Journal of Engineering Design

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/cjen20

Managing assumption-driven design change via margin allocation and trade-offs

Soufiane El Fassi, Xin Chen, Atif Riaz, Marin D. Guenov, Albert S.J. van Heerden & Sergio Jimeno Altelarrea

To cite this article: Soufiane El Fassi, Xin Chen, Atif Riaz, Marin D. Guenov, Albert S.J. van Heerden & Sergio Jimeno Altelarrea (11 Oct 2023): Managing assumption-driven design change via margin allocation and trade-offs, Journal of Engineering Design, DOI: 10.1080/09544828.2023.2259741

To link to this article: https://doi.org/10.1080/09544828.2023.2259741

Published online: 11 Oct 2023.

JOURNAL OF ENGINEERING DESIGN
https://doi.org/10.1080/09544828.2023.2259741

Taylor & Francis
Taylor & Francis Group

Managing assumption-driven design change via margin allocation and trade-offs

Soufiane El Fassi, Xin Chen, Atif Riaz, Marin D. Guenov, Albert S.J. van Heerden* and Sergio Jimeno Altelarrea

School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, United Kingdom

ABSTRACT

Assumptions are commonly introduced to fill gaps in knowledge during the engineering design process. However, the uncertainty inherent in these assumptions constitutes a risk that ought to be mitigated. That is, assumptions can negatively impact the system if they turn out to be invalid. Adverse effects may include system failure, violation of requirements, or budget and schedule overruns. In this paper, the relationships between assumptions and margins are made explicit, with the purpose of aiding risk mitigation, as well as accommodating future opportunities such as product evolvability. To this end, a novel assumption management framework is proposed, which consists of a taxonomy of margins, an algorithm for change absorber localisation, and an interactive approach for margin trade-off. The proposed framework is demonstrated with a conceptual aircraft design use case, which shows that the most relevant margins can be identified, given a revision of a set of assumptions. It is also demonstrated that the application of the method allowed the margins to be adjusted according to the confidence in the assumptions, while maintaining satisfaction of all design constraints, without unacceptable compromise of system performance.

ARTICLE HISTORY
Received 31 January 2023
Accepted 13 September 2023

KEYWORDS
Margin trade-off; assumption management; set-based design; aerospace engineering

1. Introduction

Uncertainty is an intrinsic factor when developing complex engineering products. This is because, at the early design stage, only limited knowledge regarding the product is available. Therefore, assumptions are inevitably made to fill knowledge gaps to enable decision-making. Crucially, these early-stage decisions will dominate the definition of the entire design solution and consequently represent a large portion of the entire product life cycle cost commitment (Mavris and DeLaurentis 2000). Furthermore, as knowledge is gained during the subsequent development stages, some of the assumptions may turn out to be invalid, which introduces a risk that critical design changes need to be made or that product performance may need to be compromised. Such change dependencies exist at different levels of abstraction in and across different domains

CONTACT Xin Chen xin.chen@cranfield.ac.uk Cranfield University, B83, Bedford MK43 0AL, United Kingdom
*Present address: James Watt School of Engineering, University of Glasgow, University Avenue, Glasgow, G12 8QQ, United Kingdom

(Koh 2017), highlighting the need for systematic assumption management in the design process.

A typical example of the issues encountered with assumptions in product development is in aircraft design. One way in which the aircraft design process can be described is as a sequential definition along a hierarchy of different sub-systems or disciplines, as illustrated in Figure 1 (Chen et al. 2023). In general, design teams will start with initial aircraft sizing to determine the aircraft weight, wing reference area, as well as the required engine thrust (Raymer 2018). Based on these top-level definitions, the airframe and engine will subsequently be further defined, followed by more detailed development and analysis of the lower-level sub-systems/disciplines, such as aerodynamics, structures, systems, and so forth. For example, it is infeasible to perform detailed aerodynamic analyses at the outset, because the geometry of the aircraft has not been fully specified yet. However, to proceed with the estimation of crucial performance parameters, an initial sizing must be performed. This, in turn, relies on the values of many aerodynamic coefficients being available. Therefore, assumptions must be made regarding these coefficients, often based on historical data and/or 'educated' judgements.

As the design teams proceed to the lower-level disciplines, an increasing amount of knowledge is obtained from simulations and experiments. Consequently, the assumed values of those aerodynamic coefficients will inevitably need to be revised. If the previously assumed values turn out to be too pessimistic, the aircraft will eventually be oversized. Conversely, with assumptions too optimistic, the aircraft may not be able to meet predefined requirements. Both cases will lead to a dilemma in which the designer must either proceed with the current design, of which the competitiveness may now be compromised; or change the top-level definition, which may trigger a substantial rework of all sub-systems.

Traditionally, margins have been widely used as a risk mitigation strategy to attempt to handle epistemic uncertainty and to provide scope for evolvability. Specifically, margins play a crucial role in absorbing initiated and emergent changes (Eckert, Clarkson, and Zanker 2004). After margins are 'consumed', changes can start propagating through a design, which in turn can lead to undesired iterations (Eckert, Isaksson, and Earl 2019).

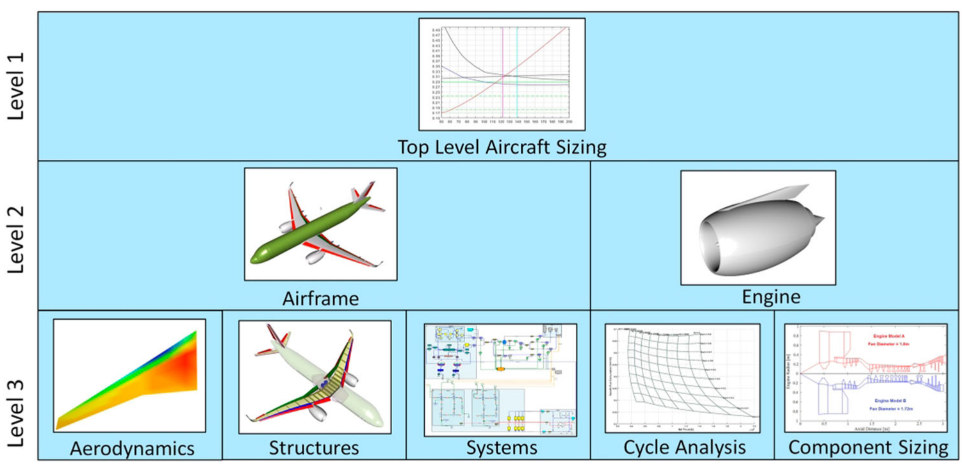

Figure 1. Hierarchical structure of a typical aircraft design problem (Chen et al. 2023).

As will be discussed in Section 2, extensive research has been conducted on change propagation (Brahma and Wynn 2021; Clarkson, Simons, and Eckert 2004; Eger, Eckert, and Clarkson 2005) and its relationship with margins (Brahma and Wynn 2020; Eckert et al. 2013; Eckert, Isaksson, and Earl 2019; Long and Ferguson 2020). However, there is still a lack of explicit definitions/descriptions of the potential relationships between assumptions and margins (El Fassi 2021). In addition, existing change propagation methods rely largely on the physical dependencies between the components within a system, whereas the links between requirements, functions, and computational models are not fully explored. Accounting for these domains of the product development process is important, as change initialisation or the effects of changes may only become apparent when these are explicitly considered. It should be emphasised that while change propagation provides essential context, the aim of this research is to find ways to accommodate a change (initiated by a revised assumption) before it propagates. Thus, the focus is on:

- the identification of margins or combinations of margins (potential change absorbers) to mitigate the potential adverse effects that may result from the revision of the assumption.
- how these margins can be traded off to balance performance and risk.

The work presented in this paper combines and extends our recent research on margin allocation (Guenov et al. 2018) and assumption management (El Fassi 2021; El Fassi, Guenov, and Riaz 2020). The scope is restricted to model-based design and systems engineering in the early stages of complex product development.

The rest of the paper is organised as follows. In Section 2, underlying definitions and concepts are introduced, and the research gaps are defined more precisely, following a brief review of the state-of-the-art. In Section 3, the proposed methods are described. These are demonstrated in Section 4, by making use of an aircraft design use case. Finally, conclusions are drawn, and future work is outlined in Section 5.

2. State-of-the-Art

2.1. Change propagation

As stated in the introduction, change propagation is not the main focus of this research, but provides essential context. That is, change propagation algorithms aim to identify the paths from one change to another, as illustrated by the red arrows in Figure 2, while our objective is to explore ways to accommodate a change (initiated by a revised assumption) before it propagates. If the change cannot be absorbed by existing margins, then appropriate change propagation methods (Brahma and Wynn 2020; Brahma and Wynn 2021; Clarkson, Simons, and Eckert 2004; Eckert et al. 2013; Eckert, Isaksson, and Earl 2019; Eger, Eckert, and Clarkson 2005; Long and Ferguson 2020) should be applied as a further action. It should also be noted that change propagation is not purely dictated by the availability/application of margins, but also by the availability of relevant resources (Koh, Caldwell, and Clarkson 2013). The reader is referred to (Brahma and Wynn 2022) for the state-of-the-art in design change propagation analysis.

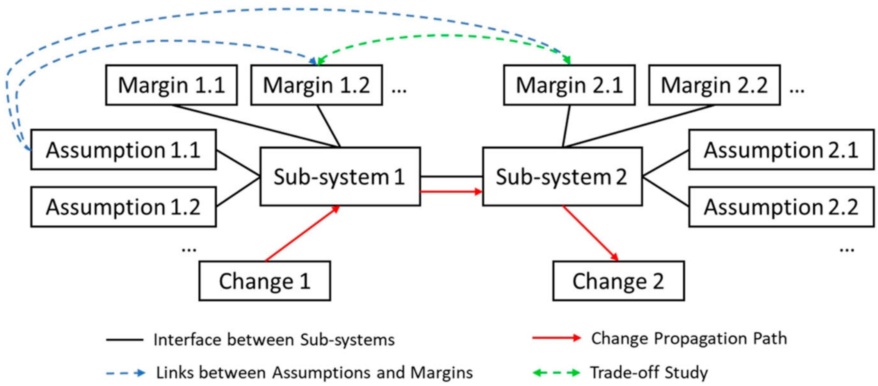

Figure 2. Scope of the research: conceptual relationship between subsystems, design changes, margins, and assumptions.

2.2. Assumptions and their lifecycle

According to the Guidelines for Development of Civil Aircraft and Systems (SAE ARP4754A), assumptions are defined as 'statements, principles, and/or premises offered without proof', and 'assumptions may be used early in the development process as a substitute for more explicit knowledge that will be available later' (SAE 2010). However, such definitions do not capture some essential characteristics of assumptions. According to Yang, Liang, and Avgeriou (2018), assumptions are:

- subjective, i.e. can be seen as assumptions by some stakeholders, or design decisions by others;
- related to other artefacts, such as requirements or components;
- dynamic, i.e. evolve with time (e.g. technological advancement may lead to component obsolescence and consequently its associated assumptions may become invalid).; and
- context-dependent, i.e. could be valid in one project, and invalid in another. As Brown (2006) argued: 'Design reuse can violate assumptions, as conditions that were true, or were assumed to be true originally may no longer be the case in the new design context'.

Another essential characteristic is that assumptions are inherently uncertain (Berner 2017; Jenkins, Woolston, and Boyd 2019), and the degree of confidence in making them varies based on the strength of background knowledge.

The fact that assumptions evolve with time suggests the idea of a lifecycle. Ostacchini and Wermelinger (2009) proposed a simple assumption lifecycle model that is composed of three stages: An assumption is made at first, which then goes through changes, and ultimately may or may not fail (i.e. become invalidated in the former case). Of interest in this research is how changes in assumptions influence the allocated margins. This is illustrated by the blue and green dashed arrows in Figure 2.

2.3. Margins in engineering design

Eckert et al. (2013) define a margin as the extent to which the value of a parameter exceeds what is necessary to fulfil requirements. A margin can, in effect, account for both uncertainty inherent in the current product (Cooke et al. 2015; Hall, Schroll, and Sharma 2020; Thunnissen 2004; Yuan et al. 2016; Zang et al. 2015), as well as for accommodation of potential future changes (Eckert, Isaksson, and Earl 2019). For the latter purpose, it is also referred to as 'excess', 'reserve', 'redundancy', or 'room for growth' in the literature (Allen, Mattson, and Ferguson 2016; Tackett, Mattson, and Ferguson 2014; van Heerden, Guenov, and Molina-Cristóbal 2019). This allows a product 'to be inherited and changed across generations (over time)', which defines the product's 'evolvability' (de Weck, Ross, and Rhodes 2012). Some specific aircraft design case studies can be found in (Lim 2009; Long and Ferguson 2017; van Heerden, Guenov, and Molina-Cristóbal 2019; Zhuravlev and Zhuravlev 2012). A comprehensive review of margin related methods and concepts within a broader engineering design context can be found in Brahma et al. (2023).

The most common approach to quantify margins is to use a deterministic representation, based on industrial standards, historical data or the intuition and experience of designers. However, such an approach is often conservative and may lead to over-designed products, resulting in performance penalties and increased costs. Furthermore, margins are added by different stakeholders without a unified way to allocate them and assess their impact on the design (Eckert et al. 2013). This is an ongoing issue (Eckert, Isaksson, and Earl 2019), where a need has been identified to track margins along with the rationale underlying their change, in addition to the need to develop mathematical models for margin management, such as presented in Touboul et al. (2019).

2.3.1. Margin as a change absorber

Margins play an important role in the management of engineering change through absorbing change (Brahma, Wynn, and Isaksson 2022). In fact, a change is required when no sufficient margin is left to absorb it (Eckert, Isaksson, and Earl 2019).

