G^{\bar{x}}_{1.3}$), while avoiding the introduction of unbalanced, associated loads, thereby steering around avoidable contributors to dependency ($G^{\bar{x}}_{3}$). Sliding friction in the key interface is minimised while reducing the size of the screw mechanism ($G^{\bar{x}}_{2.2}$). Alternative working principles such as rack and pinions would achieve this by increasing mechanism size and by relying on additional geometry, to the detriment of fulfilling Condition 3. The difference between the two embodiments, is summarised in the partial monotonicity tables shown in Table 2.

Finally, the designers of the mechanism have demonstrate pragmatism ($G^{\bar{x}}_{4.1}$-$G^{\bar{x}}_{4.2}$) in dealing with a trade-off that emerges when minimising the size of the mechanism, namely, that it comes at the cost of accuracy. Using elastic arms for positioning the Key component, the designers have scaled down this trade-off ($G^{\bar{x}}_{1.4}$ & $G^{\bar{x}}_{2.2}$). While letting the arms interface with the housing on a large diameter increases frictional resistance in rotation, this friction is close to an order of magnitude smaller than what would have occurred, had the pitch diameter of the lead screw d_p been increased to a level yielding equivalent accuracy.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

Table 2. Partial monotonicity table of a lead screw compared to that of a rack and pinion, of key design variables shown in Figure 4 w.r.t. device diameter (D_D), mechanical efficiency ($-\eta$), dosing inaccuracy ϵ_d and device length L_D

	D_D	$-\eta$	ϵ_d	L_D
d_p	+	(+)	-	
d_{arms}		+	-	
k_{arms}		+		

MT3: *Lead Screw*

	D_D	$-\eta$	ϵ_d	L_D	g_{zmin}
d_p	+	-	-		-
α		+	+		-
l_2		-	-	+	

MT4: *Rack & Pinion*

6.3. Seeking ideality in embodiment – managing constraints towards Condition 2

The configuration and layering of components in the FlexTouch reveals how numerous decisions have been made towards widening the feasible domain of key harmonious variables.

Following the preceding section on Condition 1, an evident instance of configuration being done based on widening the feasible domain is the location of the lead screw in the assembly. It is nested in the centre of the assembly (G_2^χ), i.e., it need not fit around additional components. Thus, its diameter can be reduced towards hitting hard constraints, e.g., production or thread geometry limits, rather than constraints resulting from configuration decisions (G_3^χ).

Besides the lead screw, there are many examples of embodiment decisions to increase the feasible domain. Consider the activation of dosing. By pushing the button, the user pushes a set of splines on a clutch component, the *Activation Splines* on the yellow part in Figure 5, out of their engagement with a spline interface in the pen housing that acts as a rotational lock. Before becoming free to rotate, this clutch engages the purple Key component (Figure 2), that is free to rotate in the dosing direction.

Prior to activation, the activation splines lock the spring mechanism against the housing, creating a closed force loop. This functionality could have been achieved in numerous ways, but it has been specifically located on the widest possible internal diameter of the device. Again, it would seem that the designers have striven for the ideal when locating this interface. From the user's perspective, a small device diameter is preferable, as is a low activation force. Pushing a button with a high force can cause considerable pain, since this is done after the needle has been inserted.

Placing the clutch splines in the outer-most layer of the device, the designers have achieved the largest possible contact diameter. The activation force is primarily driven by the need to overcome frictional resistance when pushing the spline interface out of engagement with the housing. This friction stems from the clutch splines withholding the torque spring. Absorbing a given torque at a large contact diameter results in a small tangential force (as $T = F_t \cdot d/2$)), and therefore low friction. Low tangential force means the mechanical stress in the device is low; therefore, less material is used and ultimately a smaller device. A large contact diameter also allows a high resolution of activation splines to carry the load, as

