designs

MDPI

Article

Coupling Structures in Model-Based Systems: An Analytical Framework Based on Axiomatic Design

Julian Borowski [1,2,*], Felix Löser [1] and Stephan Rudolph [3]

[1] IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH, 70771 Leinfelden-Echterdingen, Germany; felix.loeser@iils.de
[2] Institute for Innovative Teaching and Applied Research on Teaching and Learning/Department of Mechanical Engineering, Ravensburg-Weingarten University of Applied Sciences (RWU), Doggenriedstraße 70, 88250 Weingarten, Germany
[3] Design Theory and Similarity Mechanics Group, Institute of Aircraft Design, University of Stuttgart, 70569 Stuttgart, Germany; rudolph@ifb.uni-stuttgart.de
* Correspondence: julian.borowski@iils.de or julian.borowski@rwu.de

Abstract

The engineering design of complex systems increasingly relies on model-based methods that generate large-scale models with a vast multitude of couplings between the functional requirements and design parameters. The so-called Axiomatic Design Approach tries to characterize and evaluate the couplings between functional requirements (*FR*s) and design parameters (*DP*s) by imposing a square matrix form for the mapping. In full generality however, the physical modeling of complex systems produces rectangular matrices due to an a priori unequal number of *FR*s and *DP*s, as dictated by physical laws and already shown with mathematical rigour in previous works. By using the full rectangular form of the coupling matrix, this paper presents an analytical framework that uses classical sensitivity-based analysis to identify, quantify, and visualize coupling structures represented by rectangular design matrices. Normalized visualizations of the Jacobian matrix in the form of so-called heatmaps then help to reveal intrinsic subsystem structures and highlight areas of strong internal couplings and minimal inter-subsystem interactions. The heatmap thus helps to detect weakly coupled subsystem blocks, supports a more systematic decomposition and allows a targeted optimization in a loose analogy to the original idea of the so-called Independence Axiom in axiomatic design. This methodology permits the sequential or even independent optimization of subsystems or parts thereof, facilitating a reliable and efficient system-level design. The proposed approach is demonstrated through two applications: a Formula Student race car and a small satellite design. The case studies illustrate how rectangular matrix analysis can provide a practical basis for guiding system decomposition and design decisions in model-based engineering systems.

Keywords: axiomatic design; rectangular design matrices; coupling analysis; sensitivity analysis; system decomposition; model-based systems engineering

Check for updates

Academic Editor: Xu-Yang Cao

Received: 9 September 2026
Revised: 13 October 2026
Accepted: 24 October 2026
Published: 29 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The design and optimization of complex engineering systems increasingly relies on digitally integrated, model-based development processes [1]. In such settings, large-scale simulation models are used to explore design spaces and automate optimization, but they also generate complex systems of equations whose internal coupling structures are difficult to analyze and control [2]. Unintended couplings between functional requirements and

design parameters reduce robustness, hinder modularization, and complicate system-level optimization [3]. The concept of Axiomatic Design [4] provides a formal framework for addressing coupling through the so-called Independence Axiom, which requires that functional requirements be satisfied independently. In its classical formulation, this principle is evaluated using square design matrices that map functional requirements to design parameters. A diagonal or triangular matrix form corresponds then to uncoupled or decoupled designs, while fully populated matrices indicate coupled designs [5].

Although effective at the conceptual level, this formulation is limited when applied to model-based engineering problems [6]. In practice, sensitivity analyses of physical systems typically yield in full generality rectangular matrices that relate system characteristics to design parameters [7]. Dimensional homogeneity imposes constraints on the number of independent dimensionless quantities that can describe a physical system, as formalized by the Buckingham Π-Theorem [8]. Building on this dimensional-analysis result, Rudolph [6] shows that the corresponding general mapping between the evaluation space and the physical design-parameter space is generally rectangular rather than square and therefore represents the upper limit of the matrix representation. Thus, classical Axiomatic Design offers limited guidance for interpreting coupling and independence under such restrictive conditions [9].

This paper proposes an analytical framework for coupling analysis based on rectangular design matrices. Inspired by the concept of Axiomatic Design, the framework employs sensitivity-based analysis to quantify and uses normalized Jacobian matrices to visualize couplings as heatmaps, identify weakly coupled rectangular subsystem structures, and support systematic system decomposition [10]. The approach enables sequential or independent optimization of subsystems by exploiting decoupled block structures that reflect intrinsic subsystem organizations, thereby addressing the mathematical challenges associated with inherently rectangular design matrices in classical Axiomatic Design [11].

While sensitivity analysis and Jacobian-based system investigations are well established in engineering design, their interpretation within the framework of Axiomatic Design remains largely limited to the classical square design matrix formulation [10]. The contribution of this paper is therefore not the introduction of a new sensitivity analysis method itself, but the systematic integration of rectangular Jacobian matrices into the concept of Axiomatic Design. By interpreting sorted sensitivity structures as weakly coupled subsystem blocks, the proposed approach provides the mathematical foundation and practical extension of the Independence Axiom for model-based engineering systems in which the number of design parameters and functional requirements generally differ. The present work therefore builds explicitly on several elements developed in earlier studies. The sensitivity-based suspension analysis and the associated Formula Student results were previously investigated in [10,11], while the normalized sensitivity representation and matrix-sorting approach were developed and discussed in [11,12]. Likewise, the FireSAT design language and parts of its underlying model originate from earlier work [13,14]. These previous contributions are used here as methodological and application foundations rather than as new results of the present study. The novel contribution of this paper is their integration into a common analytical framework that interprets the full rectangular Jacobian as a generalized design matrix within Axiomatic Design, explicitly relates its coupling structure to the Independence Axiom, and derives a consistent decomposition and optimization rationale across two different model-based engineering applications. The case studies are therefore used primarily to demonstrate and contextualize this integrated framework rather than to claim the underlying engineering models or individual sensitivity results as new.

Related approaches for analyzing dependencies in engineering systems include sensitivity-based methods, Jacobian analysis, and Design Structure Matrices (DSM) [15–17].

DSM methods are widely used to represent and organize dependencies between system elements and to support modularization and decomposition of complex products and processes [17]. Likewise, sensitivity analysis and Jacobian matrices are established tools for quantifying the influence of design variables on system responses [7]. The present work does not introduce these methods themselves but interprets the full rectangular Jacobian matrix as a generalized design matrix within Axiomatic Design and uses the resulting coupling structure to identify weakly coupled subsystem blocks and support decomposition and optimization planning. Therefore, the framework is intended for model-based engineering problems where quantitative system models are available and sensitivities can be computed with respect to continuous design variables. Typical applications include simulation-driven design, multidisciplinary optimization, and model-based systems engineering. The approach is not intended to replace methods for discrete architecture selection or topological design synthesis, but rather to support decomposition and optimization once a model structure and a corresponding system of equations have been established.

The remainder of the paper is structured as follows. Section 2 reviews the fundamentals of Axiomatic Design. Section 3 analyzes coupling structures in model-based systems. Section 4 presents the proposed framework, followed by case studies in Sections 5 and 6. The paper concludes with a discussion and conclusions.

2. Axiomatic Design and Design Matrices

This chapter briefly reviews the fundamentals of Axiomatic Design, the definitions of functional requirements (FRs) and design parameters (DPs), and the concept of design matrices. These concepts form the basis for the framework and analyses presented in the following sections.

2.1. Fundamentals of Axiomatic Design

Axiomatic Design provides a formal framework for reasoning about engineering design decisions through a small set of fundamental axioms [4]. The theory aims to establish objective criteria for evaluating and comparing design alternatives, independent of domain-specific techniques [18]. At its core, Axiomatic Design distinguishes between different domains of the design process and formalizes the mappings between them. In its classical formulation, Axiomatic Design considers four domains: the customer domain, the functional domain, the physical domain, and the process domain, as illustrated in Figure 1.

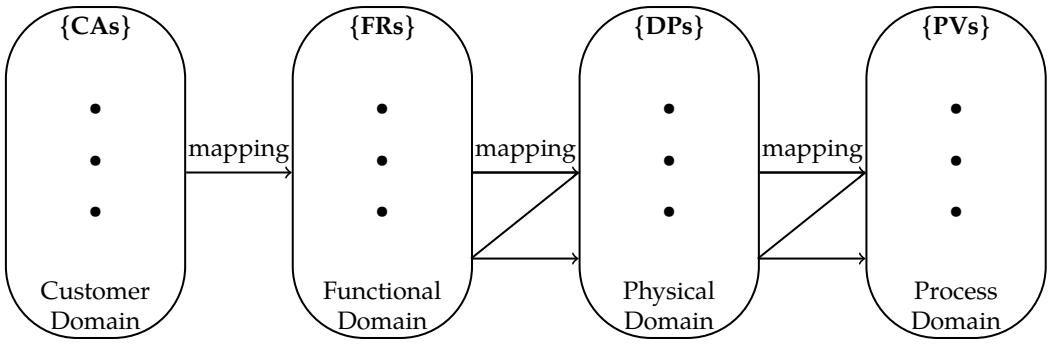


Figure 1. Four domains of Axiomatic Design with primary mappings from left to right [4].

Design activities are structured as mappings between these domains, most notably between the functional domain, where the desired system behavior is specified, and the physical domain, where design solutions are realized [19]. This structured decomposition allows designers to explicitly consider cause-and-effect relationships within complex systems. The theory is governed by two axioms. The Independence Axiom states that

the functional requirements of a design should remain independent of one another. The Information Axiom states that, among designs that satisfy the Independence Axiom, the one with the minimum information content should be selected [4]. In practice, the Independence Axiom plays a dominant role in early-stage system design, as it directly constrains the structure of the design solution, while the Information Axiom is typically applied at later stages once feasible design alternatives have been identified [20].

By framing design as a mapping between requirements and solutions, Axiomatic Design provides a formal basis for analyzing complexity, coupling, and robustness in engineered systems. This makes it particularly suitable for applications in model-based and system-level design contexts [21].

2.2. Functional Requirements and Design Parameters

Within Axiomatic Design, functional requirements (*FR*s) and design parameters (*DP*s) form the fundamental variables of the design process. Functional requirements describe what the system must achieve and are defined in the functional domain, whereas design parameters describe how these requirements are physically realized in the physical domain [4].