One of the first approaches to predict and manage changes in complex engineered systems was the Change Prediction Method (CPM) (Clarkson, Simons, and Eckert 2004), which is based on assessing the likelihood and impact of change propagation from a component to an adjacent one. However, values of likelihood and impact are derived from previous projects or expert opinion, which can be highly effort-intensive (or even unfeasible) in the context of large complex systems. Furthermore, a limit was introduced by Clarkson, Simons, and Eckert (2004) where the change propagation is considered to stop after three or four steps. The absence of explicit change absorbers in the product model could explain the need to put such a practical limit.

Margins are implicitly considered when eliciting likelihood and impact values through expert judgement, which prevents the revision of change propagation prediction as margins evolve (Long and Ferguson 2020). Subsequently, Long and Ferguson (2020) proposed to extend the CPM by accounting for the effect of decreasing margins (while keeping margins implicit). When a change is initiated, Long and Ferguson assume that the margin is completely consumed, which in turn increases the probability of change propagation. Moreover, change propagation is simulated by drawing from a uniform distribution to determine whether a component is part of the propagation path. This implies that different runs would randomly predict different propagation paths, which may not all exist in reality.

Although incorrect assumptions are known to cause change propagation in engineering design (Brahma and Wynn 2021), no approach has been proposed to explicitly relate assumptions to change propagation analysis. In this context, Brahma and Wynn (2021) suggest that 'approaches to track assumptions made during the design process could help to more effectively predict the impact of changes during that process', and that design change considerations could be integrated with margin management.

2.3.2. *Existing approaches to margin management*

To address the limitations associated with applying overly conservative margins, probabilistic approaches have been developed in the context of conceptual aerospace systems design (Cooke et al. 2015; Hall, Schroll, and Sharma 2020; Thunnissen 2004; Yuan et al. 2016; Zang et al. 2015). Such approaches can assist decision-making regarding margin allocation, for instance by showing the different combinations that yield compliant solutions. Other probabilistic approaches have been developed in different fields, such as bridge design (Michel and Fred 1986) and ship design (Mohammed et al. 2016).

Margins can also be introduced when a design involves component reuse. In such a context, Brahma and Wynn (2020) proposed the Margin Value Method (MVM), which supports locating components with residual margins and prioritising them for redesign. MVM necessitates knowledge about design parameters and their dependencies, which implies that a computational workflow is required. Furthermore, Brahma and Wynn proposed metrics to support margin analysis. Their metric regarding impact on performance is analogous to a simple One-at-a-Time sensitivity analysis, meaning that only one margin is changed at a time, while all the others are kept the same. Thus, there is an underlying implication that all margins are treated as independent.

Guenov et al. (2018) proposed an approach for interactive margin management, which is adopted in this work and allows exploring the effects of margins on: feasible values of design variables, feasible values of other margins, product performance, and probabilities of constraint satisfaction. That approach is based on a concept called *Margin Space*, which is a hypercube consisting of the ranges of all assigned margins and is bi-directionally linked to the design space (Guenov et al. 2018). Once the interval of each margin is defined, the Margin Space results from the Cartesian product of all margins. Each point in the Margin Space is considered a margin combination, of which the feasibility depends on whether the resulting performance meets the constraints. One relevant limitation is that margin evolution due to changing uncertainty is not accounted for (Guenov et al. 2018). For instance, validating assumptions would reduce epistemic uncertainty, which in turn should prompt a decrease in the levels of mitigating margins.

2.4. *Research gaps*

In summary, a change in an assumption is a potential initiator of a design change, while a margin is an important potential change absorber. However, not much has been explored regarding establishing an explicit path which connects assumptions, changes, and margins. Specifically:

(1) When assumptions are revised, there is a lack of effective methods to identify appropriate margins to absorb design changes before they propagate to other components. Such a method would necessarily have to account for non-physical dependencies, such as the links between requirements, functions, and computational models.

(2) There is a lack of methods to support the trade-off between different options if two or more margins are linked to a single change initiator.

A framework for assumption-margin management aiming to close these gaps is proposed in the next section.

3. Proposed framework

The proposed framework for assumption-margin management is described in this section. A prerequisite for the application of the framework is an initial definition of the system architecture (including a list of assumptions, margins, and their links), which should be stored as a Design Belief Network (DBN), using the method developed in (El Fassi 2021; El Fassi, Guenov, and Riaz 2020). This is briefly explained in Section 3.1.

The proposed framework consists of three parts:

(1) a taxonomy of margins and their formulations;
(2) an algorithm extended from (El Fassi 2021; El Fassi, Guenov, and Riaz 2020) for change absorber localisation;
(3) a margin trade-off approach, adapted from (Guenov et al. 2018) and enhanced with some assumption-related guidelines from (El Fassi 2021; El Fassi, Guenov, and Riaz 2020).

These three parts are described in detail in Sections 3.2, 3.3, and 3.4, respectively.

As illustrated in Figure 3, the starting point of the process is the availability of new knowledge, obtained from analysis and/or experiments, conducted after the initial definition of the system. As relevant knowledge is gained during the process, the confidence levels of all the assumptions need to be reviewed periodically (Block 1 in the flowchart), which leads to a list of revised assumptions. For each revised assumption, an algorithm is applied (Block 2) to detect change absorbers by scanning the margins stored in the predefined DBN. The algorithm also detects other assumptions on the path, between the initially revised assumption and the margins identified. These (newly-detected) assumptions can also be affected due to the design change. Therefore, the designer is notified to decide if the affected assumptions need to be revised as well. In such a case, the algorithm will be further applied to the affected assumptions to detect additional margins. This process will be repeated iteratively until the designer has revised all the affected assumptions (illustrated by 'Route a' in Figure 3).

At this point, a set of margins should be identified for all the assumptions (both initially-revised and further-affected). Once identified, the trade-off strategy adapted from Guenov et al. (2018) is used to adjust the magnitudes of these margins according to assumption-related guidelines adapted from El Fassi, Guenov, and Riaz (2020) and El Fassi (2021) (Block 3).

If there are not enough change absorbers, or there are no feasible solutions after the trade-off, a design change/iteration will be inevitable (Block 4). This may involve the allocation of new margins, change of the system definitions, or even the requirements (Route b).

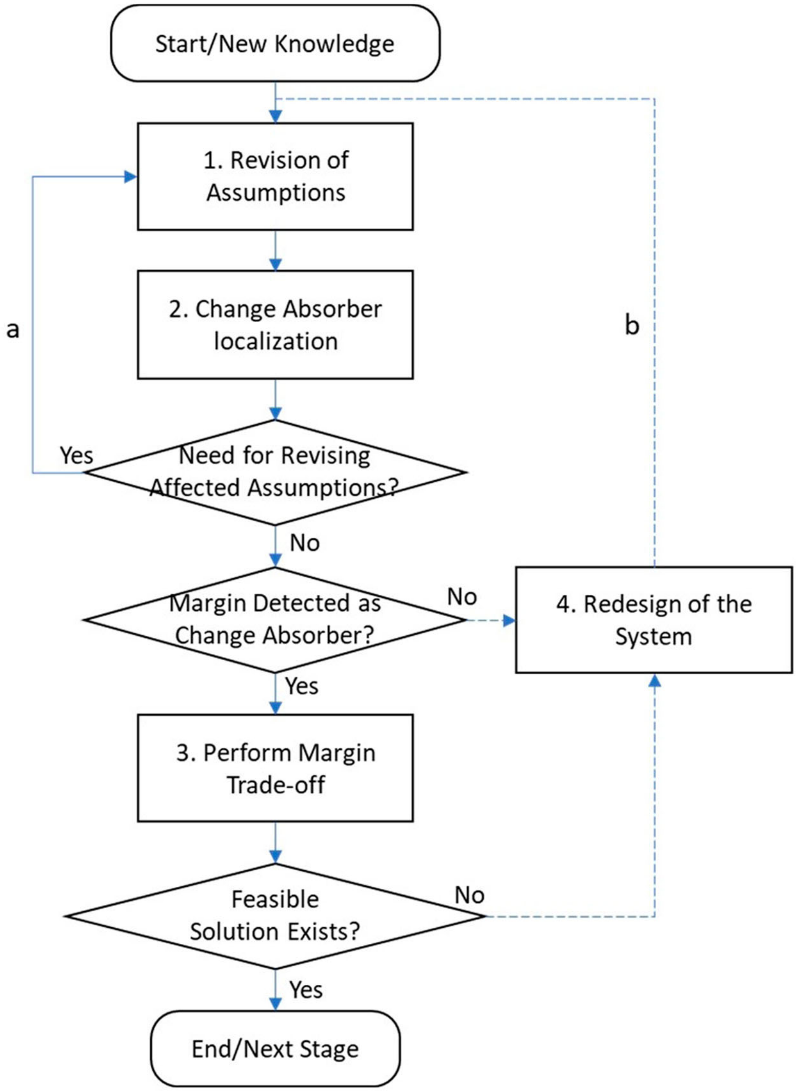

Figure 3. Flowchart of the assumption revision and margin trade-off process.

3.1. Design belief network (DBN)

A system architecture can be defined as 'the embodiment of concept, the allocation of physical/informational function to the elements of form, and the definition of relationships among the elements and with the surrounding context' (Crawley, Cameron, and Selva 2016). Functional reasoning, i.e. defining the functions to be performed by the system, plays a central role in system architecting (Umeda and Tomiyama 1997). One approach to functional reasoning is the RFLP paradigm (Kleiner and Kramer 2013) (illustrated in Figure 4), which is based on the VDI 2206 standard design methodology for mechatronic systems (VDI Verein Deutscher Ingenieure. 2004). RFLP considers that system architecting is distributed

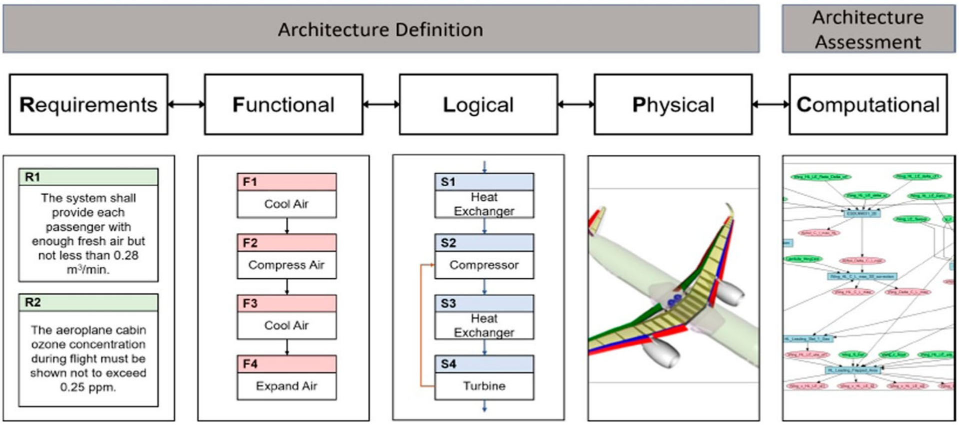

Figure 4. Requirements-Functional-Logical-Physical-Computational (RFLPC) representation of engineering systems, augmented with a computational domain (adapted from (Jimeno Altelarrea 2021)).

over four notional domains: 'Requirements', 'Functional', 'Logical' and 'Physical' (hence the abbreviation 'RFLP').

The DBN is a graph-theoretical structure that can be used to capture assumptions and their dependencies during system architecting, underpinned by RFLP. The DBN presented here relies on two building blocks: (1) the approach from Bile et al. (2018), who proposed to augment the RFLP representation of engineering systems with a 'Computational' domain (C) for model-based design, thereby enabling automated systems sizing and performance assessment (as illustrated in Figure 4); and (2) the graph-theoretical structure from Guenov et al. (2020) which captures the dependencies between the R-F-L-C domains.

The Computational domain consists of a computational workflow (illustrated in Figure 4), which can be defined as 'an ordered set of computational models' (Jimeno Altelarrea et al. 2020). Such models are associated with components to assess their behaviour, and the computational workflow sets the execution order of the models. The Computational domain can be used to capture margins and assess their impact on performance, thus supporting margin trade-off. The Physical domain captures (usually in CAD/PLM formats) the physical connectivity, topology, and spatial layout of the evolving product. It is an important domain which currently is out of scope in this research.

A DBN graph, $\mathbb{G}(\boldsymbol{V}, \boldsymbol{E})$, consists of a set of its vertices (nodes), $\boldsymbol{V}$, and edges, $\boldsymbol{E}$. The nodes of the DBN graph correspond to the captured elements of the R-F-L-C domains (i.e. requirements, functions, components, computational models, parameters, and margins), whereas the edges of the DBN graph correspond to the dependencies between the nodes (e.g. an edge connecting a component and the function it realises, or an edge connecting a requirement and its underlying assumption(s)). Additional information is carried by the attributes of each node, for instance, the confidence level of an assumption, the status of a component (frozen/unfrozen), the magnitude of a margin, and so forth. Figure 5 illustrates the data structure of the DBN.