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

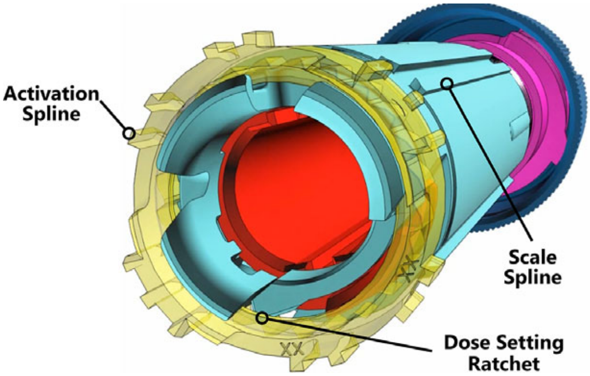

Figure 5. *Beneficial layering*: The location of the activation splines in the FlexTouch is beneficial for several reasons. The torque spring is mounted between the teal component and the red component.
Relevant Design Objectives: Minimise device diameter, minimise activation force, maximise dose accuracy.
Relevant Inequality Constraints: Interface stress in the activation spline, tangential assembly clearance in the spline interface, feature size (i.e., moulding injection pressure).
Design Guidelines Involved: $G^{\chi}_{2.1}$ and $G^{\chi}_{2.2}$.

there is a lower limit to how small features can be manufactured. Combined with lead-in surfaces for re-engagement that lower the angular error, this high resolution minimises the dosing inaccuracy. Lastly, the large diameter also scales down the influence of geometric variations in manufacturing the spline surfaces, noting that a predefined absolute spline width or position tolerance has less influence on the angular error, the larger the spline diameter is.

In summary, the designers were aware of a potentially harmonious relationship and used this awareness to configure the device, specifically to widen the feasible domain of the load bearing area of the clutch splines, while allowing the contact force in these splines to be geared down as much as possible. The reasoning at play here is the search for solutions that approach Condition 2, through decisions consistent with guidelines $G^{\chi}_{2.1} - G^{\chi}_{2.2}$, in regard to simultaneous improvement of device size, dosing accuracy and activation force.

6.4. Seeking ideality in complexity reduction towards Condition 3

The desire to reduce complexity, conforming with Condition 3, can be seen in almost all design decisions in the FlexTouch. Complexity reduction is especially driven by the large production volume required to service millions of users in the global population of people with diabetes and other chronic conditions requiring regular injections.

Primary among these is the integration of individual functionalities and working principles in a manner where almost all components contribute to all states of use and sequences of events. The FlexTouch dosing engine is designed to act in rotation. Dose setting, dose actuation, dose clicks, and dose scale are driven by the torque spring mechanism and the user turning the dial. The scale is rotationally locked to the spring mechanism via a set of splines (see Figures 5 and 6) and is mounted on a

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

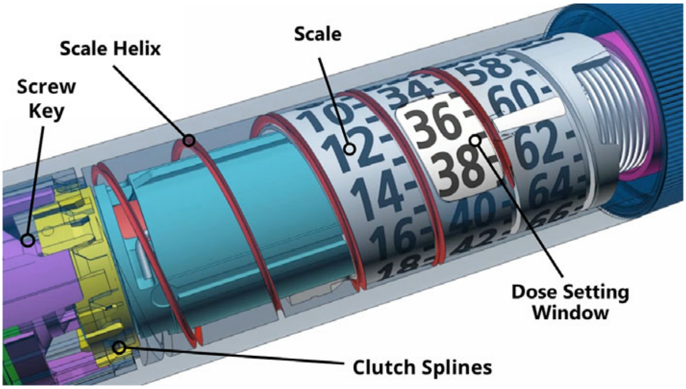

Figure 6. *Beneficial Integration.* By driving much functionality in the dosing engine using a single torque spring, and using working principles for sub-functions which rely on rotation, the FlexTouch avoids numerous contributors to the trade-offs between synchronisation and angular accuracy on one side and mechanical efficiency on the other.
Relevant Design Objectives: Minimise device diameter, scale display error, maximise dosing speed and scale number size and resolution,
Design Guidelines Involved: $G_{1.1}^{\dim(\mathbf{x})}$, $G_{1.4}^{\dim(\mathbf{x})}$, $G_{2.1}^{\chi}$,$G_{2.2}^{\chi}$.

helix inside the housing, meaning that it is screwed back and forward as the user sets a dose and the device auto-doses. Besides communicating the dose setting, the scale also acts as a rotational lock, preventing the device from being dialled above the maximum dose setting and below its zero setting.