A key principle of Axiomatic Design is the hierarchical decomposition of functional requirements and design parameters. High-level system requirements are decomposed into lower-level *FR*s, which are mapped to corresponding *DP*s at the same level of abstraction. This process of 'zigzagging' between the functional and physical domains ensures traceability and consistency throughout the design hierarchy [22]. Crucially, functional requirements are defined as solution-independent. They must not prescribe specific physical implementations, as doing so would prematurely constrain the design space and potentially violate the Independence Axiom. Design parameters, in contrast, are explicitly solution-specific and represent design decisions that can be adjusted or substituted during the design process [19]. The classical formulation of Axiomatic Design implicitly assumes a one-to-one correspondence between *FR*s and *DP*s at a given level of decomposition. While this assumption simplifies analysis and supports the evaluation of independence, it is often violated in real-world systems, where the number of functional requirements and available design parameters may differ [20]. This observation motivates a more general treatment of the *FR* and *DP* relationship, which is addressed in later sections of this paper.

2.3. Design Matrices and the Independence Axiom

The relationship between functional requirements and design parameters is formally expressed through the design equation [4]:

$$\{FR\} = [A]\{DP\} \tag{1}$$

where $\{FR\}$ is the vector of functional requirements, $\{DP\}$ is the vector of design parameters, and $[A]$ is the design matrix. Each element A_{ij} indicates whether and to what extent the design parameter DP_j influences the functional requirement FR_i. The structure of the design matrix provides a means of assessing compliance with the Independence Axiom. If the matrix is diagonal, each functional requirement is affected by one design parameter, and the design is classified as uncoupled. If the matrix is triangular, the design is decoupled, meaning that the *FR*s can be satisfied independently through a specific sequence of *DP* adjustments [4]. Fully populated matrices correspond to coupled designs, in which changes to one *DP* affect multiple *FR*s simultaneously, thereby violating the Independence Axiom [19]. Figure 2 illustrates the three typical cases:

Uncoupled **Decoupled** **Coupled**

$$\begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \begin{bmatrix} X & X & X \\ 0 & X & X \\ 0 & 0 & X \end{bmatrix} \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix}$$

Figure 2. Different design matrix structures in Axiomatic Design [4].

In classical Axiomatic Design, design matrices are typically square, reflecting the one-to-one correspondence between *FR*s and *DP*s. However, in many real-world and model-based systems, the number of functional requirements does not match the number of available design parameters [23]. Considering design matrices in their full generality is therefore essential for extending Axiomatic Design to complex engineered systems, as will be discussed in the following sections.

3. Coupling Structures in Model-Based Systems

The design, development, and optimization of complex engineering systems, such as modern vehicle suspension systems, constitute an inherently complex process spanning multiple engineering disciplines and involving several design loops [10]. Automating this iterative process through digital and model-based methods typically generates correspondingly large systems of equations. The structural analysis of couplings within these systems is critical for guiding the design process to maximize the independence of functional requirements, as mandated by the principles of Axiomatic Design [4].

The strength of interactions between design parameters (*DP*s, e.g., suspension coordinates of a vehicle) and system characteristics of the design (in this context referred to as *K*, e.g., camber or toe angle changes) can be systematically assessed using sensitivity analysis [11]. This is commonly represented via the Jacobian matrix (J), which encapsulates the normalized derivatives of the system characteristics with respect to the design parameters [10], where each entry J_{ij} reflects the change in the i-th characteristic (K_i) relative to the j-th design parameter (DP_j).

In the context of this work, the Jacobian matrix can be understood as a compact representation of how strongly system characteristics respond to changes in design parameters. Each matrix entry represents a local sensitivity and therefore quantifies the influence of a particular design parameter on a particular system characteristic [10]. Rather than introducing a new mathematical construct, the Jacobian matrix serves as an established engineering tool for capturing and organizing coupling information in a form that can subsequently be interpreted through the principles of Axiomatic Design [11].

3.1. Limitations of Square Design Matrices

The fundamental guiding principle in Axiomatic Design is the Independence Axiom, which requires that the functional requirements of a system must be maintained independently of each other [19]. Ideally, an engineering design should ensure that each functional requirement is satisfied by a dedicated design parameter without causing undesirable interactions with other requirements [4].

In mathematical terms, Axiomatic Design postulates that this functional independence is represented by a design matrix that must be either diagonal or triangular, as illustrated in Figure 2. A diagonal matrix is characteristic of an uncoupled design, where each *FR* is satisfied independently by exactly one *DP* [6]. A triangular matrix represents a decoupled design, guaranteeing the independence of *FR*s only if the *DP*s are adjusted in a specified sequence [24].

However, the assumption of square matrices in the strict application of Axiomatic Design theory is a subject of controversy in specialized literature. The primary criticism is

that the design matrix J derived from the sensitivity analysis, which describes the interaction between design parameters and system behavior, is inherently rectangular in its full generality, not square [6].

3.2. Rectangular Design Matrices in Full Generality

The general rectangular form of the design matrix is explained by the fundamental laws of physics, specifically the Buckingham Π-Theorem [8]. This theorem establishes that when mapping n dimensional variables (x_n) onto m dimensionless invariants (π_m), the condition $m \leq n$ must always hold [8]. Rudolph [6] subsequently applies this dimensional analysis framework to the mapping between the evaluation space and the physical design parameter space and shows that the corresponding general matrix is rectangular rather than square. Thus, the rectangular matrix represents the general upper limit of the mathematical representation, while a square matrix results as a special case when an appropriate number of design parameters is held constant [6]. Consequently, the resultant Jacobian matrix (J), shown in Figure 3, is typically wider than it is tall, thereby providing the 'true upper limit' for the mathematical representation rooted in the requirement of dimensional homogeneity underlying all physical modeling [6].

Rectangular Design Matrix J with Subsystem Blocks

$$\begin{bmatrix} X & X & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & X \\ 0 & 0 & X & X & X & X \\ 0 & 0 & 0 & X & X & 0 \end{bmatrix}$$

Figure 3. Schema of a general rectangular Jacobian matrix J. Diagonal elements represent direct sensitivities of vehicle kinematic characteristics K_i to their respective design parameters DP_i, off-diagonal X entries indicate existing but undesired couplings.

In this generalized matrix representation, the diagonal elements J_{ii} capture the direct sensitivities of the system behavior (in this example, the kinematic characteristics, K_i) to their respective design parameters (DP_i), whereas the off-diagonal elements J_{ij} (for $i \neq j$) quantify the unintended couplings among the various parameters and characteristics [11].

Analysis of sensitivity results in model-based systems typically reveals distinct rectangular block structures when visualized (e.g., in a heatmap) [25]. These structures demonstrate the intrinsic subsystem architecture of the design, as variables primarily couple strongly within their respective subsystems and have relatively few interface variables connecting them to the rest of the system. By sorting the matrix entries based on non-zero elements, the Jacobian matrix can be successfully arranged to visualize these decoupled rectangular areas [10].

4. Analytical Framework for Rectangular Design Matrices

The realization that the design matrix J inherent to physical modeling is rectangular in its full generality fundamentally reshapes the analytical framework for design evaluation in model-based systems. This dimensional constraint, determined by the Buckingham Π-Theorem [8], constitutes the true upper limit of mathematical representation for physical systems [6]. Building upon this generalized structure, the analytical framework utilizes advanced sensitivity analysis techniques to systematically map the complex relationships between functional requirements (*FR*s) and design parameters (*DP*s). This approach transcends the rigid requirement of a square design matrix, offering a method to identify and manage couplings even in underdetermined systems, which are typical in complex engineering endeavors where the number of n *DP*s often exceeds the number of m *FR*s $(n \geq m)$ [6].

4.1. Coupling Analysis in Model-Based Systems Engineering

Coupling analysis is integral to model-based systems engineering (MBSE) [26], providing a systematic methodology for building and populating the system of equations that underlie complex engineering products. In this framework, the systems are modeled in an object-oriented manner, with object attributes connected through mathematical equations. While certain attributes represent design parameters (*DP*s), others correspond to functional requirements (*FR*s), which inherently introduce couplings between them. Graph-based design languages and an associated design compiler are employed to manage and generate these equations efficiently, enabling the automated creation, organization, and evaluation of the model structure [27]. Coupling analysis is performed by quantifying the sensitivity of measurable system characteristics to variations in design parameters. This is typically represented by a Jacobian matrix J, which captures the magnitude and direction of these interdependencies [10].

The procedure for generating the input data for J involves systematically varying design parameters (e.g., suspension coordinates of a race car) and analyzing their effect on system characteristics (e.g., dynamic camber and toe angle of the vehicle) [11]. The measured changes are processed using the difference quotient method to approximate the derivatives of the functions [28]. Crucially, to present the derivative data in a dimensionless form, the results are dimensionally normalized at the operating point (x_0, y_0). This normalization is advantageous because the numerical result directly correlates with the degree of nonlinearity (e.g., a value of 1 signifies a linear relationship, a value of 2 a quadratic one), if the underlying equation is a monomial [12].

The resulting normalized derivatives are assembled into the Jacobian matrix J, where each non-zero entry J_{ij} quantifies the strength of the coupling between the i-th system characteristic and the j-th design parameter [11]; see Figure 4.

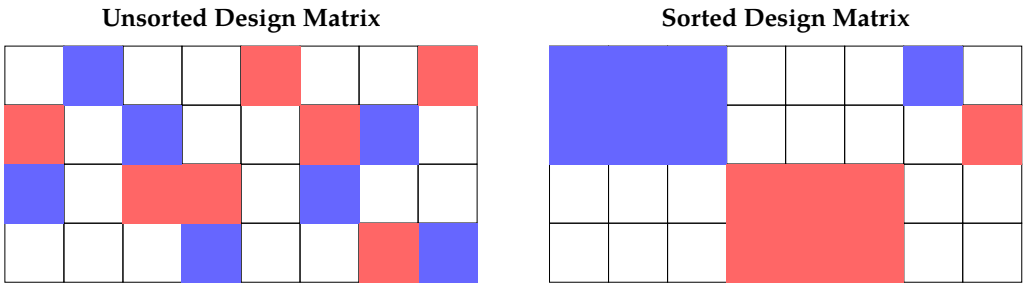


Figure 4. Heatmap of a rectangular Jacobian matrix before and after sorting. Rows represent system characteristics (K), and columns represent design parameters (DP). For the continuous sensitivity representation, red denotes positive normalized sensitivities, blue denotes negative normalized sensitivities, and white denotes no influence of the respective DP on K.

These data are typically visualized using a heatmap, which translates the numerical matrix entries into colored areas to provide a clear visual representation of the couplings. White areas indicate no influence (zero entry), while colored areas (red for proportional change, blue for inverse change) denote significant coupling [29]. A critical step in the analysis is the structural identification of the system architecture by sorting the entries of the raw Jacobian matrix, as illustrated in Figure 4.