Assumptions and their dependencies are captured as the system is being defined. An example of a DBN in the context of the conceptual design of a fighter aircraft can be found in (El Fassi 2021), and a simplified example is illustrated in Figure 6. For instance, it can be

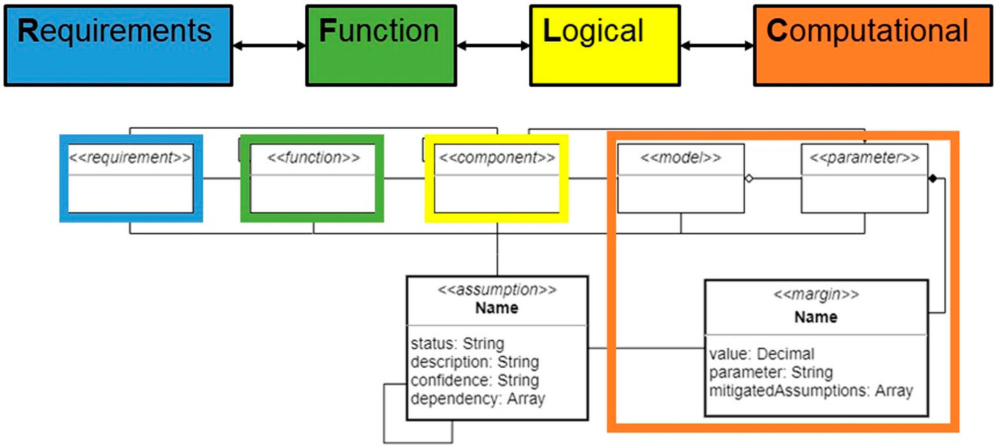

Figure 5. Design Belief Network (DBN) data structure (El Fassi 2021).

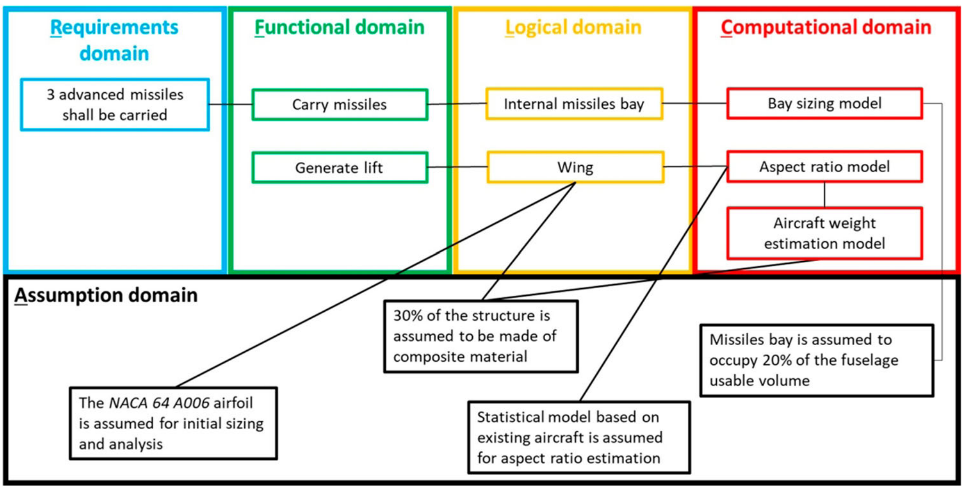

Figure 6. Simplified illustration of a DBN (adapted from (El Fassi, Guenov, and Riaz 2020)).

assumed that 30% of the structure is to be made of composite material which, as shown in Figure 6, affects both the 'Wing' component and the computational model for aircraft weight estimation. To capture this assumption, a new instance of the assumption class (cf. Figure 5) can be created and linked to the wing component via a software implementation of the DBN method. Similarly, an instance of the margin class (cf. Figure 5) can be created in the computational domain and assigned to the output of the weight estimation model in order to accommodate the uncertainty. As the design progresses, the assumptions made, as well as their dependencies, are gradually captured by the designer. Further details regarding the DBN can be found in (El Fassi 2021; El Fassi, Guenov, and Riaz 2020).

3.2. *Margin taxonomy and formulation*

Consider the computational design problem illustrated in Figure 7. The variables relevant to the design problem can be classified into four categories: design variables (x), parameters

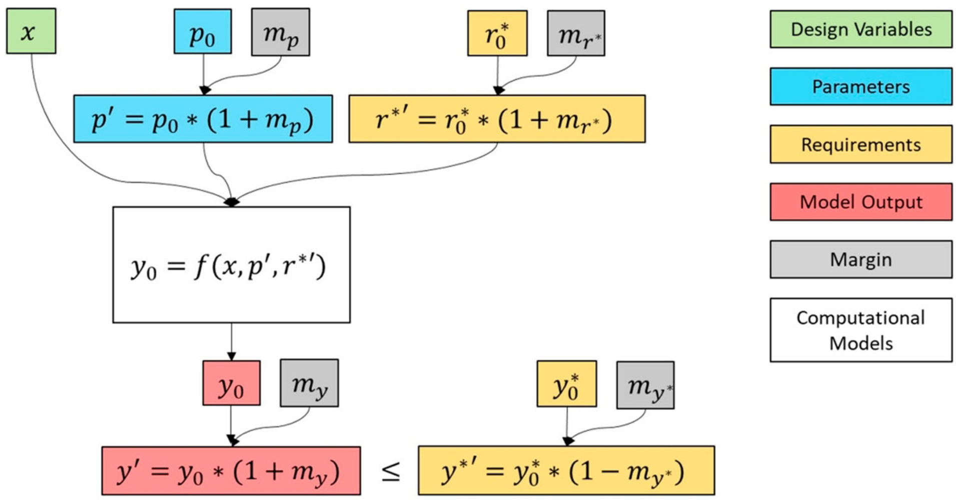

Figure 7. Context of a computational design problem.

(p), model outputs (y), and requirements (r^* and y^*). The design variables (shown in the green boxes in Figure 7) define the system under consideration. Their values are controlled by a designer or an optimiser. The parameters are indicated by the blue colour. Their values are pre-defined in the design process according to existing knowledge or are assumed if such knowledge is not available. The variables in the yellow boxes in Figure 7 represent design requirements, which are also pre-defined in the design process. A subset of these requirements is used as model inputs (r^*) while others are used to formulate problem constraints (y^*).

Mathematically, the first subset of requirements (r^*) is equivalent to the parameters (p). However, their values are subject not only to knowledge/assumptions, but also to the customer needs, which makes them less negotiable. The red block in Figure 7 indicates the output produced by a computational model, where the latter can be regarded as a function of the design variables, parameters, and (input) requirements, i.e.:

$$y = f(x, p, r^*), \tag{1}$$

where the computed output can be further subject to a set of constraints, such as:

$$y \le y^* \text{ or } y \ge y^*. \tag{2}$$

If an optimisation paradigm is adopted, the model output can also be defined as an objective function for maximisation or minimisation, which is beyond the scope of this paper, but does not affect generality.

The margins are shown in Figure 7 in the grey boxes. Within the scope of this research, four categories of margins are considered and illustrated in Figure 8. On the left side of the figure is a two-dimensional space of r^* and p, while the right part of the figure is a one-dimensional axis, y. The first category of margins encompasses those allocated to account for the uncertainty associated with the assumed parameters. For example, p is a parameter and its value is preferred to be lower (to produce a better performance). Given an initial

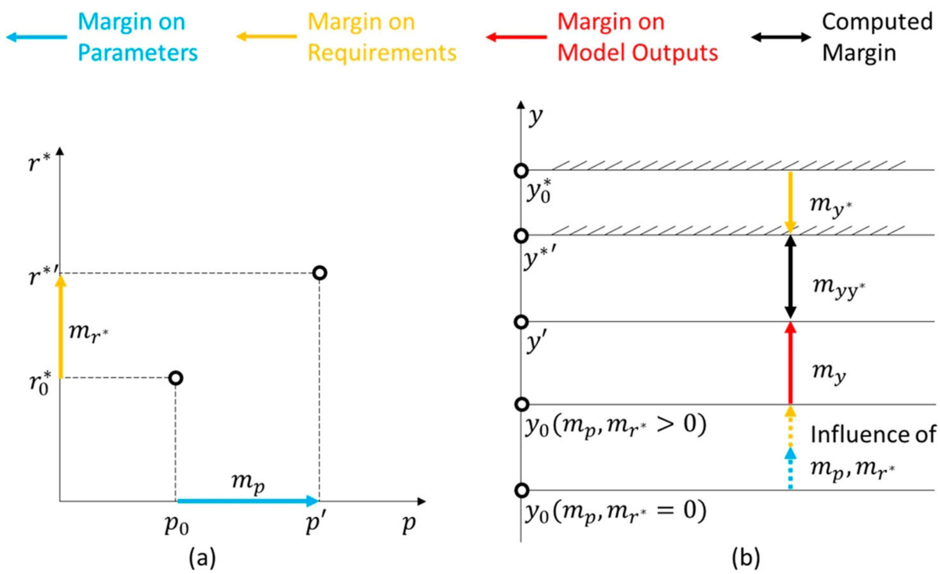

Figure 8. Illustration of four types of margins.

assumed value p_0, the margin m_p on this parameter should accommodate a less optimistic value of p compared with the current expectation (that is, a higher value of p). This is illustrated by the solid blue arrow in Figure 8(a) and its influence on the computed output y is shown by the dashed blue arrow in Figure 8(b). Mathematically, the margin can also be formulated as:

$$p' = p_0 * (1 + m_p) \tag{3}$$

The second category consists of those margins allocated to account for the uncertainty associated with the computational models employed in the design process. Consider y as a performance output (e.g. the landing distance) and assume its value is preferred to be lower. Given that the nominal computed value is y_0, the margin m_y should accommodate a scenario where the actual value of y turns out to be higher than the prediction (landing distance longer than expected). This is illustrated by the solid red arrow in Figure 8(b), and the corresponding formulation is:

$$y' = y_0 * (1 + m_y) \tag{4}$$

The third category encompasses margins allocated to account for the uncertainty associated with the requirements (in case they become more stringent, e.g. we want to further reduce the landing distance). As discussed earlier, some of the requirements (such as r^*) are used as model inputs and mathematically equivalent to the parameters. This is illustrated by the solid yellow arrow in Figure 8(a) and its influence on the computed output y is shown by the dashed yellow arrow in Figure 8(b).

$$r' = r_0 * (1 + m_r) \tag{5}$$

The margin on the constraint requirement y^* is illustrated by the solid yellow arrow in Figure 8(b) and can be formulated as:

$$y^{*\prime} = y_0^* * (1 - m_{y^*}) \tag{6}$$

The last category of margins is defined as the distance between the derated performance (e.g. computed landing distance, plus a reserve) and the enhanced requirements (e.g. required landing distance, minus a reserve), as illustrated by the solid black arrow in Figure 8(b). Although the value of this margin cannot be controlled directly, it serves as an indicator of reserve/contingency left to absorb design changes caused by other unforeseen scenarios (i.e. 'unknown unknowns').

$$m_{yy^*} = \frac{y^{*\prime} - y'}{y^{*\prime}} \tag{7}$$

3.3. Change absorber localisation

During the design process, assumptions may be revised as a result of new knowledge gained from analyses and/or experiments which are performed after the initial definition of the system. Such revisions may lead to design changes if critical assumptions turn out to be invalid or less credible. If such a scenario occurs, the second part (implemented as an algorithm) of the margin management framework is employed to detect:

- A list of margin/assumption localisation paths which start from the revised assumption (acting as a *Change Initiator*) and end with margins (acting as *Change Absorbers*).
- The type of each margin/assumption localisation path, based on the relationship between the revised assumption and identified margins (as illustrated in Figure 9).
- Assumptions associated with elements along the identified path.
- Components along the identified path which are specified as frozen.

As mentioned earlier, the input to the algorithm is a predefined Design Belief Network (DBN), where elements of the system architecture are stored as nodes, while the dependencies between each other are stored as edges. Additional information is carried by the attributes of the node, for instance, the confidence level of an assumption, the status (frozen/unfrozen) of a component, the magnitude of a margin, and so forth.

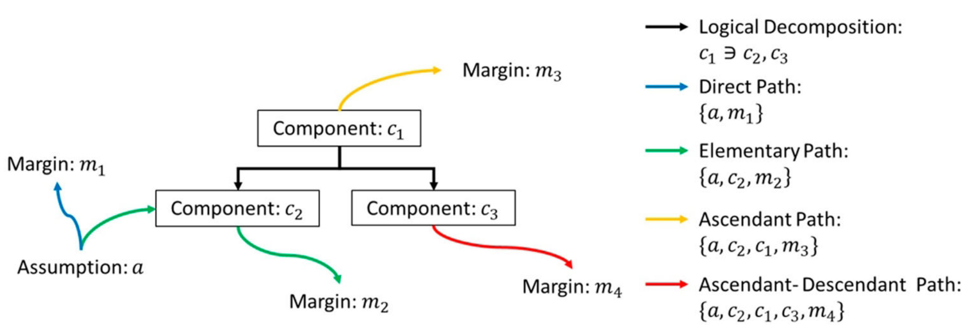

Figure 9. Illustration of different path types.

Table 1. Notations of the objects in the algorithm.

Category	Symbol	Definition
Node/Single Attribute	a	The revised assumption
	m_i	The i_{th} margin
	$c_{i,j}$	The j_{th} component in the i_{th} margin/assumption localisation path
	t_i	The type of the i_{th} margin/assumption localisation path
List of nodes	$\boldsymbol{p_i}$	The i_{th} margin/assumption localisation path (List of nodes $\{a, \ldots, m_i\}$)
	$\boldsymbol{c_i}$	List of components in the i_{th} margin/assumption localisation path
	$\boldsymbol{c_{F\,i}}$	List of frozen components in the i_{th} margin/assumption localisation path
	$\boldsymbol{a_i}$	List of assumptions associated with the nodes along the i_{th} path
	$\boldsymbol{t}$	List of types for all the margin/assumption localisation paths $\{t_1, t_2, t_3 \ldots\}$
List of Lists	$\boldsymbol{P}$	A list of all the margin/assumption localisation paths $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3 \ldots\}$
	$\boldsymbol{C_F}$	A list of lists of frozen components in each path $\{\boldsymbol{c_F}_1, \boldsymbol{c_F}_2, \boldsymbol{c_F}_3 \ldots\}$
	$\boldsymbol{A}$	A list of lists of affected assumptions in each path $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \ldots\}$
Operation	$len(*)$	Length of the list, defined as the number of nodes in the list
	$DP(*,*)$	Shortest path between the two nodes using Dijkstra's search algorithm
Graph	$\mathbb{G}$	The Design Belief Network (DBN)

The detailed steps of the algorithm and the notion of the involved objects are presented in Table 2 and Table 1, respectively. The notation rules are as follows: (1) A node or a single attribute is noted in a nonbold lowercase; (2) A list of nodes is noted as a **bold lowercase**; (3) A list of lists is noted as a **BOLD UPPERCASE**.