From a single-objective perspective, there might have been benefits in designing single modules to perform these functions independently, e.g., getting louder clicks or larger or finer resolution in a modular structure. Yet, substantial benefits arise from functionality integration. A less integrated design would have resulted in a larger and less accurate device, as all the individual modules would have to interface somehow through intermediary components/interfaces. This results in longer force loops and longer tolerance chains, causing avoidable dependencies to the detriment of Condition 1, but increasing the total number of components to the detriment of Condition 3.

At its lowest dose setting of, say, one standard unit of insulin, the device converts the rotational movement of the dial into approximately 0.15 mm of axial movement inside the drug cartridge. The single actuator (torque spring) and the scale's rotational end-stop minimise the deviation between the dose setting on the scale and the dose delivered. Having located the max/min rotational stops on the scale also maximises the accuracy and repeatability of the device, as it is the outermost component inside the device, yielding the largest possible contact diameter. The single actuator also allows parts to be layered and minimises friction without loss of accuracy. Any error in angular position is transformed into a comparatively small error in axial displacement thanks to the lead screw. An axial spring mechanism would lengthen the device or reduce the size of the deliverable dose.

In summary, the FlexTouch is an integrated product, not a modular one. Every functionality is designed around a single axis of operation, with a single actuator working in rotation, driving both the primary dosing functionality, but also visual,

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

audible and haptic feedback, relying on state changes to allow components to contribute to multiple functionalities. This is consistent with $G_{1.1}^{dim(x)}$,$G_{1.2}^{dim(x)}$ and $G_{1.5}^{dim(x)}$. This integration increases synchronisation between functionalities, ultimately increasing precision and safety. Yet, the decisions described in relation to Conditions 1 and 2, e.g., the inclusion of the dosing ratchet, are in a sense a means of functionality redistribution and reduced integration to avoid trade-off variables between a set of key objectives, namely, size and dosing accuracy. Beyond consistency with guidelines related to Condition 1, this decision is also consistent with $G_{2.2}^{dim(x)}$ and$G_{3}^{dim(x)}$.

6.5. Goal dependency of the ideal design

It should be clear from the example that the Ideal Design is goal (objectives) dependent. Goal dependency is universal in design: we cannot qualify what *good* design is without relating it to its specific objectives. The relative objective weighting, whether a subjective preference of the designer or based on a detailed understanding of stakeholder utility, only compounds the complexity of this goal dependency.

If the objectives in the FlexTouch had been different, the ideal solution would have changed correspondingly. In problems with high loads, there are diameter-pitch constraints due to self-locking behaviour of lead screws at high and low pitch angles, just as there might be considerations in regards to vibrations and wear that render a rack & pinion solution far superior. Yet, in a disposable medical device, this is not a major concern, as the design intent is only for the screw to travel out of the device, and not allow for it to be pushed back in (which is enabled by the dosing ratchet). Given the high production volumes and safety concerns, the major concern is to achieve low production complexity, high dose accuracy, repeatability and safety. The reasoning behind stating and prescribing the conditions of ideal design in a hierarchy is also exemplified in the FlexTouch; trying to minimise its complexity is of no value if it trades off safety. Hence, striving to fulfil Condition 1 is a prerequisite for seeking design decisions and solutions that fulfil Condition 3.