By arranging the rows and columns according to the number of non-zero elements, following the procedure described in ([12] Section 3.3.3), parameters appearing in fewer quantities are placed towards the left, while parameters affecting many quantities are placed towards the right; correspondingly, quantities depending on fewer parameters are placed towards the lower rows, while quantities depending on many parameters are placed towards the upper rows. This sorting can, for example, be performed in MATLAB® (version R2024b)

using the *colperm* function, with the same permutation applied to the corresponding rows and columns, thereby preserving the numerical content of the matrix. The resulting heatmap reveals distinct, generally rectangular areas that are predominantly decoupled from one another. These areas are identified as contiguous regions of high non-zero density separated from the remainder of the matrix by regions of low density or zero entries. Such regions indicate groups of parameters and system characteristics that are strongly coupled with each other but only weakly coupled to the rest of the system, and are therefore interpreted as candidate subsystems. No numerical threshold is applied to define the subsystem blocks; the sorting itself is based on the presence or absence of non-zero entries, while the block boundaries are determined from the resulting density structure. The $X/0$ representation is therefore a structural visualization of non-zero versus zero sensitivities and does not introduce a user-defined numerical threshold for subsystem identification. The continuous normalized Jacobian values are retained for quantifying the magnitude and direction of the individual couplings. This organized structure reflects the intrinsic subsystem structures of the design, showing that system variables couple strongly mainly inside their respective subsystems, typically having much fewer interface variables toward the outside [11,30].

4.2. Design Decoupling and Design Freedom

In the context of the rectangular design matrix, the principles of Axiomatic Design guide the optimization strategy, focusing on achieving an efficient and systematically structured solution [11]. The Independence Axiom motivates the reduction of unnecessary couplings between functional requirements and their physical realization [4]. Within the proposed framework, this objective is operationalized through the analysis of the coupling structure between measurable system characteristics (K) and design parameters (DP), represented by the off-diagonal Jacobian elements $|J_{ij}|$. While certain couplings may be unavoidable or even desirable from a system perspective, their explicit identification supports more informed decomposition and optimization decisions. Reducing unnecessary off-diagonal couplings increases the degree of functional independence and helps reveal subsystem boundaries. The objective is not necessarily to eliminate all interactions, but rather to understand and manage them in a transparent and systematic manner. The Information Axiom provides the second directive, which is to minimize the information content by reducing the total number of influential design parameters. Minimizing the number of parameters reduces system complexity and supports a more robust solution [4]. In practice, utilizing the sorting results of the heatmap allows designers to identify the primary design drivers (variables with the greatest influence on multiple characteristics). This insight permits the systematic selection of the most influential design parameters for optimization, enabling effective system simplification [10].

The existence of the rectangular design matrix, where typically $n > m$, indicates an inherent situation of underdetermination, meaning there are more design parameters available than functional requirements (or quantifiable constraints). This often results in redundancy or coupling [19]. The analytical framework provides the necessary tools to manage this design freedom by guiding designers to focus their optimization efforts on the most critical subset of parameters, thereby minimizing the couplings required to satisfy the functional demands [11].

4.3. Implications for System Architecture Decisions

The findings derived from the coupling analysis of the rectangular design matrix inform a highly effective optimization heuristic designed to improve the system architecture. This heuristic is based on analyzing the coupling structure of the system of equations to derive a corresponding decomposition structure [11].

The identification of weakly coupled, rectangular blocks in the Jacobian matrix is essential, as it permits the decomposition of the overall optimization problem into separate subproblems. Since the couplings between these decomposed subsystems are negligible, the design and optimization objectives can be solved sequentially or independently. This decomposition strategy substantially reduces the complexity of working with large nonlinear equation systems typical of MBSE approaches [11,26].

Figure 5 illustrates how the rectangular design matrix can be decomposed into weakly coupled subsystems, highlighting the hierarchical sequence for independent or sequential optimization.

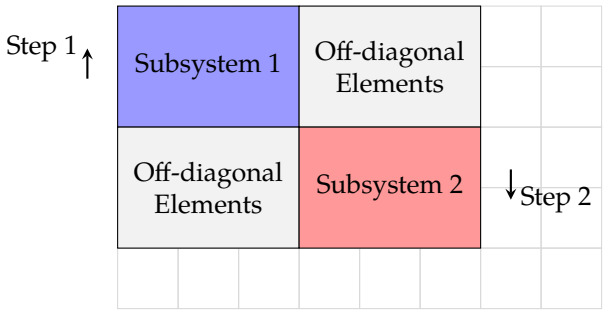


Figure 5. Hierarchical decomposition of a rectangular design matrix into weakly coupled subsystems. Arrows represent the suggested optimization sequence.

Architecturally, this modeling methodology aligns closely with the objectives of efficient object-oriented programming: prioritizing 'strong coherence and loose coupling' between modules [19]. This approach supports the separation of the engineering task into subsystems that minimize external interfaces while maintaining internal integrity. Furthermore, this analytical framework supports the inherent hierarchical nature of the system design, where complex design tasks are recursively broken down into levels of increasing detail [4]. The sequencing of the decomposition steps must follow the dependencies articulated by the effective triangular or block structure derived from the coupling analysis, ensuring that lower-level decisions are consistent with higher-level architectural choices [20]. The results from this systematic decomposition are then directly utilized in optimization routines by precisely limiting the solver's access to only the identified relevant design variables, thereby increasing computational efficiency and targeting specific design characteristics [11].

5. Case Study I: Formula Student Race Car

This section introduces a Formula Student race car suspension as a representative case study to demonstrate the application of the proposed framework for coupling analysis and system decomposition using the original idea of decoupling from Axiomatic Design in a realistic model-based engineering context.

5.1. System Decomposition and Functional Requirements

The design, development, and optimization of modern suspension systems, particularly for high-performance vehicles such as Formula Student race cars, are highly complex processes spanning multiple engineering disciplines, such as vehicle dynamics, kinematics analysis, structural optimization, collision detection, and 3D CAD design [31]. To manage this complexity, the overall process for suspension design and optimization is modeled and automated using graph-based design languages and the associated design compiler DC43® [32]. This holistic approach enables the decomposition of complex design problems into structured, automated loops.

The present case study is framed using the principles of Axiomatic Design, which provides a methodical approach to system improvement [4]. For suspension kinematics, the investigated system characteristics (K) include the dynamic changes in wheel camber, toe, and track width, roll center height, anti-dive, caster angle, and trail as well as king pin inclination and scrub radius. These characteristics represent quantifiable system responses at the physical or evaluation level. They are therefore not treated as functional requirements themselves. Instead, they serve as measurable evaluation characteristics through which the fulfillment of higher-level, solution-independent functional requirements can be assessed. Specific optimization goals often involve maximizing camber and minimizing toe angle variation during vertical suspension movement. In particular, minimizing dynamic toe angle variation, or bump steer, is crucial for ensuring predictable handling, especially on bumpy tracks [10].

5.2. Mathematical Formulation of the Suspension Kinematic Analysis

The kinematic analysis is based on the relevant suspension coordinates in the design position, as illustrated in Figure 6. For the present case study, only the vertical suspension motion is considered. The suspension is modeled as a system of rigid links, such that the relevant distances between the attachment points remain constant during the motion. The initial distances between the outer A-Arm points, the tie-rod attachment points, and the wheel center are therefore determined in this design position.

The motion is initiated by rotating the lower A-Arm about its chassis-side axis $\vec{a}$. This axis is defined by the two chassis-side points $\overrightarrow{A_{BF}}$ and $\overrightarrow{A_{BB}}$ and normalized according to

$$\vec{a} = \frac{\overrightarrow{A_{BF}} - \overrightarrow{A_{BB}}}{\left|\overrightarrow{A_{BF}} - \overrightarrow{A_{BB}}\right|}. \tag{2}$$

This requires

$$\left|\overrightarrow{A_{BF}} - \overrightarrow{A_{BB}}\right| > 0 \tag{3}$$

and the outer lower A-Arm point is subsequently expressed relative to the rotation axis as

$$\vec{P_t} = \overrightarrow{A_{BO}} - \overrightarrow{A_{BF}}. \tag{4}$$

For a prescribed rotation angle α, the corresponding rotation is described by

$$\overrightarrow{\overrightarrow{M_R}} = \begin{pmatrix} a_1^2(1-\cos\alpha)+\cos\alpha & a_1a_2(1-\cos\alpha)-a_3\sin\alpha & a_1a_3(1-\cos\alpha)+a_2\sin\alpha \\ a_2a_1(1-\cos\alpha)+a_3\sin\alpha & a_2^2(1-\cos\alpha)+\cos\alpha & a_2a_3(1-\cos\alpha)-a_1\sin\alpha \\ a_3a_1(1-\cos\alpha)-a_2\sin\alpha & a_3a_2(1-\cos\alpha)+a_1\sin\alpha & a_3^2(1-\cos\alpha)+\cos\alpha \end{pmatrix} \tag{5}$$

The transformed point is obtained from

$$\overrightarrow{P_{r,t}} = \overrightarrow{\overrightarrow{M_R}} \cdot \vec{P_t} \tag{6}$$

and translated back to the original rotation axis according to

$$\overrightarrow{A_{BO,r}} = \overrightarrow{P_{r,t}} + \overrightarrow{A_{BF}}. \tag{7}$$

The corresponding rotation of the upper A-Arm is determined such that the distance between its outer point and the transformed lower A-Arm point remains equal to the distance in the design position.

The upright-side tie rod point $\overrightarrow{T_U}$ is subsequently determined from the intersection of three spheres. Since the distances to the outer lower and upper A-Arm points and the tie-rod length remain constant, the corresponding constraints are

$$\left|\overrightarrow{T_U\,T_C}\right|^2 = (T_{U,x} - T_{C,x})^2 + (T_{U,y} - T_{C,y})^2 + (T_{U,z} - T_{C,z})^2 \tag{8}$$

$$\left|\overrightarrow{T_U A_{BO}}\right|^2 = (T_{U,x} - A_{BO,x})^2 + (T_{U,y} - A_{BO,y})^2 + (T_{U,z} - A_{BO,z})^2 \quad (9)$$

$$\left|\overrightarrow{T_U A_{TO}}\right|^2 = (T_{U,x} - A_{TO,x})^2 + (T_{U,y} - A_{TO,y})^2 + (T_{U,z} - A_{TO,z})^2 \quad (10)$$

The resulting system yields two possible intersection points. The physically valid solution is selected by comparison with the known solution in the design position.

The wheel-center coordinate $\overrightarrow{W_C}$ is determined analogously from three constant-distance constraints:

$$\left|\overrightarrow{W_C T_U}\right|^2 = (W_{C,x} - T_{U,x})^2 + (W_{C,y} - T_{U,y})^2 + (W_{C,z} - T_{U,z})^2 \quad (11)$$

$$\left|\overrightarrow{W_C A_{BO}}\right|^2 = (W_{C,x} - A_{BO,x})^2 + (W_{C,y} - A_{BO,y})^2 + (W_{C,z} - A_{BO,z})^2 \quad (12)$$

$$\left|\overrightarrow{W_C A_{TO}}\right|^2 = (W_{C,x} - A_{TO,x})^2 + (W_{C,y} - A_{TO,y})^2 + (W_{C,z} - A_{TO,z})^2 \quad (13)$$

The physically valid intersection is selected according to the known wheel-center configuration in the design position. The remaining kinematic points are determined using the same rigid-link constraints.