In lines 3–7, given a revised assumption (a) and for each margin (m_i) recorded in the design belief network ($\mathbb{G}$), Dijkstra's search algorithm (Dijkstra 1959; Saoub 2017) is used to find the shortest path $\boldsymbol{p_i}$, which is a list of linked nodes from a to m_i, with the minimum number of nodes (including requirements, functions, components, models, variables, etc.) in between. All the paths are stored in $\boldsymbol{P}$, in ascending order of length, that is, the number of nodes in each path. Note that this is based on the graph traversal algorithm, which involves no optimisation, but rather a ranking process. The resulting 'shortest' path is not necessarily the 'optimal' one, because optimality relies on additional factors, such as the ease of modifying a component, extra cost introduced, and so forth.

In lines 8–30, each path is explored (traversed), to determine its type and to record the associated components and assumptions that may be further affected. Specifically, in lines 10–11, if a and m_i are the only two nodes in the margin/assumption localisation path, then this margin is directly linked to the assumption, which defines the path type as '*Direct*' (illustrated by the blue arrow in Figure 9). This occurs only if the link is defined manually when creating the system architecture and the corresponding design belief network (before revision of the assumptions).

In lines 12–13, if there is only one component node in the margin/assumption localisation path, then the type will be noted as '*Elementary*', because other components will not be influenced (illustrated by the green arrows in Figure 9). If there is more than one component, lines 14–22 constitute a check of the relationships between these components. In a path, if the previous component is always a *Child* node of the subsequent one, the path is considered as '*Ascendant*' (illustrated by the orange arrow), otherwise the path will be noted as '*Ascendant-Descendant*' (illustrated by the red arrow). In a practical design setting, this may necessitate a coordination process between multiple designers or design teams. The Parent–Child relationships are captured from the DBN. If the path is neither '*Direct*' nor has any associated components, the type will be noted as '*Undetermined*' and left to the designer to make a judgement. The type of each path t_i will be stored in a list called $\boldsymbol{t}$.

Table 2. Change absorber localisation algorithm.

Algorithm: Change Absorber Localisation

input: $\mathbb{G}$, a

output: $\boldsymbol{P}, \boldsymbol{t}, \boldsymbol{C_F}, \boldsymbol{A}$

begin

  Initialize $\boldsymbol{P}, \boldsymbol{t}, \boldsymbol{C_F}, \boldsymbol{A}, \boldsymbol{a_i}, \boldsymbol{c_i}, \boldsymbol{c_{F_i}}$ as empty lists

  for $(i = 1\ to\ number\ of\ margins\ in\ \mathbb{G})$

    Perform Dijkstra's method to find the shortest path $\boldsymbol{p_i} = DP(a, m_i)$ in $\mathbb{G}$

    Append $\boldsymbol{p_i}$ to $\boldsymbol{P}$

  end for

  Sort $\boldsymbol{P}$ in ascending order of length of each path (number of nodes)

  for $(i = 1\ to\ number\ of\ paths\ in\ \boldsymbol{P})$

    $\boldsymbol{c_i} \leftarrow$ all component nodes in path $\boldsymbol{p_i}$

    if $len(\boldsymbol{p}_i) = 2$

      $t_i =$ "$Direct$"

    else if $len(\boldsymbol{c_i}) = 1$

      $t_i =$ "$Elementary$"

    else if $len(\boldsymbol{c_i}) > 1$

      $t_i =$ "$Ascendant$"

      for $(j = 1\ to\ (number\ of$ components $in\ \boldsymbol{c_i}) - 1)$

        if $c_{i,j+1}$ is not a parent node of $c_{i,j}$

          $t_i =$ "$Ascendant - Descendant$"

          break

        end if

      end for

    else

      $t_i =$ "$Undetermined$"

    end if

    Append t_i to $\boldsymbol{t}$

    $\boldsymbol{c_{F_i}} \leftarrow$ frozen component nodes in path $\boldsymbol{p_i}$

    Append $\boldsymbol{c_{F_i}}$ to $\boldsymbol{C_F}$

    $\boldsymbol{a_i} \leftarrow$ assumption nodes associated with nodes in path $\boldsymbol{p_i}$

    Append $\boldsymbol{a_i}$ to $\boldsymbol{A}$

  end for

  return $\boldsymbol{P}, \boldsymbol{t}, \boldsymbol{C_F}, \boldsymbol{A}$

end

Additionally, the Status attribute of each component within each path can be checked to determine whether that component has been frozen. Such information is also expected to support the assessment of change impact, in the sense that frozen components are more costly to change. In lines 26–27 and lines 28–29, the frozen components and affected assumptions each along the margin/assumption localisation path will be recorded and stored in $\boldsymbol{C_F}$ and $\boldsymbol{A}$, respectively.

Note that in Figure 9, c_i is the unique notation of the i_{th} component in the architecture. In the algorithm and Table 1, each component $c_{i,j}$ has two subscripts indicating the path and sequence, that is, the same component can have different indexes in different paths.

3.4. Margin revision and trade-offs

By applying the algorithm presented in Section 3.3, for each revised assumption, a list of margin/assumption localisation paths will be generated and sequenced in an ascending order of path length. Each path is terminated with a margin that can potentially be used as a change absorber.

If the confidence level of an assumption increases, the associated margins could be reduced, as there is a lower risk for the assumption to be invalid. On the other hand, the associated margins should be increased if the assumption becomes less credible or invalid. However, the margins cannot be modified arbitrarily without impacting the feasibility and performance of the design solution. In addition, there are multiple candidate margins for each revised assumption, which leads to different possible solutions.

To tackle these issues, a set-based approach from Guenov et al. (2018) is adopted. It enables interactive trade-off studies such that a set of feasible and promising solutions can be identified. Specifically, the method considers trade-offs between three pairs: (1) margins and margins/design variables; (2) margins and performance variables; (3) margins and probabilities of constraint satisfaction. For that purpose, design of experiment studies is employed to populate the design space and margin space. By visualising the constraints using contour plots, a set of feasible solutions could be down-selected.

An illustrative example with two design variables (x_1, x_2), two margins (m_1, m_2), and one constraint is given in Figure 10, where each green point indicates a solution (that is a combination of x_1, x_2, m_1, m_2 values) in the design-margin space. The points located in the white regions are feasible solutions, which satisfy the constraint; while those in the grey regions are infeasible solutions, which violate the constraint.

In Figure 10(a), selecting different points (DP1 and DP2) in the design space leads to different feasible regions in the margin space. In this example, choosing DP1 would result in more feasible margin combinations, compared with DP2. In Figure 10(b), the designer first selected a point in the design space and then a point in the margin space. It can be seen that after applying the margins, the feasible region in the design space has been reduced. The margin-performance and margin-probability trade-offs can be performed in a similar manner, by linking points in the margin spaces to Pareto fronts and probabilities in the corresponding output spaces, respectively.

As mentioned earlier, there are multiple candidate margins for each revised assumption. Subsequently, the dimensions of the combinatorial margin space could be very high, which may lead to prohibitive computational cost in producing the design set.

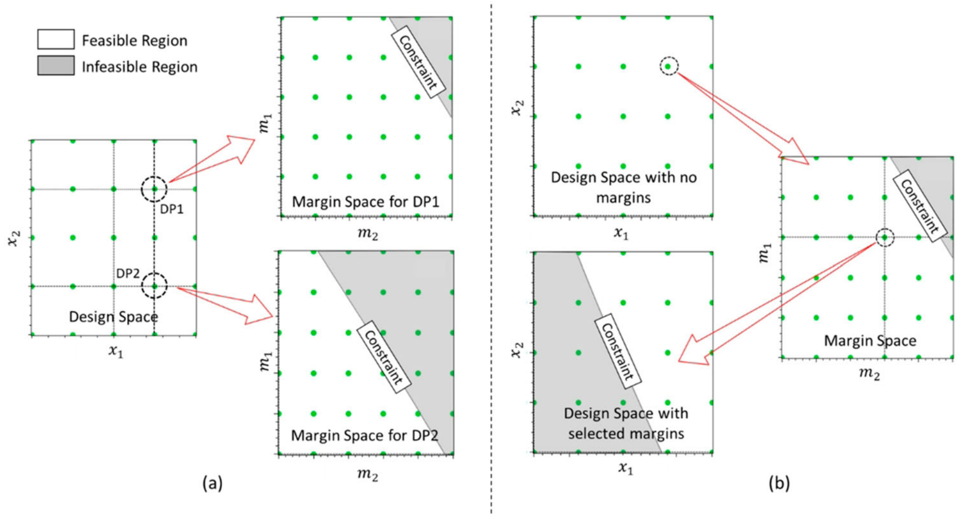

Figure 10. Trade-off between margins and other margins/design variables (adapted from (Guenov et al. 2018)).

Therefore, the following strategies are adopted to gradually accommodate more margins in the trade-off study. The strategies are based on the lengths and types of the margin/assumption localisation paths, which are produced by the algorithm as presented in Section 3.3.

For each revised assumption, its directly associated margin should first be included in the trade-off study. If any of the following scenarios are encountered: (1) the assumption is not directly linked to any margins; (2) there are no feasible solutions after the trade-off; or (3) the performance has been substantially impacted; then the margins located in the '*Elementary*' paths should be included as additional options. If the trade-off results are still not satisfactory, the margins located in other paths should also be included. Within each category of paths, the margins with the shortest distance to the revised assumption should be included first. Finally, a margin/assumption localisation path should be avoided if it contains nodes which are 'frozen' components. It must be emphasised that the strategies above only serve as guidelines, and it is the choice of the designer to include their margin candidates. In practice, the choices may also be subject to other factors which are difficult to quantify, such as the cost of changing a specific margin. This part is beyond the current scope of the paper.

4. Demonstration

The proposed framework is demonstrated with a ***realistic, but not real*** case study, adapted from Raymer (2018). It concerns the conceptual design of a lightweight fighter aircraft. The original use case is augmented here with additional hypothetical requirements regarding the future upgradability of structural materials, engine, and weapon systems. This illustrates a realistic scenario where novel, but less mature technologies are considered at an early design stage for product evolvability.

Due to a lack of historical data and prior experience, assumptions are made to fill knowledge gaps and margins are assigned to mitigate risks. Apart from the perspective of design evolvability, assumptions and margins are also defined to account for uncertainty introduced by parameters, computational models, and flexible requirements.

4.1. Use-case description

4.1.1. Architecture definition in R-F-L domains

This section defines the system architecture of the aircraft in the Requirement (R), Functional (F), and Logical (L) domains. The Physical (*P*) domain (e.g. digital mock-up) is not included here for the sake of simplicity.

In the original example from (Raymer 2018), the requirements are defined in line with the details of each mission segment (e.g. take-off, climb, cruise, acceleration, supersonic dash, combat, loitering, and landing). In this illustrative example, these requirements are simplified and grouped into three categories, as summarised in Table 3. In the computational domain, the first category is associated with design inputs, which are equivalent to the r^* in Equation (1). The second category is specified on performance outputs and formulated as (inequality) constraints, as defined by Equation (2). The third group represents non-numerical requirements associated with the evolvability of the aircraft.

The functional and logical domains are defined using an in-house tool called AirCADia Architect (Guenov et al. 2020). The hierarchical decomposition views of the functions and components are illustrated in Figures 11 and 12, respectively. In the definition of the architecture, the links between the functions and components are also captured, as illustrated in Figure 13. The hierarchical decompositions together and the cross-domain links are stored as an integrated graph in the software.

4.1.2. Computational domain

The schematic view of the computational domain is shown in Figure 14. Here, the same colour scheme as in Figure 7 is used to distinguish different types of variables (as discussed in Section 3.2). Note that this is the computational workflow before margin allocation and all the variables are defined with their initial/nominal values.

The design variables include aircraft wing loading and thrust-to-weight ratio, which are typical variables used in initial aircraft sizing. The parameters are related to the aerodynamic and engine performance, which need to be assumed based on existing aircraft with similar characteristics. The model outputs include take-off and landing rolling distances, as well as the sustained turn load factor, computed with equations from (Mattingly et al. 2018;

Table 3. Use-case requirements.

Categories	Symbol	Description
Design inputs	r_1	Total combat range (outbound + inbound) is 500 nautical miles
	r_2	Total combat time is 3 min
	r_3	Total payload weight is 1,240 pounds
Performance outputs	r_4	The take-off ground rolling distance should be less than 1,000 ft
	r_5	The landing ground rolling distance should be less than 1,000 ft
	r_6	Maintain a sustained turn in combat of at least 5 g
Evolvability requirements	r_7	Aircraft must be adaptable to incorporate a future advanced engine
	r_8	Aircraft must be adaptable to incorporate future advanced weapon systems

Raymer 2018). The right-hand side of the workflow is the so-called mass-fuel loop that computes the take-off weight of the aircraft. Further details of each variable are summarised in Table 4.