7. Discussion

Successful mechanical design synthesis requires a mix of creativity, qualitative reasoning, systematic analysis and engineering judgment. As discussed in the introduction, there are different design approaches that prescribe principles and tools to achieve a “good” design. However, heuristic design approaches are often either too general or too context-specific to support synthesis in practice. Early-stage design methods will likely miss the discontinuous influence of constraints on dependencies between design objectives in complex mechanical assembliest. At the other extreme, mathematical optimisation methods tend to focus on fine-tuning an existing embodiment without challenging the decisions that have led to it, thus also neglecting the value of insights that may allow for an improved embodiment beyond parametric change.

As a consequence, design practice is still largely reliant on the context specific experience of the designers involved. One might even surmise that this is a driving

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

factor for design engineers often remaining in the same sector of industry throughout their careers (e.g. pharma, automotive, aerospace, consumer electronics, etc.).

The notion of the ideal design presented in this article seeks to address this gap. The conjectures and conditions of an ideal design provide a mathematical basis for formulating reasoning patterns in synthesis, guiding design decisions towards a "better" optimisation problem and solutions that promise a "good" optimum. By identifying characteristics that one can strive to achieve in design, the conditions of ideal design provide a foundation for initial design synthesis and evaluation. The conditions can be used to inform a given decision by supporting important modes of reasoning, e.g., "would I introduce trade-off variables by choosing a mechanism that works in rotation over one that works through linear motion?," "does the resulting load path create avoidable or detrimental constraints?," or "can I combine these parts into one without creating a trade-off?"

Hence, the notion of ideal design addresses the intrinsic difficulty in designing and analysing a concept without considering its embodiment. While this notion is especially relevant in mechanical design due to the central role of geometry, it generally applies when upstream design decisions may result in additional constraints for the subsequent design phase. Following the presented conjectures and conditions avoids neglecting constraints and relying on predefined and context-dependent design objectives; moreover, it accepts the reality of dependencies and seeks to mitigate them.

The ideas presented arguably apply to design synthesis in general and are thought to advance our understanding of good design practices. This is of particular relevance given the rapid technological progress and increasingly multi-disciplinary product development tasks. Instead of reusing and incrementally improving long-used solutions in many traditional industries, e.g., as described for the automotive sector by McMahon (1994), the systematic consideration of constraint activity across domains will improve our ability to aspire for the ideal.

At the same time, the notion of an ideal design by no means contests the importance of ideation for successful design synthesis, nor the nature of design as a learning activity as highlighted by Hatchuel & Weil (2009). An important aspect in this regard, is that the ideal solution is goal dependent. The work presented in this article has no influence on whether the designer even designs towards the *right* goals, just as the ideal design solutions might remain, by definition, partially unknown until all relevant goals are identified. Hence, new knowledge might make previous decisions, which were aimed at approaching *an* ideal, obsolete.

While derived from a mathematical basis, the conjectures and conditions of ideal design are put forth to support this creative learning process, whether early stage or gradually moving towards detailed engineering tasks. Not all trade-offs can or should be dealt with up front. But, early-stage design decisions are key for converging towards a solution with as few trade-offs as possible, a wide feasible domain, and few trade-off variables – hence a high likelihood of a good optimum.

The design guidelines for ideal design synthesis are neither exhaustive nor are they on a level of specificity that ensures that it would always be obvious to the designer how to get close to the ideal design through synthesis. Also, the analysis required for their application may be occasionally onerous – particularly for designers expecting to do everything computationally. As with the general use of optimisation in practical design situations, the required effort – hence cost – must

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

match the expected benefits. The mathematical foundation of the guidelines may provide some added comfort in making these early design decisions.

This last point is also underlined by how the guidelines can be used a posteriori to explain decisions made in the concept and embodiment design of the case shown in the example section. As demonstrated through the symbolic monotonicity analysis of the FlexTouch example, it seems we can indeed show a link between the context-specific design reasoning employed by designers in design synthesis, the conditions of ideal design and many of the suggested design guidelines. With this retrospective study example, we have not demonstrated the empirical validity of either the guidelines or the conditions, but we hopefully provide some insight into how the guidelines can work in practice.