The complete vertical motion is generated by varying the rotation angle α of the lower A-Arm over the whole rebound to bump range. For each value of α, the constrained coordinates are calculated and stored as a corresponding suspension configuration. The resulting sequence of configurations provides the basis for calculating the kinematic suspension characteristics K, which are subsequently used for the sensitivity analysis.

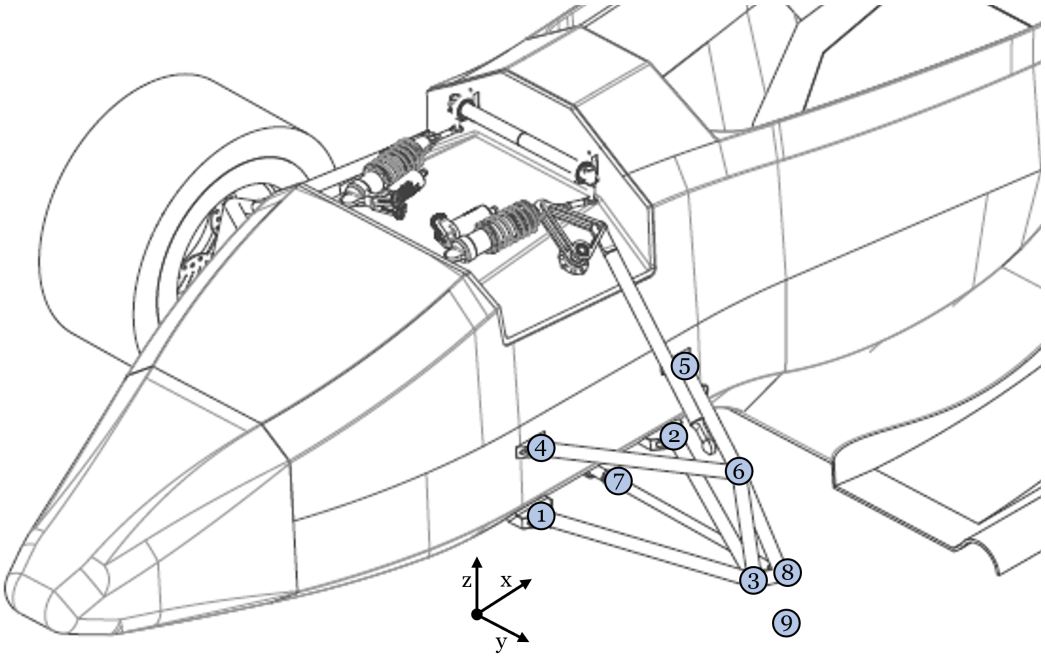


Figure 6. Suspension coordinates used for analysis: [11]. A-Arm bottom front (A_{BF}) ①, A-Arm bottom back (A_{BB}) ②, A-Arm bottom out (A_{BO}) ③, A-Arm top front (A_{TF}) ④, A-Arm top back (A_{TB}) ⑤, A-Arm top out (A_{TO}) ⑥, tie rod chassis (T_C) ⑦, tie rod upright (T_U) ⑧, wheel contact patch (W_{CP}) ⑨.

5.3. Design Matrix Construction

To understand the relationships between the design variables and the kinematic characteristics, a sensitivity analysis is performed [33]. This analysis aims to identify the design drivers that have the strongest influence on the static and dynamic characteristics of the suspension, thereby improving the kinematics optimization process [11].

The design parameters (*DP*s) that are subject to variation include the coordinates (x, y, z) of critical suspension points (e.g., A-Arm attachment points or tie rod points), which amount to 24 parameters. Along with the track width, these parameters result in 25 variables for evaluation, as illustrated in Figure 6.

The effect of varying these *DP*s on the kinematic characteristics (K) is represented using their derivatives, which are determined using the difference quotient method [28]:

$$\frac{\Delta y}{\Delta x} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{14}$$

Accordingly, the rows of the Jacobian matrix represent system characteristics, and the resulting matrix quantifies their local sensitivity to solution-specific design parameters. To ensure comparability of the values of the derivatives and to reveal their nonlinear relationships, the derivatives are dimensionally normalized using the operating point (x_0, y_0) [12]:

$$\left\| \frac{\partial y_i}{\partial x_j} \right\| = \frac{\partial y_i}{\partial x_j} \left| \frac{x_{j0}}{y_{i0}} \right| \tag{15}$$

This normalization makes the sensitivities dimensionless but does not account for the engineering importance, allowable tolerances, or priorities of individual characteristics. The normalized derivatives are then organized into a normalized Jacobian matrix:

$$\left\| J_y(x) \right\| = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} \left| \frac{x_{10}}{y_{10}} \right| & \frac{\partial y_1}{\partial x_2} \left| \frac{x_{20}}{y_{10}} \right| & \cdots & \frac{\partial y_1}{\partial x_n} \left| \frac{x_{n0}}{y_{10}} \right| \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} \left| \frac{x_{10}}{y_{m0}} \right| & \frac{\partial y_m}{\partial x_2} \left| \frac{x_{20}}{y_{m0}} \right| & \cdots & \frac{\partial y_m}{\partial x_n} \left| \frac{x_{n0}}{y_{m0}} \right| \end{pmatrix} \tag{16}$$

Each element of the normalized Jacobian matrix represents the local sensitivity of a system characteristic to a design parameter. The resulting matrix therefore quantifies the influence of individual design parameters on the kinematic characteristics. For the analysis of coupling structures, the distribution of sensitivity values across the matrix is examined to identify groups of strongly or weakly coupled characteristics and design parameters [10]. To visualize these relationships, the Jacobian matrix is presented as a heatmap, as shown later in Figure 7.

5.4. Design Optimization and Decoupling Strategies

According to the Independence Axiom, the objective is to minimize the couplings between the design parameters and the kinematic characteristics [4,11]. Minimizing these couplings is achieved by reducing the off-diagonal elements of the Jacobian matrix to the lowest possible values. Therefore, the generated data can be strategically sorted based on the number of non-zero elements [10]. This sorting process groups the most influential variables in the upper part of the heatmap, allowing primary design drivers to be easily identified. Furthermore, this rearrangement reveals rectangular areas that are largely decoupled from each other. Figure 7 shows a comparison of the resulting heatmaps prior sorting (left) and after sorting (right). The decoupled rectangular areas are marked accordingly. For the Formula Student race car suspension, the analysis revealed that the position of the A-Arms in the z-direction affects the largest number of kinematic characteristics and is thus of crucial importance [11].

The systematic block structure observed in the sorted heatmap reflects the intrinsic subsystem structures, where subsystem variables primarily couple internally and have fewer interface variables externally [4]. These topological insights form the basis for deriving a decomposition structure that allows system objectives to be optimized separately due to minimal or non-existent couplings after decomposition [11].

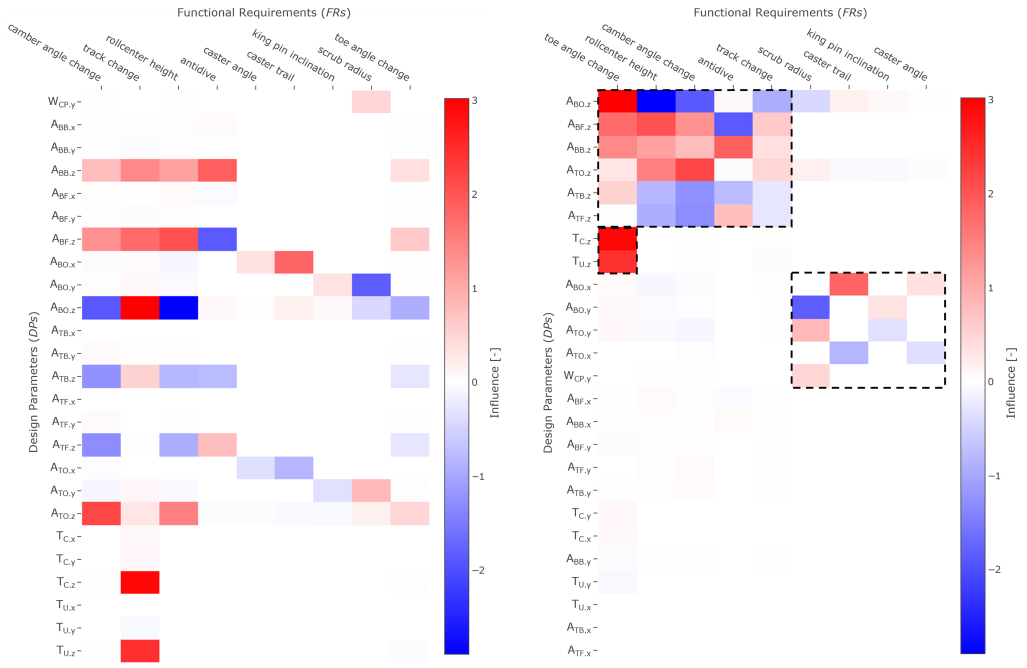


Figure 7. Heatmap prior sorting (**left**) and post sorting (**right**). Reproduced from [11] and used here to illustrate the coupling-structure interpretation within the proposed framework. Red denotes positive, blue negative, and white no influence of each design parameter on its respective system characteristic.

Figures 7 and 8 visualize representative heatmaps in which the decoupled blocks can be identified and used to optimize the corresponding subsystems separately.