4.1.3. Assumptions and margins

During the design process, a set of assumptions are made to fill gaps in knowledge. As summarised in Table 5, the assumptions can also be categorised into three groups. The first category of assumptions is made up of the values of the parameters (as defined in Table 4), the second is associated with ensuring adequate provision for the incorporation of potential future technologies, whereas the third accounts for potential changes in requirements

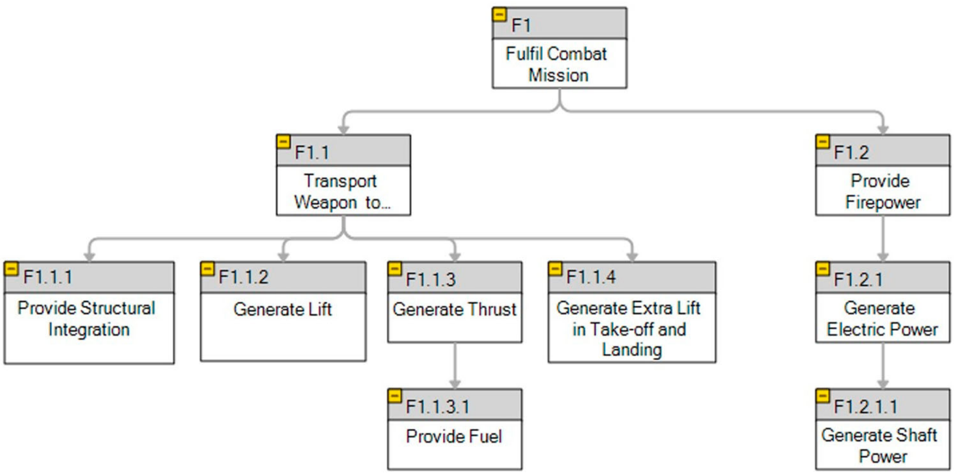

Figure 11. Functional decomposition view of the example.

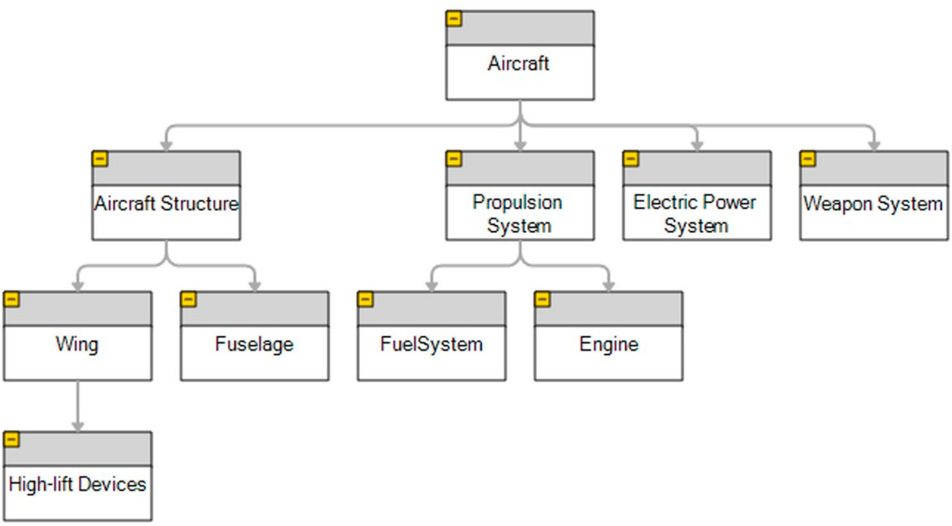

Figure 12. Product decomposition view of the example.

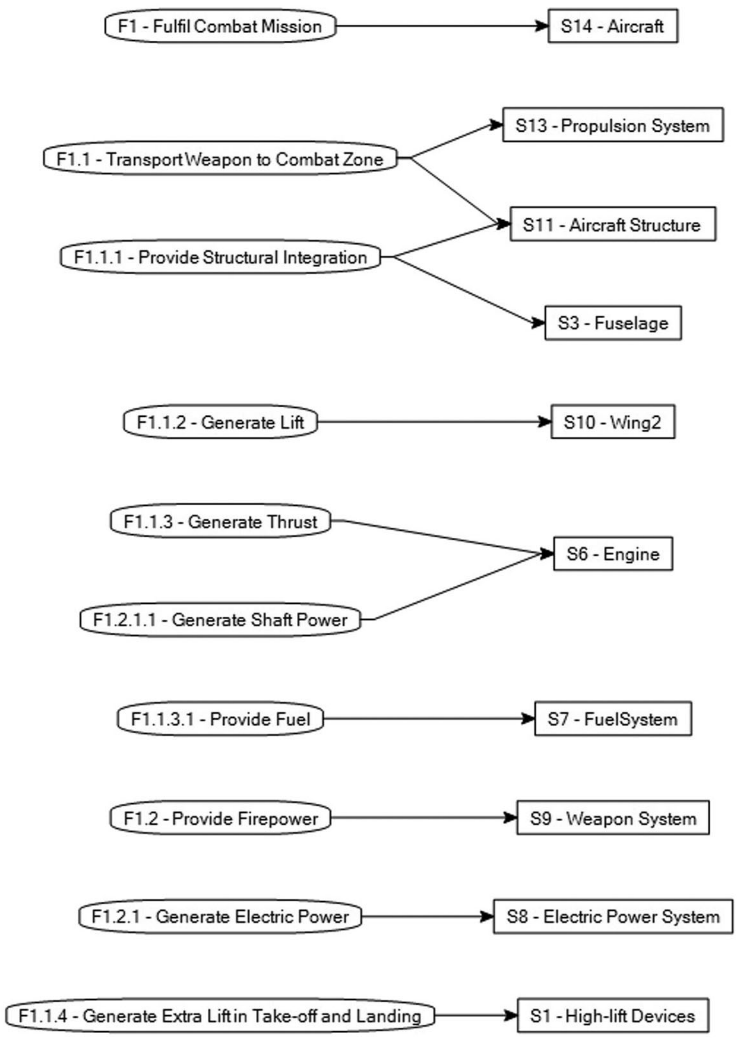

Figure 13. Function-means mapping of the example.

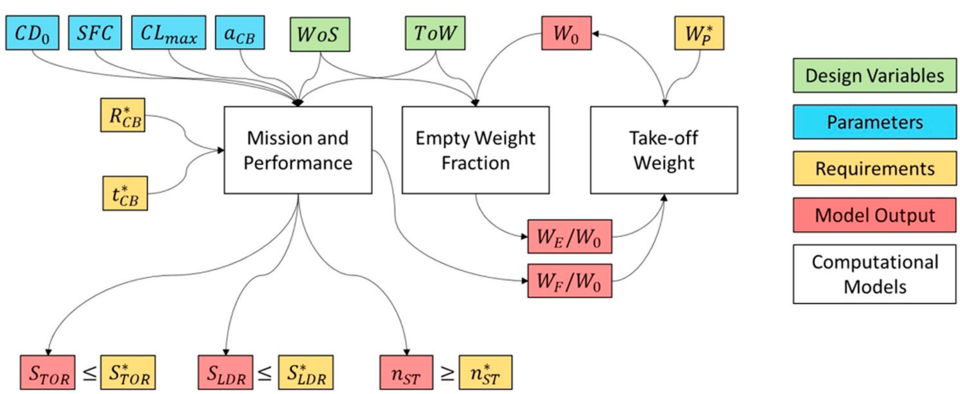

Figure 14. The example use-case computational domain.

Table 4. Details of each variable in the computational domain.

Categories	Name	Symbol	Description
Design variables	Wing loading [lb/ft^2]	WoS	Aircraft take-off maximum weight, divided by wing reference area.
	Thrust-to-weight ratio	ToW	Aircraft sea level static thrust divided by maximum take-off weight.
Parameters	Cruise zero-lift drag coefficient	CD_0	Drag coefficient at zero lift (lift and drag coefficients are non-dimensional variables representing aircraft aerodynamic performance. They are defined as a force divided by wing reference area and dynamic pressure).
	Maximum lift coefficient	CL_{max}	Maximum lift coefficient in take-off and landing configurations (with high-lift device fully deployed).
	Cruise thrust specific fuel consumption [$lb/hr/lb$]	SFC	Mass of fuel consumed by the engine during cruise per unit time and per unit of thrust generated.
	Engine thrust lapse rate in combat conditions	a_{CB}	Maximum available thrust from engine in combat condition (0.9 Mach, 30000 ft) divided by the sea level static thrust.
Requirements	Required combat range [nm]	R^*_{CB}	Outbound + inbound travelling distance to the combat zone (r_1 in Table 3).
	Required combat time [min]	t^*_{CB}	Time of the aircraft in combat manoeuvre and thrust setting (r_2 in Table 3).
	Required payload weight [lb]	W^*_P	Total weight of weapons (missiles and ammunition) and other detachable mission related equipment, such as targeting pods, Electronic Countermeasure (ECM) pods, and so forth (r_3 in Table 3).
	Required take-off rolling distance [ft]	S^*_{TOR}	Distance for the aircraft to accelerated from static to lift-over during take-off.
	Required landing rolling distance [ft]	S^*_{LDR}	Distance for the aircraft to decelerate from touch down to full stop during landing.
	Required load factor in sustained turn	n^*_{ST}	Load factor for the aircraft to maintain in a sustained (i.e. without losing speed) turn in combat.
Performance output	Empty weight fraction	W_E/W_0	The ratio between empty weight and take-off weight (the former is defined as the aircraft weight excluding crew, payload, and fuel).
	Fuel weight fraction	W_F/W_0	The ratio between fuel weight and take-off weight.
	Take-off weight [lb]	W_0	Weight when the aircraft starts to take-off, with full load weapons and fuel (excluding the fuel burn during warmup and taxiing).
	Computed take-off rolling distance [ft]	S_{TOR}	The actual computed ground rolling distance in take-off, which should be less than S^*_{TOR}.
	Computed landing rolling distance [ft]	S_{LDR}	The actual computed ground rolling distance in landing, which should be less than S^*_{LDR}.
	Computed load factor in sustained turn	n_{ST}	The actual computed load factor in sustained turn, should be larger than n^*_{ST}.

in the future (beyond the first entry-into-service date). Both the second and third categories of assumptions are related to the evolvability of the aircraft.

To mitigate the risks introduced by the assumptions and other sources of uncertainty, the following margins are allocated (summarised in Table 6). The first category of margins is

Table 5. Assumptions made for the design case-study.

Categories	Symbol	Description	Confidence level
On parameters	a_1	The zero-lift drag coefficient CD_0 is assumed to be 0.014 in accordance with existing aircraft of similar configuration.	Moderate
	a_2	The maximum lift coefficient during take-off and landing, CL_{max}, is assumed to be 1.8 according to data on existing aircraft of a similar configuration.	High
	a_3	The baseline engine specific fuel consumption SFC is assumed to be 1.18 according to data for an existing engine on a similar aircraft.	Moderate
	a_4	The engine thrust lapse rate in combat condition a_{CB} is assumed to be 0.53 according to data on existing engine on a similar aircraft.	Moderate
On future technology	a_5	Future engine will offer 20% lower fuel consumption ($\Delta SFC_{Eng} = -20\%$).	Moderate
	a_6	Future high-energy weapon system will require 10% more power consumption ($\Delta SFC_{Wpn} = 10\%$).	Low
	a_7	Future high-energy weapon system will be 10% heavier than the current weapon combinations ($\Delta W_P^* = 10\%$).	Low
On future requirements	a_8	Future mission will require 10% extra range ($\Delta R_{CB}^* = 10\%$).	Moderate
	a_9	Future mission will require 10% reduction in take-off rolling distance ($\Delta S_{TOR}^* = -10\%$).	Moderate

used to accommodate uncertainty associated with the assumed parameters. Note that the -20% and $+10\%$ in the formulation of SFC' come from assumptions a_5 and a_6 in Table 5, which account for the influence of the future engine and weapon systems, respectively. Margins in the second category are allocated to the output of the computational models. Margins in the third category are allocated to the requirements to make them more stringent. Note that the 10% deviations are added according to assumptions a_7, a_8, and a_9, respectively. Margins in the last category are computed as the distance between the derated performance and the enhanced requirements.

If a margin is assigned directly for a specific assumption, the link should be recorded in the DBN. However, there is not necessarily a one-to-one mapping relationship between the assumptions and margins. For example, the designer may decide not to allocate any margin for the assumed CL_{max}, due to the high confidence in available information. In addition, one margin could be used to account for uncertainty associated with multiple assumptions. For instance, m_2 on the engine SFC can be linked to a_3, a_5, and a_6. Indeed, allocating a separate margin for each of these three assumptions may provide better traceability, but it will also complicate the problem in terms of additional variables and equations. It is up to the designer to formulate the problem considering other factors, such as available resources.

On the other hand, multiple margins can be allocated to cover a single assumption. In this case study, both m_8 and m_9 can be used to accommodate a potential future change in the requirement on take-off rolling distance. While m_8 is controlled by the designer to

Table 6. Margin allocated for risk mitigation.