8. Conclusion

We proposed the notion of an ideal design as just that: an ideal to aspire to even when knowing that perfection is out of reach. We can seek this ideal through targeted avoidance of dependencies that cause trade-offs and restrictive constraints, while striving to minimise product complexity. We presented conditions and conjectures of ideal design and a corresponding collation of design guidelines that can help fulfil these conditions.

The guidelines stem from formal proofs and the mathematical extensions of monotonicity analysis into multiobjective problems as treated in Sigurdarson *et al.* (2022a, 2022b). Just as with monotonicity analysis in general, the approach in this article is derived from a rigorous mathematical basis but remains opportunistic in nature, perhaps more so than monotonicity analysis as it is oriented towards design rather than analysis. We may not be able to use the ideal design guidelines, but if we do, a high quality design is a likely outcome.

Further, the ideal design synthesis framework presents a new perspective on what *good design* entails, which is distinct from existing prescriptive design frameworks such as axiomatic design or TRiZ. Namely, it relies on formal proofs that emerge directly from analysis of the shape and location of the Pareto set, accounts for the vast difference between how constraints and objectives affect design problems, the discontinuous influence of active constraints upon dependency, all the while deliberately abstaining from the perspective that all dependencies can or indeed should be avoided in design practice. Given that the conditions involve quantifiable characteristics, it is possible that they might be employed in computational design studies to measure and compare how good each designed alternative is.

References

Ahmed, S., **Wallace, K. M.** & **Blessing, L. T.** 2003 Understanding the differences between how novice and experienced designers approach design tasks. *Research in Engineering Design* **14** (1), 1–11; doi:10.1007/s00163-002-0023-z.

Altshuller, G. 1984 *Creativity as an Exact Science*. Gordon and Breach; doi:10.1201/9781466593442.

Arthur, W. B. 1993 Why do things become more complex? *Scientific American* **268** (5), 144–144; doi:10.1038/scientificamerican0593-144.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

Athan, T. W. & **Papalambros, P. Y.** 1996a A quasi-Monte Carlo method for multicriteria design optimization. *Engineering Optimization* **27** (3), 177–198; doi:10.1080/03052159608941405.

Athan, T. W. & **Papalambros, P. Y.** 1996b A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization* **27** (2), 155–176; doi:10.1080/03052159608941404.

Cagan, J. & **Agogino, A. M.** 1987 Innovative design of mechanical structures from first principles. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* **1** (3), 169–189; doi:10.1017/S0890060400000275.

Cagan, J., **Campbell, M. I.**, **Finger, S.** & **Tomiyama, T.** 2005 A framework for computational design synthesis: Model and applications. *Journal of Computing and Information Science in Engineering* **5** (3), 171–181; doi: 10.1115/1.2013289.

Chakrabarti, A., **Shea, K.**, **Stone, R.**, **Cagan, J.**, **Campbell, M.**, **Hernandez, N. V.** & **Wood, K. L.** 2011 Computer-based design synthesis research: An overview. *Journal of Computing and Information Science in Engineering* **11** (2); doi:10.1115/1.3593409.

Chouinard, U., **Achiche, S.** & **Baron, L.** 2019 Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory. *Mechatronics*, 140–153; doi:10.1016/j.mechatronics.2019.03.009.

Cross, N. 2004 Expertise in design: An overview. *Design Studies* **25** (5), 427–441; doi:10.1016/j.destud.2004.06.002.

Cunningham, J. D., **Shu, D.**, **Simpson, T. W.** & **Tucker, C. S.** 2020 A sparsity preserving genetic algorithm for extracting diverse functional 3d designs from deep generative neural networks. *Design Science* **6**, e11; doi:10.1017/dsj.2020.9.

Curtis, S. K., **Hancock, B. J.** & **Mattson, C. A.** 2013 Usage scenarios for design space exploration with a dynamic multiobjective optimization formulation. *Research in Engineering Design* **24** (4), 395–409; doi:10.1007/s00163-013-0158-0.