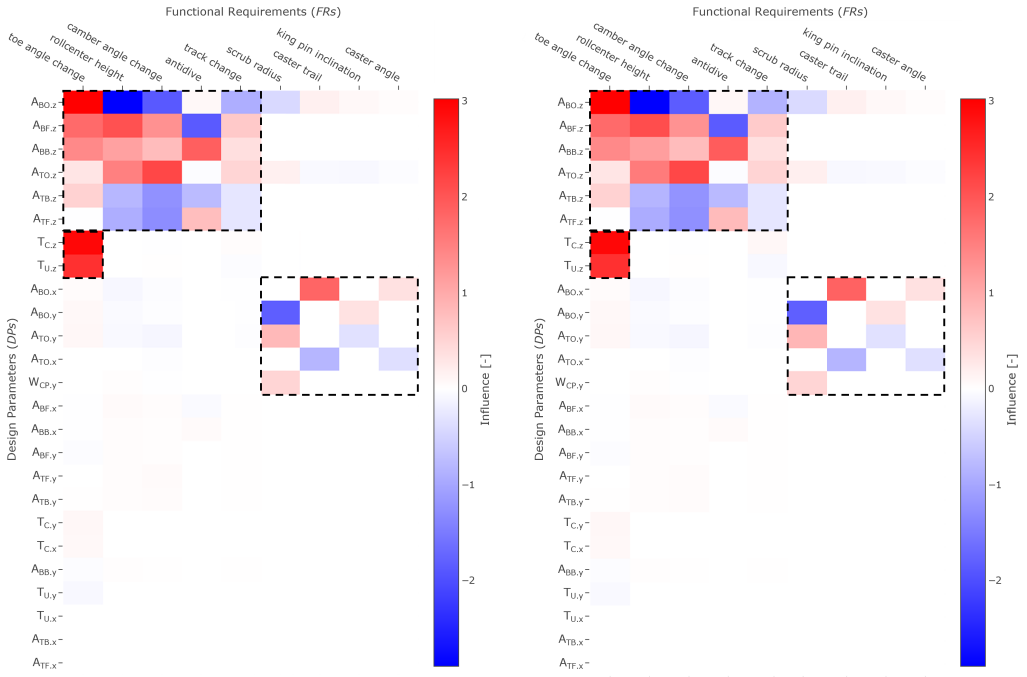


Figure 8. Comparison of heatmaps at the operating points before optimization (**left**) and after optimization (**right**). Reproduced from [11] and used here to illustrate the consistency of the decomposition results and design parameter sensitivities. Red denotes positive, blue negative, and white no influence of each design parameter on its respective system characteristic.

The objective is to leverage the specific structure of these heatmaps, which is characteristic of model-based approaches in which individual subsystems are integrated into an

overall system. From a global perspective, optimizing the suspension system is a nonlinear problem. Therefore, the applied decomposition techniques are only valid in the vicinity of a given operating point [10]. Due to the relatively small variations around this operating point, the optimization problem can be solved sequentially. After completing the optimization procedure, the governing equations are solved jointly within the context of the complete system [11]. Figure 8 illustrates heatmaps for two operating points (x_0, y_0).

The heatmap on the left shows the operating point before optimization, and the heatmap on the right shows the operating point after optimization. Comparing the two heatmaps reveals consistency in both the decomposition results and the orders of magnitude of the design parameter influences. This consistency supports the validity of the sequential solution strategy for the given optimization problem at the operating point [10]. Building on this consistency in the decomposition insights derived from the sensitivity analysis, the design optimization process becomes much more targeted and efficient [11].

6. Case Study II: Small Satellite

The FireSAT mission is an earth observation satellite mission detailed within the well-established book *Space Mission Analysis and Design* [34] (often called SMAD-book for short), covering the entire design of a satellite mission in one single document. The SMAD-book serves as reference textbook in many systems engineering courses worldwide and the encoding of a very large portion of the book's content in a design language makes the satellite design machine-analyzable as will be reported in the following.

The main payload of the FireSAT is an infrared camera for forest fire detection. As the subsystems of a satellite are highly interconnected, changing one singular parameter of a subsystem will have knock-on effects on the other subsystems. This means that even minor optimizations in one of the subsystems, for example the resolution of the satellite's payload, will potentially impact the design of other subsystems quite substantially. In the following sections, these interactions are visualized with several heatmaps.

6.1. FireSAT Design Language

In analogy to the previous Formula Student case study, the FireSAT mission is automatically designed by executing a design language within the systems engineering development environment DC43® [32]. Much of the design language is based on the original works already presented in [13,14]. The full design language encompasses the design of the payload, orbit, structure, power system, communication telemetry, and command system as well as the attitude and orbit control system. Each of these subsystems comprises multiple components performing different functions. Executing the design language includes the design or selection of specific components to fulfill subsystem requirements. The results of the design language presented in this work were generated using an updated and modified version of the original design language, mostly due to an update of the underlying systems engineering software framework DC43®, from version two to version four. The current FireSAT design language comprises a total of 206 class definitions, 402 attributes and 760 design constraints in the form of symbolic equations.

The design of the satellite is separated into several consecutive steps. First, the payload is designed based on the chosen orbit altitude as well as the desired ground sample distance. This results in the dimensions of the payload, its mass, generated data and power demand. Each of these parameters is a main driver in the subsystems of structure, attitude and orbit control, data handling, and power. These subsystems are interconnected: the structure provides mounting areas for solar cells, while components such as data memories or reaction wheels require both power and available space.

As no subsystem can be designed independently of the other, an iteration is required until the demands of all subsystems are met. This preliminary iteration contains only topological modifications, for example the selection of specific components, such as sensor types or the number of solar panels. This means that all parameters not set during model creation are automatically derived via their constraints, expressed as symbolic equations. These symbolic equations are the prerequisites for creating the heatmaps later.

The second design loop contains the packaging logic. This packaging decouples some of the design parameters, as the size of the satellite structure now depends on distances provided by an external CAD-kernel, no longer available for analysis within the scope of the heatmaps. Resizing the satellite therefore requires a redesign of the other subsystems, as the required torque for pointing the satellite or the available area for mounting solar cells may change. Consequently, the heatmaps shown later are derived from an intermediary design step, where most of the satellite geometry is not yet fully realized.

The figures in the following section contain a parameter naming convention based on this design language. As the design language is structured in classes containing attributes, a parameter is named *class name {graph id}.attribute name*. For example, the length of the satellite is defined in the class *MechanicalBudget*, so the parameter *MechanicalBudget{#13}.length* describes the length of the satellite, and #13 means that the instance of the class *MechanicalBudget* has the node id 13 in the graph. This allows the differentiation of multiple component instances from the same class, for example two antennas.

6.2. Selection of Optimization Parameters

The number of parameters detailing the entire satellite pre-design is in the several hundreds. Consequently, this requires a careful selection and filtering process of design parameters to find suitable candidates for design optimization later. In this example, a total of 56 input parameters define the design of the satellite, resulting in 422 design parameters. These parameters have been filtered to a reduced view of 19 input parameters and 70 resulting design parameters. The filtering criteria were defined as parameters that cannot be modified, parameters with only a limited number of non-zero couplings to other design parameters, and intermediary parameters whose influence on system-level performance is represented through other parameters. The heatmap can be seen in Figure 9.

The input parameters comprise constants of nature, fixed design requirements and parameters describing selected components of the shelf (COTS). Only the first 19 of these parameters are shown. The omitted parameters are parameters of COTS which cannot be changed and impact only a few other parameters. In most cases, these were the masses and dimensions of individual components.

The design parameters are obtained by solving the mathematical constraints defined in the symbolic equations of the individual class definitions. These describe different system and component attributes, such as a required torque for actuator selection. In the case of the FireSAT design language, the design parameters may also contain requirements which depend on other design decisions. This necessitates the implementation of an iterative design variation within the design logic of the design language, which is re-executed until all requirements are met. One such example is the detection accuracy of the optical instrument. The detection accuracy depends heavily on the sensors selected during the design process. This can be seen in Figure 9 in the coupling of *OpticalInstrument{#11}.detectionAccuracy* and *HG1802{#71}.angularAccuracy*. HG1802 is the name of a COTS horizon sensor with a given angular accuracy as defined by the data sheet. The input parameter of its angular accuracy only impacts the detection accuracy of the satellite.

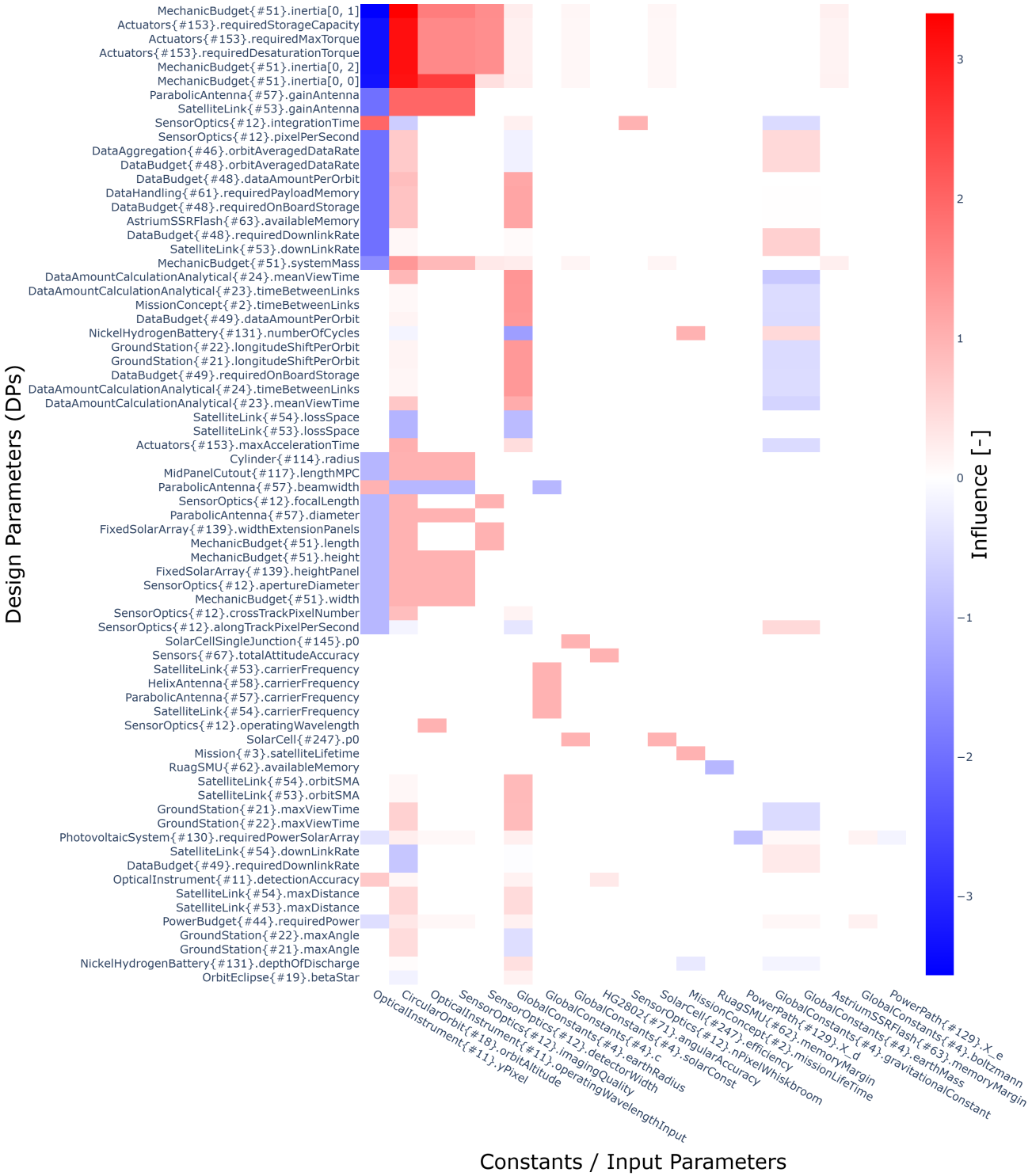


Figure 9. Sorted heatmap of a FireSAT satellite with parameters covering a selection of requirements, constants of nature and properties from selected components on the *x*-axis and resulting design parameters on the *y*-axis. Parameter naming convention explained in text. The color scale represents the sign and magnitude of the normalized sensitivity, with red denoting positive sensitivities and blue denoting negative sensitivities.