Category	Symbol	Justification	Formulation
On parameters	m_1	To account for potentially underestimated CD_0	$CD_0{}' = CD_0 * (1 + m_1)$
	m_2	To account for potentially underestimated *SFC*	$SFC' = SFC * (1 - 20\% + 10\% + m_2)$
	m_3	To account for potentially overestimated a_{CB}	$a_{CB}{}' = a_{CB} * (1 - m_3)$
On model outputs	m_4	To account for potentially underestimated W_E/W_0	$(W_E/W_0)' = W_E/W_0 * (1 + m_4)$
	m_5	To account for potentially underestimated S_{TOR}	$S_{TOR}{}' = S_{TOR} * (1 + m_5)$
On requirements	m_6	To account for potentially more stringent requirement of W_P^* (more payload weight)	$W_P^{*\prime} = W_P^* * (1 + 10\% + m_6)$
	m_7	To account for potentially more stringent requirement of R_{CB}^* (longer combat range)	$R_{CB}^{*\prime} = R_{CB}^* * (1 + 10\% + m_7)$
	m_8	To account for potentially more stringent requirement S_{TOR}^* (shorter take-off rolling distance)	$S_{TOR}^{*\prime} = S_{TOR}^* * (1 - 10\% - m_8)$
Computed margins	m_9	Computed margin between the required and computed take-off rolling distance	$m_9 = \dfrac{S_{TOR}^*{}' - S'_{TOR}}{S_{TOR}^*{}'}$
	m_{10}	Computed margin between the required and computed landing rolling distance	$m_{10} = \dfrac{S_{LDR}^*{}' - S_{LDR}}{S_{LDR}^*{}'}$
	m_{11}	Computed margin between the required and computed sustained turn factor	$m_{11} = \dfrac{n_{ST} - n_{ST}^*{}'}{n_{ST}^*{}'}$

actively move the constraint, m_9 is computed as a result. A positive value of the latter will be able to accommodate unforeseen scenarios which require the aircraft to take off within a further shortened distance (e.g. a damaged runway).

Finally, a margin can be allocated without direct link(s) to any assumptions. This is exemplified by m_4 and m_5, which are allocated to account for discrepancy of computational results, while the assumptions are implicitly embedded in the corresponding models.

The complete definition of all the requirements, functions, components, models, variables, assumptions, and dependencies between those entities have led to 69 nodes and 199 edges in the DBN, which is illustrated as a Design Structure Matrix (DSM) in Figure 15 (in the Appendix). This DBN is used as a starting point for the assumption revision and margin trade-offs.

4.2. Initial design space exploration

After setting up the problem, an initial DoE study is performed to populate the design space of wing loading (*WoS*) and thrust-to-weight ratio (*ToW*). The initial values of all the margins are fixed at 10%. This value is selected arbitrarily for illustration purposes only. In a practical design case, the values could be based on statistical data or expert elicitation.

The study results are produced and visualised using an in-house software, AirCADia Explorer (Guenov et al. 2014a; 2014b). A screenshot of the output of the software is shown in Figure 16 (with enhanced font sizes for clarity). Figure 16(a) depicts the design space of *WoS*

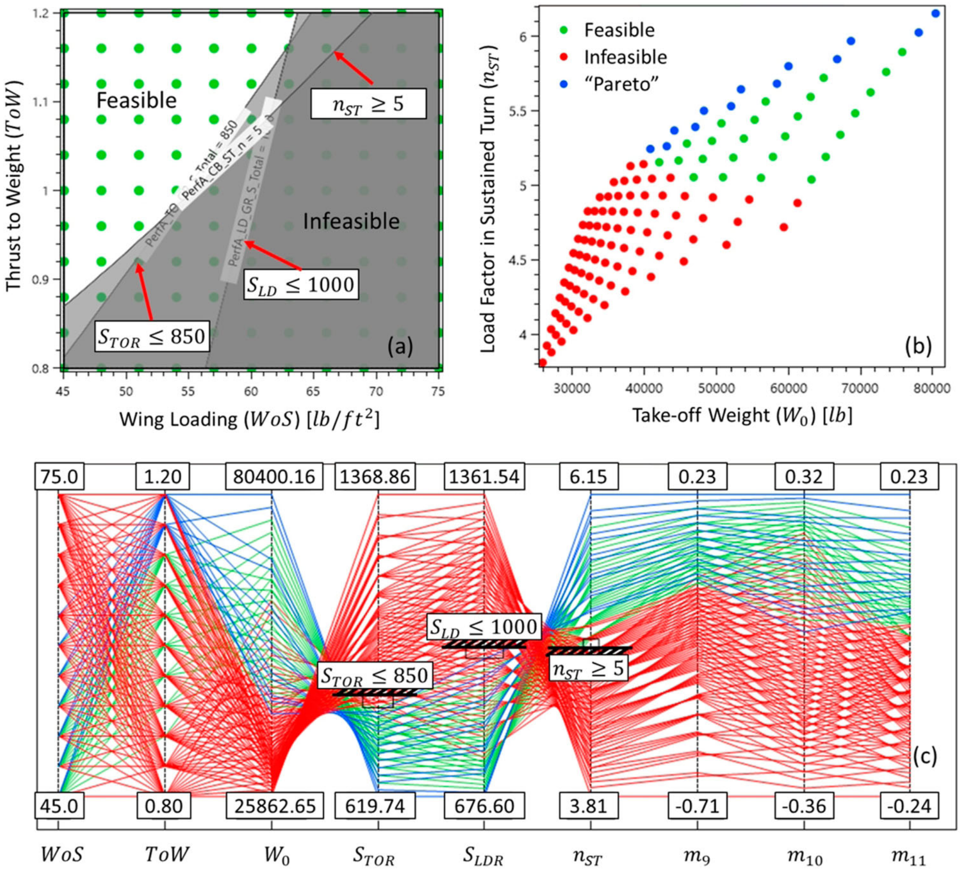

Figure 16. Initial design space exploration study results.

and ToW, where each green point is a design solution (a combination of WoS and ToW). The constraints are plotted as contour lines which delimit the feasible and infeasible regions in the design space (coloured in white and grey, respectively).

Shown in Figure 16(b) is a scatter plot of the performance space, showing the resulting take-off weights (W_0) and load factors in sustained turn (n_{ST}). Here, the feasible and infeasible design solutions are indicated by green and red points, respectively. The blue points are the 'Pareto' solutions (although this is not a strict optimisation study) with nondominated W_0 (lower the better) and n_{ST} (higher the better).

Figure 16(c) depicts a Parallel Coordinate Plot (PCP) (Inselberg 2009) of the high-dimensional space of all the variables, using the same colouring scheme as in Figure 16(b). In this plot, each vertical axis represents a variable while each polyline represents a single point in the multi-dimensional space. Specifically, the three axes on the rightmost side are the computed margins, as defined in Table 6.

4.3. Revision of assumptions and margin trade-offs

Consider an example scenario where new knowledge is introduced through a series of more comprehensive analyses (e.g. regarding aerodynamics, propulsion, weapon systems,

Table 7. Revision of assumptions.

Assumptions	Confidence level legacy	Confidence level updated
a_1: Zero lift drag coefficient	**Moderate**	**High**
a_2: The maximum lift coefficient in take-off and landing	**High**	**Moderate**
a_3: The baseline engine specific fuel consumption	**Moderate**	**Low**
a_4: The engine thrust lapse rate in combat condition	Moderate	Moderate
a_5: Expected lower fuel consumption due to more efficient engine	**Moderate**	**Low**
a_6: Expected required high power off-take due to advanced weapon system	Low	Low
a_7: Future requires higher payload	**Low**	**Moderate**
a_8: Future required combat range	Moderate	Moderate
a_9: Future required take-off rolling distance	Moderate	Moderate

and so forth), which leads to a revision of the assumptions made earlier. This process is represented by block 1 of the flowchart in Figure 3.

Specifically, the confidence levels of the assumed zero-lift drag coefficient (a_1) and expected future payload increment (a_7) have increased. Meanwhile, the confidence levels of the assumed maximum lift coefficient (a_2), baseline engine specific fuel consumption (a_3), and the expected reduction of the future engine specific fuel consumption (a_5) have decreased. The affected assumptions are emphasised in bold fonts in Table 7. The legacy (i.e. confidence levels prior to revision), and updated confidence levels are also provided.

For each of the affected assumptions, the algorithm proposed in Section 3.3 is applied to detect potential margin/assumption localisation paths which end with suitable margins (Block 2 in Figure 3). A complete summary of all the margin/assumption localisation paths is provided in Table 8 (see Appendix). In this case study, it indicates m_1 for a_1, m_6 for a_7, m_2 for both a_3 and a_5. Whilst these margins are directly linked to the corresponding assumptions, there are no margins specifically allocated for a_2, where the shortest margin/assumption localisation path leads to m_5.

After identification of the margins, a trade-off study (Block 3 in Figure 3) is performed by exploring different combinations of their values, using methods described in Section 3.4. For assumptions a_1 and a_7, the confidence levels have increased. Therefore, the corresponding margins m_1 and m_6 can be reduced (e.g. between 0% and 10%). For assumptions a_2, a_3, and a_5, which have reduced confidence levels, the corresponding margins m_2 and m_5 need to be increased (e.g. between 10% and 20%). The results of the margins design of experiment studies are shown in Figure 17. This figure illustrates the impact of margin revision on the feasible design space – the blue boxes indicate constraints which become less stringent, while the orange boxes represent constraints that become more stringent.

Figure 17(a) depicts the reference design space where all the margins are set with the initial value of 10%. In Figure 17(b), the margin on the zero-lift drag coefficient CD_0 (m_1 as defined in Table 6) has been reduced from 10% to 0% due to the increased confidence level on the corresponding assumption. This has led to slightly less stringent constraints on the sustained turn load factor and take-off rolling distance (due to lower drag), while the constraint on landing rolling distance has become more stringent (because it takes longer time for the aircraft to deaccelerate in landing due to lower drag).

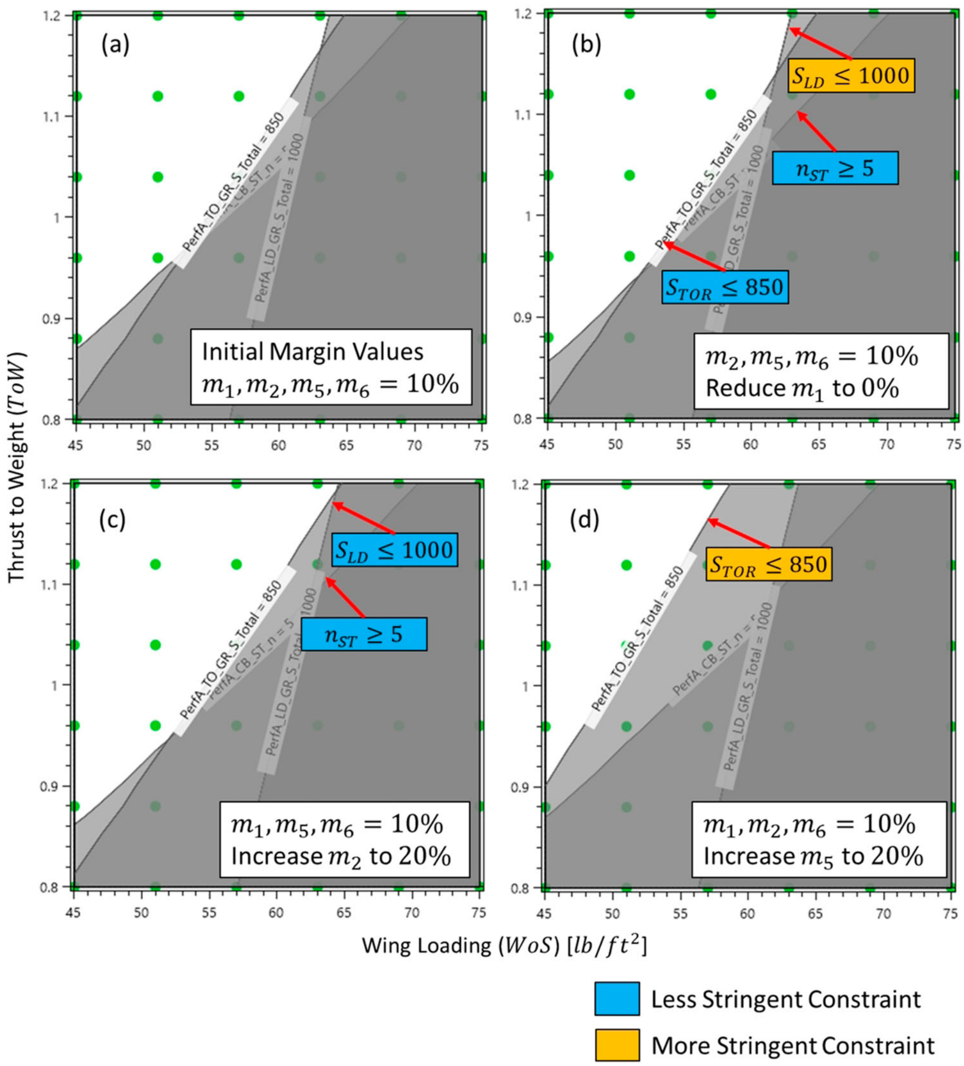

Figure 17. Margins trade-off studies.

In Figure 17(c), increasing margin m_2 (as defined in Table 6) on specific fuel consumption (*SFC*) from 10% to 20% has led to less stringent constraints on both the sustained turn load factor and landing distance. This is because more fuel is burned during the mission, and the weight fractions of the aircraft in combat and landing (compared with take-off) have become lower. There is no impact on the take-off rolling distance as the weight fraction in take-off is always 1. Note that the weight fractions should be distinguished from the actual take-off weight, which will become larger if the specific fuel consumption is higher (as more fuel needs to be carried).

Figure 17(d) shows the influence of increasing margin m_5 (as defined in Table 6) on the computational model for take-off rolling distance. By increasing this margin from 10% to 20% (to account for a less credible maximum lift coefficient in take-off), the take-off constraint has become substantially more stringent, which requires a higher thrust-to-weight ratio (larger engine) and lower wing loading (more wing area). The margin on the payload (i.e. m_6 on W_P^*) has no impact on the design space but will influence the actual take-off weight of the aircraft.