Daly, S. R., **Seifert, C. M.**, **Yilmaz, S.** & **Gonzalez, R.** 2016 Comparing ideation techniques for beginning designers. *Journal of Mechanical Design* **138** (10), 08; doi:10.1115/1.4034087.

Das, I. 1999 A preference ordering among various Pareto optimal alternatives. *Structural Optimization* **18**, 30–35; doi:10.1007/BF01210689.

de Weck, O. L., **Roos, D.**, **Magee, C. L.** & **Vest, C. M.** 2011 *Life-Cycle Properties of Engineering Systems: The Ilities*, pp. 65–96.

Deb, K. & **Srinivasan, A.** 2006 Innovization: Innovating design principles through optimization. *GECCO 2006 – Genetic and Evolutionary Computation Conference* **2**, 1629–1636.

Design Council 2007 Eleven lessons: Managing design in eleven global companies. Technical Report 272099.

Ebro, M. 2016 *Applying robust design in an industrial context*. PhD Thesis, Technical University of Denmark.

Eckert, C. M. & **Stacey, M. K.** 2014 Constraints and conditions: Drivers for design processes. In *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations* (ed. A. Chakrabarti & L. T. M. Blessing), pp. 395–415. Springer London. doi:10.1007/978-1-4471-6338-1{_}19.

Eisenbart, B., **Gericke, K.**, **Blessing, L. T.** & **McAloone, T. C.** 2017 A DSM-based framework for integrated function modelling: Concept, application and evaluation. *Research in Engineering Design* **28** (1), 25–51; doi:10.1007/s00163-016-0228-1.

Eppinger, S. D. & **Browning, T. R.** 2018 Design structure matrix methods and applications; doi:10.7551/mitpress/8896.001.0001.

Fonseca, C. M. & **Fleming, P. J.** 1998 Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*; doi:10.1109/3468.650319.

French, M. J. 1985 *Conceptual Design for Engineers*; doi:10.1007/978-3-662-11364-6.

French, M. J. 1992 The opportunistic route and the role of design principles. *Research in Engineering Design* **4** (3), 185–190; doi:10.1007/BF01607946.

Frey, D., **Palladino, J.**, **Sullivan, J.** & **Atherton, M.** 2007 Part count and design of robust systems. *Systems Engineering* **10** (3), 203–221; doi: 10.1002/sys.20071.

Fu, K. K., **Yang, M. C.** & **Wood, K. L.** 2016 Design principles: Literature review, analysis, and future directions. *Journal of Mechanical Design, Transactions of the ASME* **138** (10), 1–13; doi:10.1115/1.4034105.

Gamache, J.-F., **Vadean, A.**, **Capo, M.**, **Rochefort-Beaudoin, T.**, **Dodane, N.** & **Achiche, S.** 2023 Complexity-driven layout exploration for aircraft structures. *Design Science* **9**, e13; doi:10.1017/dsj.2023.12.

Georgiades, A., **Sharma, S.**, **Kipouros, T.** & **Savill, M.** 2019 ADOPT: An Augmented Set-Based Design Framework with Optimisation *(Vol.* **5***)*; doi: 10.1017/dsj.2019.1.

Ghasemi, P., **Yuan, C.**, **Marion, T.** & **Moghaddam, M.** 2024 Dcg-Gan: Design concept generation with generative adversarial networks. *Design Science* **10**, e14; doi:10.1017/dsj.2024.13.

Göhler, S. M. 2017 *Metric-driven robust design – robustness quantification of complex engineering systems*. PhD Thesis, Technical University of Denmark.

Gunawan, S. & **Azarm, S.** 2005 Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization* **29** (1), 50–60; doi: 10.1007/s00158-004-0450-8.

Hatchuel, A. & **Weil, B.** 2009 C-k design theory: An advanced formulation. *Research in Engineering Design* **19**, 181–192; doi:10.1007/s00163-008-0043-4.

Isaksson, O. & **Eckert, C.** 2020 *Product development 2040: Technologies are just as good as the designer's ability to integrate them*. Technical Report September, Design Society Report DS107.