Some input parameters with high impact on the design are unmodifiable, such as natural constants (e.g., Boltzmann constant, or speed of light), or cannot be feasibly varied, e.g., the radius of the Earth. Other design parameters are only intermediary parameters, for example the carrier frequencies of the antennas define the mass of the antennas which in turn defines the overall system mass. Individual component masses also impact the

inertia which in turn results in a required torque for the actuators. While relevant for the individual subsystem design, these parameters are excluded from the optimization of the overall satellite design because their influence is represented through other parameters. After filtering, five input parameters remain that enable the optimization of key performance indicators, such as the dimensions and the mass of the satellite, the available power from the solar array or the torque required from the reaction wheels. The normalized heatmap in its filtered form is shown in Figure 10.

Due to the high interdependence of the satellite subsystems, optimizing individual aspects is difficult at the system level. The approach reveals certain parameters that can be optimized for, without impacting other system aspects. As an example, the length of the satellite bus can be varied without impacting its width and height. The detector width of the optical system impacts only the length attribute, while width and height depend on imaging quality and chosen operating wavelength.

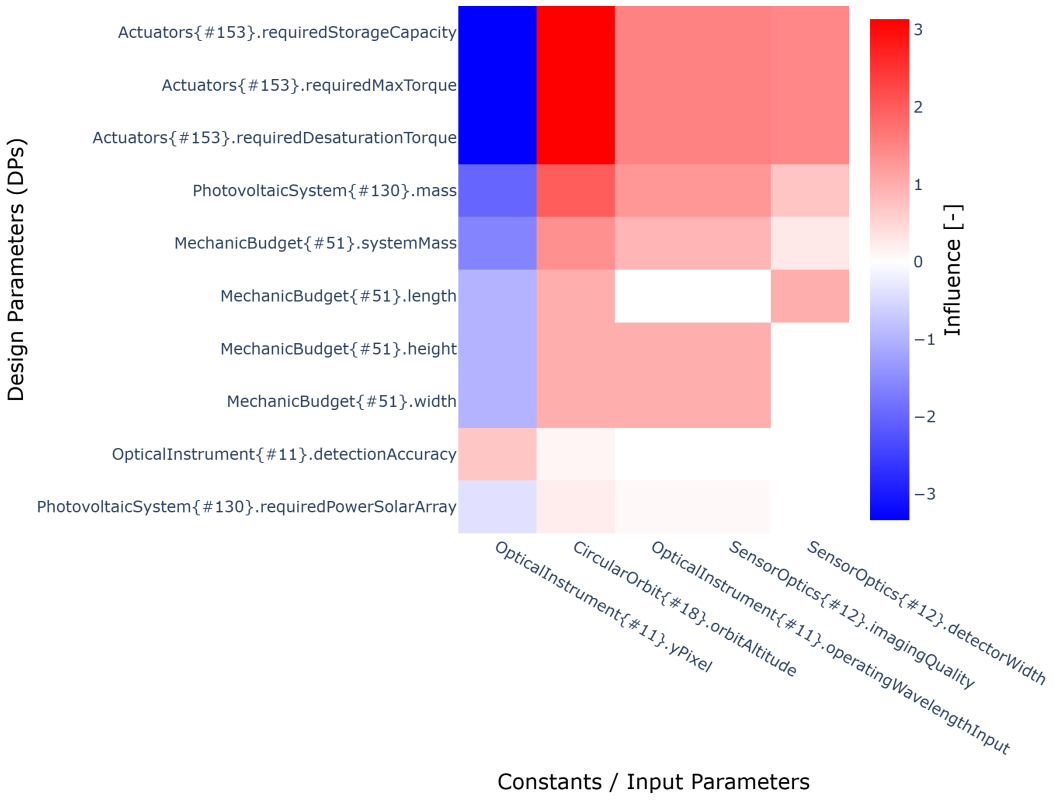


Figure 10. Filtered heatmap of the FireSAT satellite with key performance indicators on the *y*-axis and design variables on the *x*-axis. The color scale represents the sign and magnitude of the normalized sensitivity, with red denoting positive sensitivities and blue denoting negative sensitivities.

Finally, it should be noted that the machine execution of any design language generates the underlying network of design constraints in the form of hundreds of symbolic equations. These equations are automatically solved, and the normalized Jacobian is computed from their solutions. Sorting and filtering are then individual, customizable analysis steps that depend on the user's intent when searching for and inspecting 'interesting' design couplings. Of course, these filtering steps are context-dependent and allow an individualized design analysis based on the system designer's intent.

In consequence and as a conclusion from the benefits observed in the two case studies, such a system-wide coupling analysis can provide a valuable capability in software platforms intended for complex system design.

7. Discussion

The presented results show that coupling analysis based on rectangular design matrices constitutes a natural and practically relevant extension of classical Axiomatic Design for model-based systems engineering. Sensitivity-derived Jacobian matrices reflect the structural realities of physical modeling and dimensional homogeneity, making rectangular matrices the general case rather than the exception. Based on these findings, the discussion is structured into two parts: the use case discussion in Section 7.1 and the method discussion in Section 7.2.

7.1. Use Case Discussion

The normalized heatmap visualization proves effective for exposing intrinsic subsystem structures. In both case studies, strong intra-subsystem couplings and weak inter-subsystem interactions emerge as clearly identifiable rectangular blocks. This provides an objective, model-based foundation for system decomposition and enables an operational interpretation of the Independence Axiom without enforcing artificial square-matrix constraints. A supplementary example illustrating the automated equation solving and heatmap generation in a graph-based design language is provided in Appendix A. It should be noted that the objective of the framework is not to eliminate all couplings within a system. Complex engineering products inherently contain necessary interactions between subsystems. The proposed analysis therefore aims to identify and manage coupling structures, while highlighting opportunities for decomposition where weakly coupled subsystem boundaries naturally emerge.

In this work, these previously developed engineering models and sensitivity results are not treated as new experimental findings; they rather serve as established application instances through which the proposed integrated interpretation can be examined. The Formula Student use case demonstrates that locally valid decomposition enables sequential optimization around a defined operating point, with stable coupling structures before and after optimization. The FireSAT satellite use case highlights the framework's usefulness for early-stage architectural reasoning and decision making, where multiple architectural options with large parameter sets each must be filtered and structured prior to optimization.

Despite all the mentioned advantages in both use cases, several limitations must be acknowledged. The decomposition is local, based on linearized sensitivities around an operating point, so strong nonlinearities, large design changes, or discrete topological architecture decisions may alter coupling structures. The present study does not quantify the effective neighborhood of validity or evaluate the condition number of the Jacobian. Likewise, no dedicated test under strong nonlinearities or large design variations is included. A check before and after optimization is therefore necessary to assess whether the identified coupling structure remains sufficiently consistent for the intended optimization sequence. In addition, the present study does not include an uncertainty, tolerance, or robustness analysis; accordingly, robustness in the statistical or tolerance-based sense is not demonstrated by the presented results. Furthermore, subsystem identification via matrix sorting is, at least in its current form, still heuristic and therefore not yet formally unique, as discussed in [12] which provides further sorting options and details. Despite this open research question, the framework provides nevertheless a transparent, reproducible method for guiding design decisions where the concept of Axiomatic Design may offer only limited guidance.

Beyond engineering design, the proposed decomposition principle may also be applicable to other domains characterized by structured dependency networks. Investigating such cross-domain applications constitutes an interesting direction for future research.

7.2. Method Discussion

The normalized heatmap visualization was derived from the system equations in dimensional form (i.e., from equations of physical quantities with physical units, such as a physical quantity like a mass m measured in $[kg]$, or a velocity v measured in $[m/s]$). Suppose that a system variable equation y has the following form of a monomial equation, i.e., (with $n \in N, |n| \geq 1$) [12]

$$y = c\,x^n \tag{17}$$

and for which the following point probe (x_0, y_0) with $y_0 = c\,x_0^n$ is valid. From Equation (17), the following partial derivative $\partial y/\partial x$ can be derived with [12]

$$\frac{\partial y}{\partial x} = n\,c\,x^{n-1} \tag{18}$$

and the normalization using the point solution (x_0, y_0) from Equation (18) finally yields [12]

$$\left|\frac{x_0}{y_0}\right| \cdot \left.\frac{\partial y}{\partial x}\right|_{x=x_0} = \left|\frac{x_0}{y_0}\right| \cdot n\,c\,{x_0}^{n-1} = n \cdot \frac{c\,{x_0}^{n-1}}{|c\,{x_0}^{n-1}|} = \quad n \cdot sign(c\,{x_0}^{n-1}). \tag{19}$$

It is therefore no surprise that the numerical values observed in most normalized heatmaps remain between -3 and $+3$, since in many cases the physical coupling of most design variables is of linear, quadratic or cubic nature or their inverse. Of course, this is deduced from the special monomial form of Equation (17) and does not hold for other kinds of relationships except for cases in which the local solutions and derivatives of these other kinds of relationships at the point (x_0, y_0) are approximated by a monomial in the form of Equation (17), for which the shown derivation holds.

Additional methodological remark: Besides this normalized dimensionless representation of the derivatives in the form of the heatmaps, an investigation of further dimensionless methods in system representation and analysis was undertaken also in other contexts (e.g., in the project ALARMBOT, in which a digital twin for a motor engine was investigated). The mathematical distinction between a simple normalization (i.e., a fraction of two or several variables with cancelling physical units, often used in numerical solution schemes where a value u is normalized with a reference value u_0 in form of the dimensionless fraction u/u_0) and a true dimensionless invariant (as well a fraction of several variables with cancelling physical units, however with an additional constraint on the variables used) is the way in which the dimensionless invariant is built. Dimensionless invariants are always built as monomials of a set of physical variables $(x_1, \ldots, x_n)$, which are simultaneously coupled in a complete and dimensionally homogeneous physical relationship $f(x_1, \ldots, x_n) = 0$. If this condition is fulfilled, the dimensionless monomials qualify as dimensionless invariants.