It can be seen that, in this specific example, the impact of margins on the design variables is not dramatic. This is because both design variables (wing loading and thrust-to-weight ratio) are ratios. The impact of the margins is more obvious on the 'Pareto Front' in the performance space as shown in Figure 18. Figure 18(a) is the reference performance space where the lower bounds of all the margins are set at 10%. The red points on the top-right corner of each figure are the invalid solutions of which the mass-fuel loops fail to converge. In Figure 18(b), only the lower bounds of m_1 and m_6 are reduced to 0%. In Figure 18(c), only the lower bounds of m_2 and m_5 are increased to 20%. In Figure 18(d), the lower bounds of all the margins are increased or decreased according to the confidence level changes of the corresponding assumptions. It can be seen that a higher value of the margins will shift the Pareto front to the lower right corner of the performance space (resulting in a heavier and less manoeuvrable aircraft). Such a shift in the Pareto front is expected when evolvability is considered – a consideration that typically leads to a product that is 'overdesigned' for its original purpose (Lim 2009; Long and Ferguson 2017; van Heerden, Guenov, and Molina-Cristóbal 2019).

5. Conclusions and future work

Presented in this paper is a novel framework for managing (containing) assumption-driven design change via margin allocation and trade-offs. One essential feature of the framework is that the relations between assumptions and margins are made explicit. The assumptions are used to fill knowledge gaps and may be revised periodically when new knowledge is introduced in the design process. This, in turn, could initiate a design change that needs to be contained, if possible, by existing margins in order to mitigate the impact.

In this regard, a novel algorithm was proposed to detect a list of potential margin/assumption localisation paths, which start from the revised assumption and end at different margins that can act as change absorbers. The algorithm also ranks the paths according to the type and distance of each assumption-margin path. In contrast to existing change propagation methods, which are normally restricted to the logical domain, the proposed approach traverses all domains in the R-F-L-C representation of engineering systems. Therefore, whether an assumption is related to a requirement, a solution, or any other element, a change absorber could be found by traversing the entire network. This capability is made possible by the dependencies between assumptions, margins, and other elements captured as part of the proposed DBN.

After identification of the most relevant margins (by the algorithm), interactive methods adapted from Guenov et al. (2018) are employed to trade off these margins according to the credibility of the revised assumptions, while maintaining satisfaction of all the design constraints and without over-compromise of the system performance. The proposed framework was demonstrated using a conceptual aircraft design use case to illustrate its practical value.

As this is a first attempt to integrate assumptions with margin management, a number of limitations were identified, which form plans for future work:

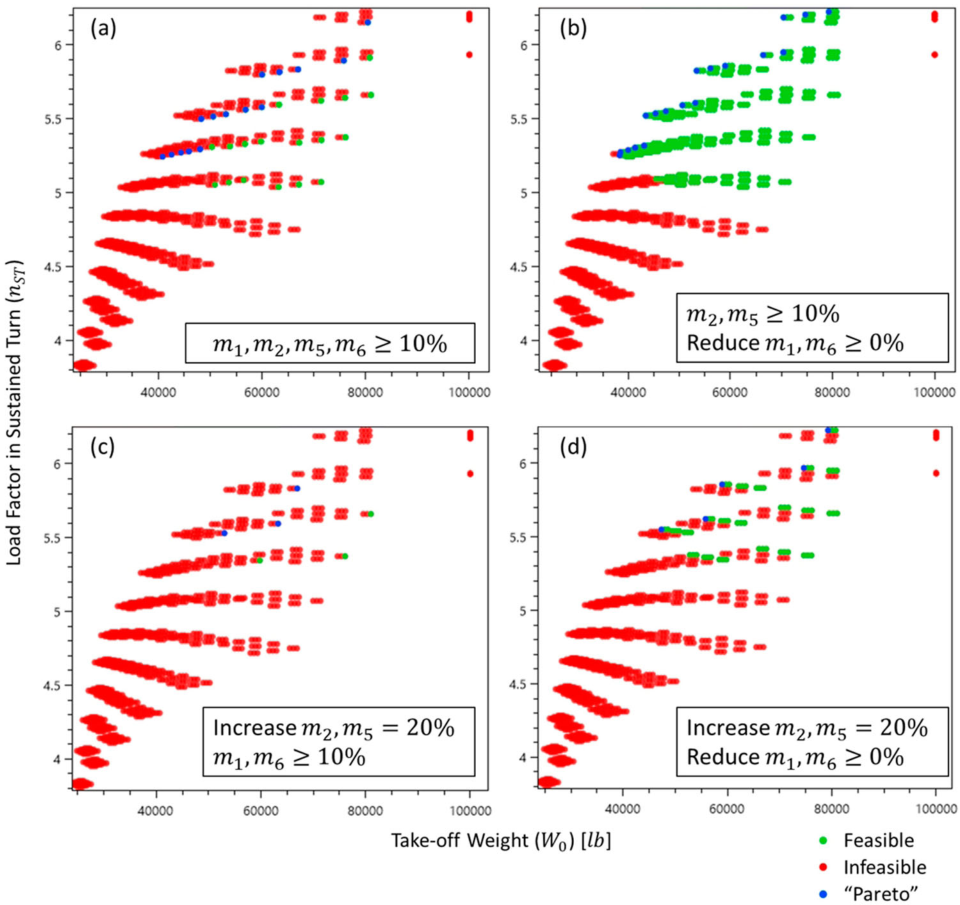

Figure 18. Impact of margins on the Pareto front location in the performance space.

- **Scalability**. The scalability of the proposed methods was not fully assessed, as only a limited number of assumptions and a simplified system architecture were covered in the demonstration process. For example, many concurrent changes in assumptions could potentially lead to conflicting suggestions for margin revision. To address this limitation, belief merging and judgment aggregation (Pigozzi 2021) appear to be promising approaches to aggregate conflicting margin revision suggestions into a consistent suggestion.
- **Extensibility**. The scope of the work presented here was confined to computational design at the conceptual stage of complex product development. However, as pointed out in (Koh 2017), change dependencies exist at different levels of abstraction across different domains. Furthermore, assumptions could be subject to change throughout the development process, up to the integration and test phases. If not managed properly, this introduces a potential risk of nugatory iterations due to late changes. In industry, such a risk is normally managed by a 'stage-gated' approach, augmented with maturity metrics (Harrison 2010). The purpose of these is to ensure that systems

mature in synchrony with each other. Therefore, the proposed approach and assumption management in general need to be aligned/integrated with the gated design review process.

- **Stochastic treatment of change propagation**. The proposed framework aims to accommodate a change (initiated by a revised assumption) before it propagates. If this change cannot be absorbed by the existing margins, then change propagation methods can be applied as a further action. For example, the Change Absorber Localisation method presented could potentially be extended to include the likelihood of changes propagating from one specific component/subsystem to another, or the likelihood of a specific margin to fully absorb a change. Although it may not always be possible to estimate such likelihoods, one avenue that could be worth exploring is the use of historical data or expert opinion. For instance, in Clarkson, Simons, and Eckert (2004), propagation likelihoods between sub-systems were elicited from deputy chief engineers involved in developing the EH101 helicopter. Such a stochastic treatment would allow capturing the uncertainty inherent in change propagation, which is currently missing in the proposed (deterministic) approach.

Data availability

The design of experiment study results as represented in Figures 16–18 can be accessed through the Cranfield Online Research Data (CORD) repository system using the following link: DOI:10.17862/cranfield.rd.23695158.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Allen, Jeffrey D., Christopher A. Mattson, and Scott M. Ferguson. 2016. "Evaluation of System Evolvability Based on Usable Excess." *Journal of Mechanical Design* 138 (9). https://doi.org/10.1115/1.4033989.

Altelarrea, Sergio Jimeno. 2021. "Building Safety into the Conceptual Design of Complex Systems. An Aircraft Systems Perspective." PhD diss., Cranfield University.

Berner, C. L. 2017. "Contributions to Improved Risk Assessments: To Better Reflect the Strength of Background Knowledge." PhD diss., University of Stavanger.

Bile, Yogesh, Atif Riaz, Marin D. Guenov, and Arturo Molina-Cristóbal. 2018. "Towards Automating the Sizing Process in Conceptual (Airframe) Systems Architecting." In *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Kissimmee, FL: AIAA. https://doi.org/10.2514/6.2018-1067.

Brahma, Arindam, Scott Ferguson, Claudia Eckert, and Ola Isaksson. 2023. "Margins in Design – Review of Related Concepts and Methods." *Journal of Engineering Design*, 1–34. https://doi.org/10.1080/09544828.2023.2225842.

Brahma, Arindam, and David C. Wynn. 2020. "Margin Value Method for Engineering Design Improvement." *Research in Engineering Design* 31 (3): 353–381. https://doi.org/10.1007/s00163-020-00335-8.

Brahma, Arindam, and David C. Wynn. 2021. "A Study on the Mechanisms of Change Propagation in Mechanical Design." *Journal of Mechanical Design* 143 (12): 121401. https://doi.org/10.1115/1.4050927.

Brahma, Arindam, and David C. Wynn. 2022. "Concepts of Change Propagation Analysis in Engineering Design." *Research in Engineering Design* 34:117–151. https://doi.org/10.1007/s00163-022-00395-y.

Brahma, Arindam, David C. Wynn, and Ola Isaksson. 2022. "Use of Margin to Absorb Variation in Design Specifications: An Analysis Using the Margin Value Method." *Proceedings of the Design Society* 2:323–332. https://doi.org/10.1017/pds.2022.34.

Brown, David C. 2006. "Assumptions in Design and Design Rationale." In *DCC' 06 Workshop on Design Rationale: Problems and Progress. Design Computing and Cognition*.

Chen, Xin, Adriano Isoldi, Atif Riaz, Christos Mourouzidis, Akin Keskin, Dale Smith, Marin D. Guenov, and Vassilios Pachidis. 2023. "Evaluation of a Collaborative and Distributed Aircraft Design Environment, Enabled by Microservices and Cloud Computing." In *AIAA SCITECH 2023 Forum*. National Harbor, MD: AIAA. https://doi.org/10.2514/6.2023-1163.

Clarkson, *P*. John, Caroline Simons, and Claudia Eckert. 2004. "Predicting Change Propagation in Complex Design." *Journal of Mechanical Design* 126 (5): 788–797. https://doi.org/10.1115/1.1765117.

Cooke, Roger M., Thomas A. Zang, Dimitri N. Mavris, and Jimmy C. Tai. 2015. "Sculpting: A Fast, Interactive Method for Probabilistic Design Space Exploration and Margin Allocation." In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Dallas, TX: AIAA. https://doi.org/10.2514/6.2015-3440.

Crawley, Edward, Bruce Cameron, and Daniel Selva. 2016. *Systems Architecture: Strategy and Product Development for Complex Systems*. 1st ed. Hoboken, NJ: Pearson.

de Weck, Olivier L., Adam Michael Ross, and Donna H. Rhodes. 2012. "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)." *Third International Engineering Systems Symposium CESUN* 2012:1–12.

Dijkstra, E. W. 1959. "A Note on Two Problems in Connexion with Graphs." *Numerische Mathematik* 1 (1): 269–271. https://doi.org/10.1007/BF01386390.

Eckert, Claudia, P. John Clarkson, and Winfried Zanker. 2004. "Change and Customisation in Complex Engineering Domains." *Research in Engineering Design* 15 (1): 1–21. https://doi.org/10.1007/s00163-003-0031-7.

Eckert, Claudia, Chris Earl, Safaa Lebjioui, and Ola Isaksson. 2013. "Components Margins through the Product Lifecycle." In *IFIP Advances in Information and Communication Technology*, edited by Alain Bernard, Louis Rivest, and Debasish Dutta, 39–47. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-41501-2_5.

Eckert, Claudia, Ola Isaksson, and Chris Earl. 2019. "Design Margins: A Hidden Issue in Industry." *Design Science* 5:e9. https://doi.org/10.1017/dsj.2019.7.

Eger, Tido, Claudia Eckert, and *P* John Clarkson. 2005. "The Role of Design Freeze in Product Development." In *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*. Melbourne: Engineers Australia. https://search.informit.org/doi/10.3316informit.383909035227839.

El Fassi, Soufiane. 2021. "Assumption Management in Model-based Systems Engineering: An Aircraft Design Perspective." PhD diss., Cranfield University.

El Fassi, Soufiane, Marin D. Guenov, and A. Riaz. 2020. "An Assumption Network-Based Approach to Support Margin Allocation and Management." In *Proceedings of the Design Society: DESIGN Conference*, Vol. 1, 2275–2284. Cambridge University Press. https://doi.org/10.1017/dsd.2020.25.

Guenov, Marin D., Xin Chen, Arturo Molina-Cristóbal, Atif Riaz, Albert S.J. Van Heerden, and Mattia Padulo. 2018. "Margin Allocation and Tradeoff in Complex Systems Design and Optimization." *AIAA Journal* 56 (7): 2887–2902. https://doi.org/10.2514/1.J056357.

Guenov, Marin D., Marco Nunez, Arturo Molina-Cristóbal, Varun Datta, and Atif Riaz. 2014a. "Aircadia – An Interactive Tool for the Composition and Exploration of Aircraft Computational Studies at Early Design Stage." In *In the 29th Congress of the International Council of the Aeronautical Sciences*. St Petersburg, Russia: ICAS.

Guenov, Marin. D., M. Nunez, A. Molina-Cristóbal, V. Sripawadkul, V. C. Datta, and A. Riaz. 2014b. "Composition, Management, and Exploration of Computational Studies at Early Design Stage." In *Computational Intelligence in Aerospace Sciences, Progress in Astronautics and Aeronautics*, edited by Massimiliano Vasile and Victor M. Becerra, 415–460. Reston: AIAA.