Jain, P. & **Agogino, A. M.** 1990 Theory of design: An optimization perspective. *Mechanism and Machine Theory* **25** (3), 287–303; doi:10.1016/0094-114X(90)90030-N.

Kasprzak, E. M. & **Lewis, K. E.** 2001 Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. *Structural and Multidisciplinary Optimization* **22** (3), 208–218; doi:10.1007/s001580100138.

Kelly, J. C., **Maheut, P.**, **Petiot, J.-F.** & **Papalambros, P. Y.** 2011 Incorporating user shape preference in engineering design optimisation. *Journal of Engineering Design* **22** (9), 627–650; doi:10.1080/09544821003662601.

Kleinmuntz, B. 1990 Why we still use our heads instead of formulas: Toward an integrative approach. *Psychological Bulletin* **107** (3), 296–310; doi:10.1037/0033-2909.107.3.296.

Liu, Y.-C., **Chakrabarti, A.** & **Bligh, T.** 2003 Towards an 'ideal' approach for concept generation. *Design Studies* **24** (4), 341–355; doi:10.1016/S0142-694X(03)00003-6.

Lyu, N. & **Saitou, K.** 2005 Topology optimization of multicomponent beam structure via decomposition-based assembly synthesis. *Journal of Mechanical Design, Transactions of the ASME* **127** (2), 170–183; doi:10.1115/1.1814671.

Marler, R. T. & **Arora, J. S.** 2004 Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* **26** (6), 369–395; doi: 10.1007/s00158-003-0368-6.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press

Matthiassen, B. 1997 *Design for robustness and reliability – improving the quality conciousness in engineering design*. PhD thesis, Technical University of Denmark.

Mattson, C. A. & **Messac, A.** 2005 *Pareto frontier based concept selection under uncertainty, with visualization.* Technical report.

McMahon, C. A. 1994 Observations on modes of incremental change in design. *Journal of Engineering Design* **5** (3); doi:10.1080/09544829408907883.

Mokhtarian, H., **Coatanéa, E.** & **Paris, H.** 2017 Function modeling combined with physics-based reasoning for assessing design options and supporting innovative ideation. *AI EDAM* **31** (4), 476–500; doi:10.1017/S0890060417000403.

Müller, J. R., **Isaksson, O.**, **Landahl, J.**, **Raja, V.**, **Panarotto, M.**, **Levandowski, C.** & **Raudberget, D.** 2019 Enhanced function-means modeling supporting design space exploration. *AI EDAM* **33** (4), 502–516; doi:10.1017/S0890060419000271.

Onarheim, B. 2012 Creativity from constraints in engineering design: Lessons learned at Coloplast. *Journal of Engineering Design* **23** (4), 323–336; doi:10.1080/09544828.2011.631904.

Otto, K. N. & **Antonsson, E. K.** 1991 Trade-off strategies in engineering design. *Research in Engineering Design* **3** (2), 87–103; doi:10.1007/BF01581342.

Pahl, G. & **Beitz, W.** 2007 *Engineering Design — A Systematic Approach*, 3rd ed. doi: 10.1016/0261-3069(96)84970-3.

Papalambros, P. & **Wilde, D. J.** 1978 Global non-iterative design optimization using monotonicity analysis. *Journal of Mechanical Design, Transactions of the ASME* **78**; doi: 10.1115/1.3454115.

Papalambros, P. Y. & **Shea, K.** 2001 Creating structural configurations. In *Formal Engineering Design Synthesis*, pp. 93–125. Cambridge University Press; doi: 10.1017/CBO9780511529627.007.

Papalambros, P. Y. & **Wilde D. J.** 2017 *Principles of Optimal Design*. Cambridge University Press; doi: 10.1017/9781316451038. https://www.cambridge.org/core/product/identifier/9781316451038/type/book.

Pugh, S. 1990 *Total design: integrated methods for successful product engineering*. Addison-Wesley Pub. Co.