The existence of these dimensionless invariants is guaranteed by the Π-Theorem of Buckingham [8] and these dimensionless invariants have interesting properties, which are closely related to the principle of physical similarity [27]. Despite our intensive research efforts within this and other research projects so far, our idea to exploit the additional existence of a dimensionless digital twin as the dimensionless counterpart to the dimensional system equations underlying the normalized sensitivity analysis appeared to be a conceptual idea we have so far not been able to exploit despite its potential which engaged us at the time in this research direction.

From a methodological viewpoint the results demonstrate that coupling structures derived directly from model-based sensitivity information provide a consistent, physically grounded basis for architectural reasoning and systematic system-level optimization, reconnecting the *a posteriori* paradigm of 'Axiomatic Design' with the *a priori* software engineering paradigm of 'Strong Coherence and Loose Coupling' commonly used in object-oriented programming and therefore implicitly underlying any digital modeling effort in object-oriented graph-based design languages.

8. Conclusions

This study presents an analytical framework for a systematic coupling analysis based on rectangular design matrices. By combining the sensitivity analysis with a normalized heatmap visualization, the approach enables an easy and straightforward identification and interpretation of coupling structures without imposing square-matrix assumptions inherent to the original concept of Axiomatic Design.

The central contribution is a generalized interpretation of the original idea of the Independence Axiom through the identification of weakly coupled rectangular subsystem blocks. This perspective supports systematic system decomposition, targeted parameter selection, and sequential or independent subsystem optimization. The presented case studies confirm the practical relevance of the framework for complex engineering systems in real-world applications.

Our future works will continue to investigate an automated subsystem identification, the systematic integration with MBSE workflows, and also further applications beyond typical engineering design domains.

Author Contributions: Conceptualization, J.B. and S.R.; methodology, J.B. and S.R.; validation, J.B. and F.L.; investigation, J.B. and F.L.; writing—original draft preparation, J.B., F.L. and S.R.; writing—review and editing, J.B., F.L. and S.R.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Federal Ministry for Economic Affairs and Energy as part of the project 'ALARMBOT—Assembly Line Abnormality Recognition Method Based On Twin' under Grant No. 03EE5143C (University of Stuttgart). The authors are responsible for the content of this publication.

Supported by:



on the basis of a decision
by the German Bundestag

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to julian.borowski@iils.de.

Acknowledgments: The authors gratefully acknowledge the financial support provided by the Foundation for Innovation in Higher Education (Stiftung Innovation in der Hochschullehre), as part of the project 'Lehrarchitektur' at RWU Ravensburg-Weingarten University of Applied Sciences, for covering the article processing charges (APC) associated with this publication. Furthermore, the authors wish to thank the cooperating Formula Student Team of the Ravensburg-Weingarten University for making use of graph-based design languages and the corresponding compiler DC43® in their development process as well as sharing their design data and development experience with us.

Gefördert durch die



Stiftung
Innovation in der
Hochschullehre

Conflicts of Interest: Julian Borowski and Felix Löser are employed by IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH. The authors declare no conflicts of interest related to the research presented in this manuscript.

Appendix A

This section is intended to present a self-contained example of the equation solving and the heatmap generation from a graph-based design language in a nutshell. It starts with the thermodynamic model of a gas turbine as an illustrative example with 11 symbolic equations and 19 variables. It is taken from [27], where it is described in further detail. In order to match the number of unknowns with an equal number of equations, it is clear that 8 additional boundary conditions are needed to make an equation system with 11 equations and 19 variables solvable.

The thermodynamic cycle of a gas turbine represents a rectangle in the (T, s)-diagram with the four corners of the thermodynamic states $1, 2, 3$ and 4. If the two thermodynamic quantities p and T are known, the complete thermodynamic state can be computed for a known medium with known gas constant R from the ideal gas equation $p = \rho RT$. The thermodynamic cycle process starts with the environmental conditions (state 1 with p_1, T_1), followed by an isentropic compression in the compressor (into state 2 with p_2, T_2), which is followed by an isobaric combustion in the combustion chamber (into state 3 with p_3, T_3). Then an isentropic expansion in the turbine (into state 4 with p_4, T_4) is followed by an isobaric cooling of the exhaust gas back to environmental conditions (back to state 1 with p_1, T_1) to close the thermodynamic cycle. Nevertheless, in order to understand the internal information processing within the design language, no special knowledge of the physics of a gas turbine is required, since for the understanding of the information processing it is sufficient to consider the nonlinear equations just as given.

The creation of a design language is a digital modeling effort of the involved physics. From a physics point of view, the class diagram in Figure A1a needs to contain the 11 equations to model the physics and the definitions of the 19 physical variables as 19 attributes. Since 8 variables are already initialized with values, the 11 equations allow for the remaining 11 unknowns to be solved. It is important to note that the 11 equations are defined in their implicit form, without any predetermined indication which variables in the equations will be later solved for. This is done automatically via an inbuilt constraint processing technique known as solution path generator. The technology of the solution path generator allows the declarative definition of the physics, which guarantees a maximum in flexibility in later *what*... *if*... scenarios in respect to conventional procedural programming of the physics, which would require significant reprogramming. The result of the automated equation solving is written back into the design graph shown in Figure A1c, which is generated during the execution of the activity diagram in Figure A1b during the compilation process of the design language.

Figure A2 shows the corresponding solution sequence of the equations. This sequence starts of course with the given 8 boundary conditions shown in grey color and follows then a dedicated solution sequence determined at run-time based on a dependency analysis of the equations. This is realized by the aforementioned solution path generator.

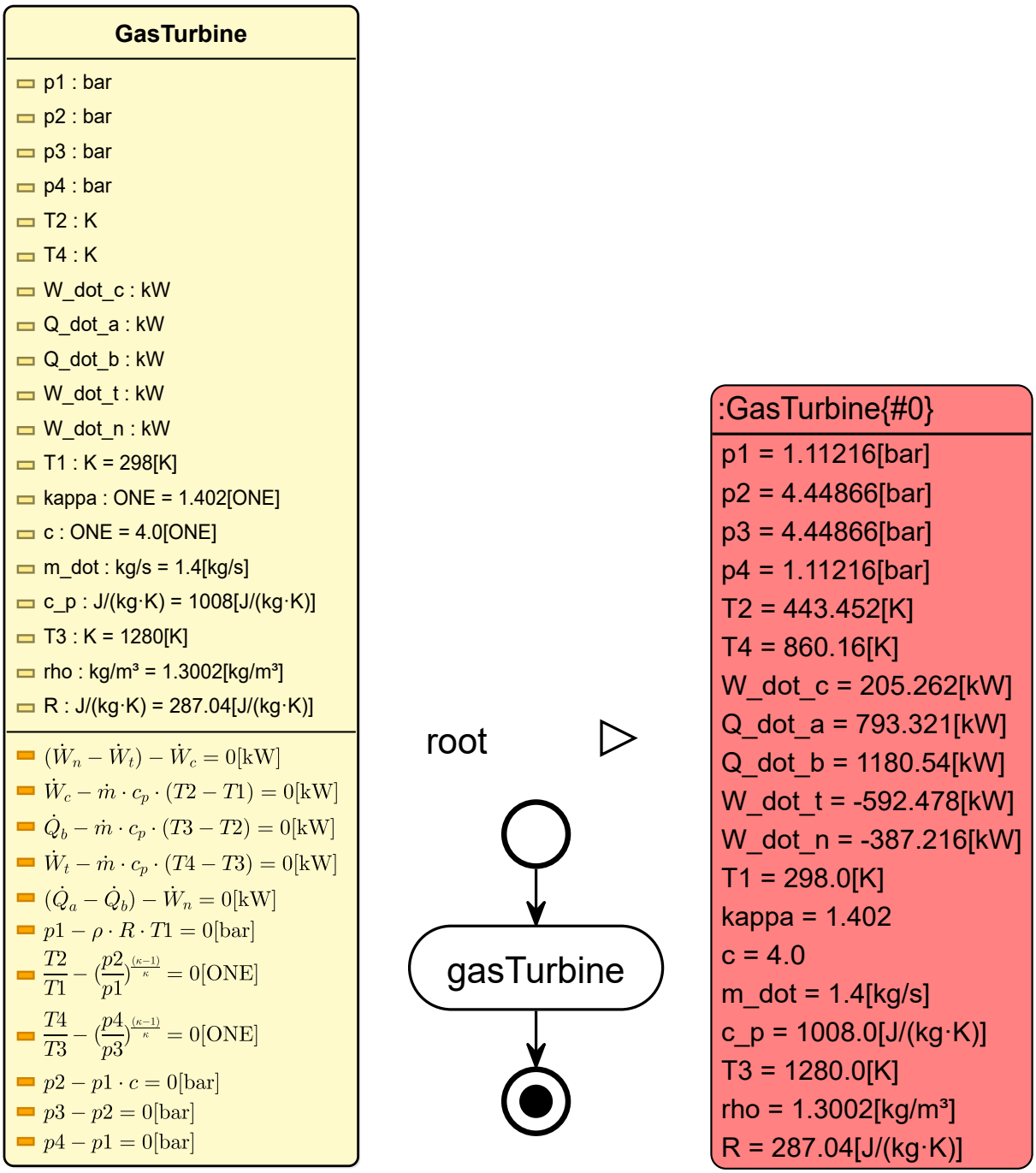


Figure A1. Gas turbine example: class diagram, activity diagram and design graph. (**a**) Class diagram (showing 19 attributes and 11 equations). (**b**) Activity diagram. (**c**) Design graph.

Finally Figure A3 shows the resulting normalized heatmaps in both unsorted and sorted form. The coloring with red and blue allows a quick overview about the positive or negative influence of changes in the 8 boundary conditions on the 11 other design parameters. As a concrete reading example in Figure A3, the power delivered by the gas turbine $\dot{W}_n$ (alias *GasTurbine{#0}.W_dot_n*) grows linearly with the mass flow $\dot{m}$ (alias *GasTurbine{#0}.m_dot*) and similarly do $\dot{Q}_a$, $\dot{Q}_b$, $\dot{W}_c$ and $\dot{W}_t$.