Guenov, Marin D., Atif Riaz, Yogesh H. Bile, Arturo Molina-Cristobal, and Albert S.J. van Heerden. 2020. “Computational Framework for Interactive Architecting of Complex Systems.” *Systems Engineering* 23 (3): 350–365. https://doi.org/10.1002/sys.21531.

Hall, Kyle, Peter Schroll, and Sanjiv Sharma. 2020. “Managing Margins under Uncertainties Surrogate Modelling and Uncertainty Quantification.” In *Complex Systems Design & Management*, edited by Guy André Boy, Alan Guegan, Daniel Krob, and Vincent Vion, 49–63. Cham: Springer. https://doi.org/10.1007/978-3-030-34843-4_5

Harrison, N. 2010. “Warship Design Maturity Measurement and Analysis Throughout the Lifecycle.” In *International Conference on Systems Engineering in Ship and Offshore Design*. Bath.

Inselberg, Alfred. 2009. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. New York: Springer.

Jenkins, D., R. Woolston, and M. Boyd. 2019. “Designing for Uncertainty (Assumption-Based Design).” *Newdesign*, June 5, 2019. no. 138:22–25. https://www.dca-design.com/latest/designing-uncertainty-assumption-based-design

Jimeno Altelarrea, Sergio, Atif Riaz, Marin D. Guenov, and Arturo Molina-Cristobal. 2020. “Enabling Interactive Safety and Performance Trade-offs in Early Airframe Systems Design.” In *AIAA Scitech 2020 Forum*. Orlando, FL: AIAA. https://doi.org/10.2514/6.2020-0550.

Kleiner, Sven, and Christoph Kramer. 2013. “Model Based Design with Systems Engineering Based on RFLP Using V6.” In *Smart Product Engineering. Lecture Notes in Production Engineering*, edited by Michael Abramovici and Rainer Stark, 93–102. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-30817-8_10.

Koh, Edwin C.Y. 2017. “A Study on the Requirements to Support the Accurate Prediction of Engineering Change Propagation.” *Systems Engineering* 20 (2): 147–157. https://doi.org/10.1002/sys.21385.

Koh, Edwin C.Y., Nicholas H.M. Caldwell, and *P*. John Clarkson. 2013. “A Technique to Assess the Changeability of Complex Engineering Systems.” *Journal of Engineering Design* 24 (7): 477–498. https://doi.org/10.1080/09544828.2013.769207.

Lim, D. 2009. “A Systematic Approach to Design for Lifelong Aircraft Evolution.” PhD diss., Georgia Institute of Technology.

Long, Daniel, and Scott Ferguson. 2017. “A Case Study of Evolvability and Excess on the B-52 Stratofortress and F/A-18 Hornet.” In *Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Cleveland, OH, USA. https://doi.org/10.1115/DETC2017-68287.

Long, Daniel, and Scott Ferguson. 2020. “Studying Dynamic Change Probabilities and their Role in Change Propagation.” *Journal of Mechanical Design* 142 (10): 101701. https://doi.org/10.1115/1.4046674.

Mattingly, Jack D., William H. Heiser, Keith M. Boyer, Brenda A. Haven, and David T. Pratt. 2018. *Aircraft Engine Design*. 3rd ed. AIAA, Inc. https://doi.org/10.2514/4.105173.

Mavris, Dimitri N., and Daniel A. DeLaurentis. 2000. “A Probabilistic Approach for Examining Aircraft Concept Feasibility and Viability.” *Aircraft Design* 3 (2): 79–101. https://doi.org/10.1016/S1369-8869(00)00008-2.

Michel, Ghosn, and Moses Fred. 1986. “Reliability Calibration of Bridge Design Code.” *Journal of Structural Engineering* 112 (4): 745–763. https://doi.org/10.1061/(ASCE)0733-9445(1986)112:4(745).

Mohammed, E. Alfred, S. D. Benson, S. E. Hirdaris, and R. S. Dow. 2016. “Design Safety Margin of a 10,000 TEU Container Ship through Ultimate Hull Girder Load Combination Analysis.” *Marine Structures* 46:78–101. https://doi.org/10.1016/j.marstruc.2015.12.003.

Ostacchini, Ireo, and Michel Wermelinger. 2009. “Managing Assumptions during Agile Development.” In *Proceedings of the 2009 ICSE Workshop on Sharing and Reusing Architectural Knowledge*. Vancouver, Canada. https://doi.org/10.1109/SHARK.2009.5069110.

Pigozzi, Gabriella. 2021. "Belief Merging and Judgment Aggregation.” *The Stanford Encyclopedia of Philosophy,* July 8, 2015. https://plato.stanford.edu/entries/belief-merging/.

Raymer, Daniel P. 2018. *Aircraft Design: A Conceptual Approach*. 6th ed. Reston: AIAA. https://doi.org/10.2514/4.104909.

Saoub, K. R. 2017. *A Tour through Graph Theory*. 1st Editio. Boca Raton: Chapman and Hall/CRC. https://doi.org/10.1201/9781315116839

Society of Automotive Engineers. 2010. *SAE ARP4754A: Guidelines for Development of Civil Aircraft and Systems*.

Tackett, Morgan W. P., Christopher A. Mattson, and Scott M. Ferguson. 2014. "A Model for Quantifying System Evolvability Based on Excess and Capacity." *Journal of Mechanical Design* 136 (5): 051002. https://doi.org/10.1115/1.4026648.

Thunnissen, Daniel P. 2004. "Method for Determining Margins in Conceptual Design." *Journal of Spacecraft and Rockets* 41 (1): 85–92. https://doi.org/10.2514/1.9211.

Touboul, A., J. Reygner, F. Mangeant, and *P*. Benjamin. 2019. *A Formal Framework to Define Margins in Industrial Processes*. HAL open science. https://hal.science/hal-02156493/.

Umeda, Y., and T. Tomiyama. 1997. "Functional Reasoning in Design." *IEEE Expert* 12 (2): 42–48. https://doi.org/10.1109/64.585103.

van Heerden, Albert S. J., Marin D. Guenov, and Arturo Molina-Cristóbal. 2019. "Evolvability and Design Reuse in Civil Jet Transport Aircraft." *Progress in Aerospace Sciences* 108:121–155. https://doi.org/10.1016/j.paerosci.2019.01.006.

Verein Deutscher Ingenieure. 2004. "VDI 2206: Design Methodology for Mechatronic Systems." 2004-06. Verein Deutscher Ingenieure.

Yang, Chen, Peng Liang, and Paris Avgeriou. 2018. "Evaluation of a Process for Architectural Assumption Management in Software Development." *Science of Computer Programming* 168 (December): 38–70. https://doi.org/10.1016/j.scico.2018.08.002.

Yuan, J., M. Savill, T. Kipouros, S. Coggon, and F. Scarpa. 2016. "Probabilistic Margin Assessment of Aircraft Conceptual Design Using a Modified Reliability Based Design Optimization Methodology." In *Proceedings of Isma2016 International Conference on Noise and Vibration Engineering and USD2016 International Conference on Uncertainty in Structural Dynamics*, 4463–4477.

Zang, Thomas A., Sankaran Mahadevan, Jimmy C. Tai, and Dimitri N. Mavris. 2015. "A Strategy for Probabilistic Margin Allocation in Aircraft Conceptual Design." In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Dallas, TX: AIAA. https://doi.org/10.2514/6.2015-3443

Zhuravlev, Pavel, and Vladimir Zhuravlev. 2012. "Significance of Modifications for Development of Passenger Airplanes." *Aircraft Engineering and Aerospace Technology* 84 (3): 172–180. https://doi.org/10.1108/00022661211222021

Appendix

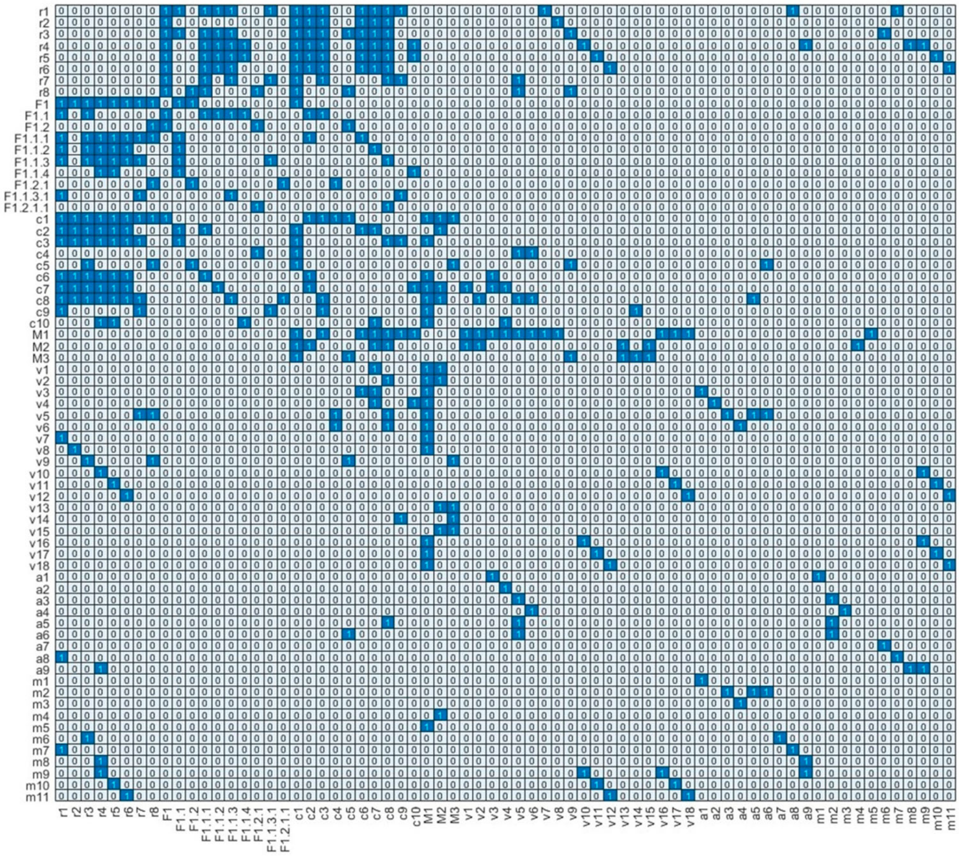

Figure 15. The example use-case Design Belief Network, stored as a design structure matrix.

Table 8. Complete list of all the margin/assumption localisation paths.

Revised Assumption	Distance	Path	Type
a1	2	a1, m1.	Direct
	4	a1, v3, M1, m5.	Undetermined
	5	a1, v3, c7, M2, m4.	Elementary
	5	a1, v3, c6, r3, m6.	Elementary
	5	a1, v3, c6, a1, m7.	Elementary
	5	a1, v3, c6, r4, m8.	Elementary
	5	a1, v3, c6, r4, m9.	Elementary
	5	a1, v3, c6, r5, m10.	Elementary
	5	a1, v3, c6, r6, m11.	Elementary
	6	a1, v3, M1, c8, a5, m2.	Elementary
	6	a1, v3, M1, v6, a4, m3.	Undetermined
a2	4	a2, v4, M1, m5.	Undetermined
	5	a2, v4, c7, M2, m4.	Elementary
	5	a2, v4, c7, r3, m6.	Elementary
	5	a2, v4, c7, a2, m7.	Elementary
	5	a2, v4, c7, r4, m8.	Elementary
	5	a2, v4, c7, r4, m9.	Elementary
	5	a2, v4, c7, r5, m10.	Elementary
	5	a2, v4, c7, r6, m11.	Elementary
	6	a2, v4, c7, v3, a1, m1.	Elementary
	6	a2, v4, M1, c8, a5, m2.	Elementary
	6	a2, v4, M1, v6, a4, m3.	Undetermined
a3	2	a3, m2.	Direct
	4	a3, v5, M1, m5.	Undetermined
	5	a3, v5, c8, M2, m4.	Elementary
	5	a3, v5, c8, r3, m6.	Elementary
	5	a3, v5, c8, a3, m7.	Elementary
	5	a3, v5, c8, r4, m8.	Elementary
	5	a3, v5, c8, r4, m9.	Elementary
	5	a3, v5, c8, r5, m10.	Elementary
	5	a3, v5, c8, r6, m11.	Elementary
	6	a3, v5, M1, v3, a1, m1.	Undetermined
	6	a3, v5, c4, v6, a4, m3.	Elementary
a5	2	a5, m2.	Direct
	4	a5, c8, M2, m4.	Elementary
	4	a5, c8, M1, m5.	Elementary
	4	a5, c8, r3, m6.	Elementary
	4	a5, c8, a5, m7.	Elementary
	4	a5, c8, r4, m8.	Elementary
	4	a5, c8, r4, m9.	Elementary
	4	a5, c8, r5, m10.	Elementary
	4	a5, c8, r6, m11.	Elementary
	5	a5, c8, v6, a4, m3.	Elementary
	6	a5, c8, M1, v3, a1, m1.	Elementary
a7	2	a7, m6.	Direct
	6	a7, m6, r3, c5, a6, m2.	Elementary
	6	a7, m6, r3, c1, M2, m4.	Elementary
	6	a7, m6, r3, c1, M1, m5.	Elementary
	6	a7, m6, r3, F1, a7, m7.	Undetermined
	6	a7, m6, r3, F1, r4, m8.	Undetermined
	6	a7, m6, r3, F1, r4, m9.	Undetermined
	6	a7, m6, r3, F1, r5, m10.	Undetermined
	6	a7, m6, r3, F1, r6, m11.	Undetermined
	7	a7, m6, r3, c6, v3, a1, m1.	Elementary
	7	a7, m6, r3, c8, v6, a4, m3.	Elementary