Purshouse, R. C. & **Fleming, P. J.** 2003 Conflict, harmony, and Independence: Relationships in evolutionary multi-criterion optimisation. In *Evolutionary Multi-Criterion Optimization* (ed. C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, and K. Deb), pp. 16–30. Springer Berlin Heidelberg.

Regenwetter, L., **Nobari, A. H.** & **Ahmed, F.** 2022 Deep generative models in engineering design: A review. *Journal of Mechanical Design* **144** (7), 071704; doi:10.1115/1.4053859.

Reimlinger, B., **Lohmeyer, Q.**, **Moryson, R.** & **Meboldt, M.** 2020 Exploring how design guidelines benefit design engineers: An international and global perspective. *Design Science* **6**, e9; doi:10.1017/dsj.2020.3.

Sigurdarson, N. S., **Eifler, T.**, **Ebro, M.** & **Papalambros, P. Y.** 2022a Multiobjective monotonicity analysis: Pareto set dependency and trade-offs causality in configuration design. *Journal of Mechanical Design* **144** (3); doi:10.1115/1.4052444.

Sigurdarson, N. S., **Eifler, T.**, **Ebro, M.** & **Papalambros, P. Y.** 2022b A novel approach to configuration redesign: Using multiobjective monotonicity analysis to Alter the Pareto-set. *Journal of Mechanical Design*, 1–19; doi:10.1115/1.4053524.

Sillitto, H. G. 2009 On systems architects and systems architecting: Some thoughts on explaining and improving the art and science of systems architecting. In *19th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2009*. doi: 10.1002/j.2334-5837.2009.tb00995.x.

Simon, H. A. 1996 *The Science of the Artificial*, 3rd ed. MIT Press.

Skakoon, J. G. 2008 *The Elements of Mechanical Design*; doi: 10.1115/1.802670.

Suh, N. P. 1998 Axiomatic design theory for systems. *Research in Engineering Design – Theory, Applications, and Concurrent Engineering* **10** (4), 189–209; doi:10.1007/s001639870001.

Torry-Smith, J. M., **Mortensen, N. H.** & **Achiche, S.** 2014 A proposal for a classification of product-related dependencies in development of mechatronic products. *Research in Engineering Design* **25** (1), 53–74; doi:10.1007/s00163-013-0161-5.

Ullman, D. G. 2017 *The Mechanical Design Process*, 6th ed. McGraw-Hill New York.

Ulrich, K., **Eppinger, S. D.** & **Yang, M. C.** 2020 *Product Design and Development*, 7th ed. McGraw Hill.

Unal, M., **Warn, G. P.** & **Simpson, T. W.** 2016 Quantifying tradeoffs to reduce the dimensionality of complex design optimization problems and expedite trade space exploration. *Structural and Multidisciplinary Optimization* **54** (2), 233–248; doi: 10.1007/s00158-015-1389-7.

Wang, Z., **Liang, X.**, **Li, M.**, **Li, S.**, **Liu, J.** & **Zheng, L.** 2025 Towards cognitive intelligence-enabled product design: The evolution, state-of-the-art, and future of ai-enabled product design. *Journal of Industrial Information Integration* **43**, 100759; doi:10.1016/j.jii.2024.100759.

Weber, C. 2014 Modelling products and product development based on characteristics and properties. In *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations* (ed. A. Chakrabarti & L. T. M. Blessing), pp. 327–352. Springer London; doi: 10.1007/978-1-4471-6338-1_16.

Wielinga, B. & **Schreiber, G.** 1997 Configuration-design problem solving. *IEEE Expert-intelligent Systems and Their Applications* **12** (2); doi:10.1109/64.585104.

Wynn, D. C. & **Eckert C. M.** 2017 *Perspectives on Iteration in Design and Development* (*Vol.* **28**). Springer London; doi: 10.1007/s00163-016-0226-3.

https://doi.org/10.1017/dsj.2025.10034 Published online by Cambridge University Press