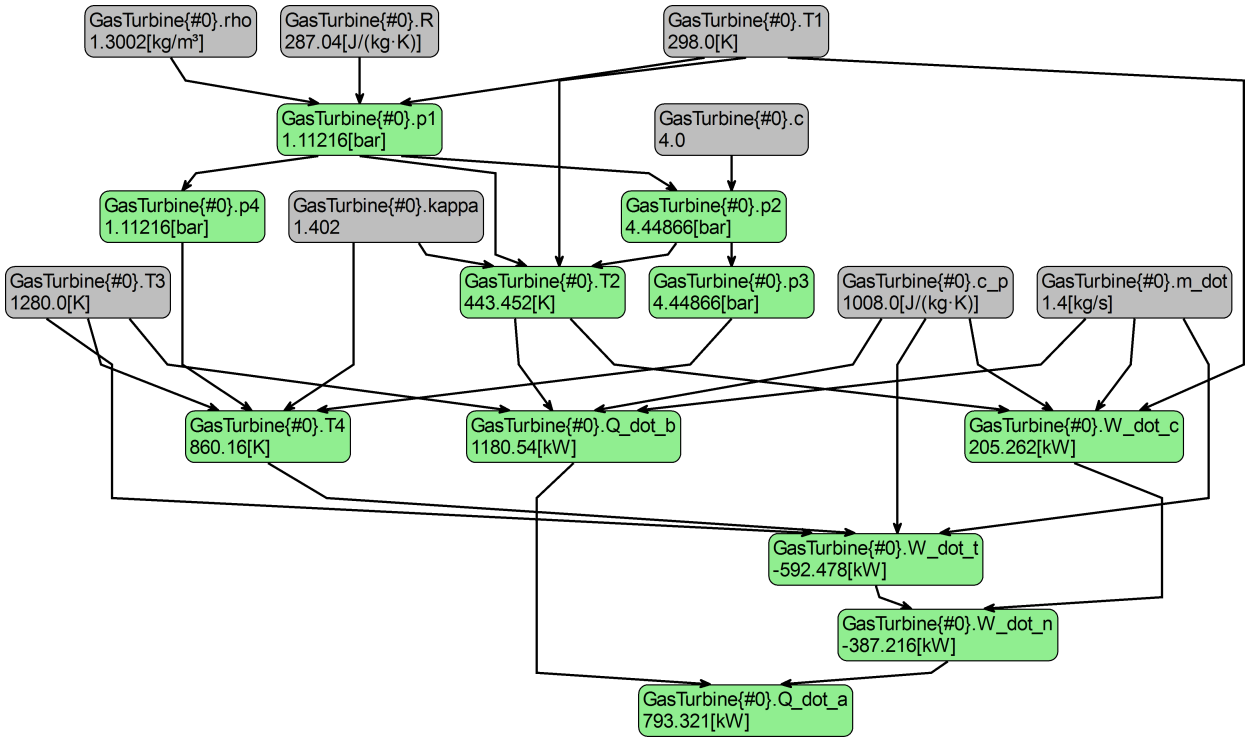


Figure A2. Gas turbine example: solution path.

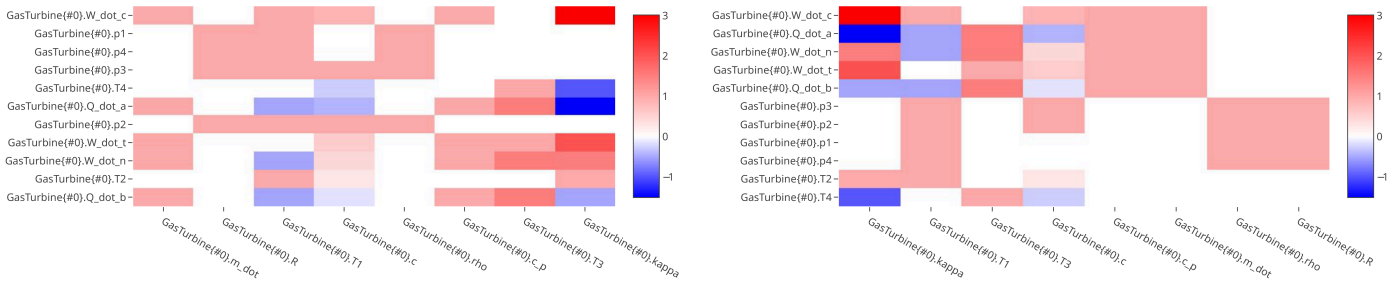


Figure A3. Gas turbine example: heatmaps prior to sorting (**left**) and post sorting (**right**).

Final remark on modeling. In the example in Figure A1a, the digital modeling has been simplified as much as possible; i.e., all equations and boundary conditions have been put in one single class *GasTurbine* for their later instantiation, equation solving and heatmap generation. This has been done to avoid additional complexity to instantiate several classes with potentially several rules, thus increasing the modeling complexity beyond the level necessary to understand the equation processing.

In a typical modeling effort, the equations would be partitioned into different classes; i.e., the isentropic compression equation would be in a class called *Compressor*, the isobaric combustion equations would go into a class called *CombustionChamber* and the isentropic expansion equations into a class called *Turbine* and so forth. Such a dedicated splitting of the equations during modeling would allow much more flexibility in later instantiations; e.g., it would even allow the modeling of multi-stage gas turbines instead of only one single hard-coded single-stage turbine shown here. However, then the activity diagram contains more rules to instantiate the classes, diverting the focus from understanding the automated equation solving and heatmap generation and shifting it towards the many different alternatives in digital modeling.

References

1. Vizitiu, C.; Dinculescu, A.C.; Marin, M.G.; Nistorescu, A.I.; Dominey, K.A.; Kristály, D.M. Optimizing early-stage efficiency in complex system development via model-based systems engineering (MBSE) and concurrent engineering (CE) integration. *Res. Eng. Des.* **2025**, *36*, 12. [CrossRef]
2. Harbin, J.; Gerasimou, S.; Matragkas, N.; Zolotas, T.; Calinescu, R.; Alpizar Santana, M. Model-driven design space exploration for multi-robot systems in simulation. *Softw. Syst. Model.* **2023**, *22*, 1665–1688. [CrossRef]
3. Bjarklev, K.; Eifler, T.; Mortensen, N.H.; Linnebjerg, S.; Ebro, M. Product behavior complexity metric for early prioritization of tolerance analysis tasks. *Des. Sci.* **2020**, *6*, e1. [CrossRef]
4. Suh, N. *The Principles of Design*; Oxford Press: New York, NY, USA, 1990; Volume 1.
5. Pecoraro, F.; Pourabbas, E.; Rolli, F.; Parretti, C. Digitally Sustainable Information Systems in Axiomatic Design. *Sustainability* **2022**, *14*, 2598. [CrossRef]
6. Rudolph, S. Upper and lower limits for 'the principles of design'. *Res. Eng. Des.* **1996**, *8*, 207–216. [CrossRef]
7. Choi, K.K.; Kim, N.H. *Structural Sensitivity Analysis and Optimization 1*, 1st ed.; Springer: New York, NY, USA, 2005.
8. Buckingham, E. On Physically Similar Systems: Illustration of the Use of Dimensional Equations. *Phys. Rev.* **1914**, *4*, 345–376. [CrossRef]
9. Su, J.C.-Y.; Chen, S.-J.; Lin, L. A structured approach to measuring functional dependency and sequencing of coupled tasks in engineering design. *Comput. Ind. Eng.* **2003**, *45*, 195–214. [CrossRef]
10. Borowski, J.; Rudolph, S. Automation and Optimization of the Design and Development Process for a Formula Student Racing Car Suspension. In *2025 Stuttgart International Symposium*; SAE Technical Paper 2025-01-0270; Society of Automotive Engineers (SAE): Warrendale, PA, USA, 2025.
11. Borowski, J. Digitaler Entwurf eines Formula Student Fahrzeugs am Beispiel von Fahrwerk und Antriebsstrang zur 3D-Geometrie- und Kinematikerstellung. Ph.D. Thesis, Department of Aerospace Engineering, University of Stuttgart, Stuttgart, Germany, 2026. [CrossRef]
12. Bölling, M. Lösungspfadbasierte Analysen im Entwurf Komplexer Systeme. Ph.D. Thesis, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen der Universität Stuttgart, Stuttgart, Germany, 2015. [CrossRef]
13. Groß, J. Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten. Ph.D. Thesis, Institute of Aircraft Design, University of Stuttgart, Stuttgart, Germany, 2014. [CrossRef]
14. Loeser, F.; Rudolph, S. Facilitating Topological Design Decisions in early mission design phases with automated Knowledge Graph generation by means of Graph-Based Design Languages. In *AIAA SciTech Conference, Orlando, FL, USA, 12–16 January 2026*; AIAA: Reston, VA, USA, 2026. [CrossRef]
15. Saltelli, A.; Ratto, M.; Andres, T.; Campolongo, F.; Cariboni, J.; Gatelli, D.; Saisana, M.; Tarantola, S. *Global Sensitivity Analysis: The Primer*; John Wiley and Sons, Ltd.: Hoboken, NJ, USA, 2008. [CrossRef]
16. Khalil, H.K. *Nonlinear Systems*; Prentice Hall: Upper Saddle River, NJ, USA, 2002; Volume 3.
17. Steward, D.V. The design structure system: A method for managing the design of complex systems. *IEEE Trans. Eng. Manag.* **1981**, *28*, 71–74. [CrossRef]
18. Kim, S.-J.; Suh, N.; Kim, S.-G. Design of Software Systems Based On Axiomatic Design. *Robot. Comput.-Integr. Manuf.* **1991**, *8*, 243–255. [CrossRef]
19. Suh, N. Axiomatic Design Theory for Systems. *Res. Eng. Des.* **1998**, *10*, 189–209. [CrossRef]
20. Tate, D. A Roadmap for Decomposition: Activities, Theories, and Tools for System Design. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999. Available online: http://hdl.handle.net/1721.1/9429 (accessed on 12 December 2025).
21. Zhang, X.; Zhang, Q.; Zhao, Q.; Zhao, S.; Zhao, Y.; Guo, Y.; Zhao, Z.; Song, L. An axiomatic system engineering design method based on NSGA-II algorithm applied to complex systems. *Sci. Rep.* **2025**, *15*, 31766. Correction in *Sci. Rep.* **2025**, *15*, 43941. [CrossRef] [PubMed]
22. Puik, E.; Lafeber, R. Axiomatic Design. 2023. Available online: https://www.dspe.nl/wp-content/uploads/2023/06/Mikroniek-2023-3-Axiomatic-Design-1.pdf? (accessed on 14 December 2025).
23. Mabrok, M.A.; Efatmaneshnik, M.; Ryan, M. Including non-functional requirements in the Axiomatic Design process. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings*; IEEE: Piscataway, NJ, USA, 2015; pp. 54–60.
24. Nordlund, M.; Lee, T.; Kim, S.-G. Axiomatic Design: 30 Years After. In *Proceedings of the ASME 2015 International Mechanical Engineering Congress and Exposition*; American Society of Mechanical Engineers (ASME): New York, NY, USA, 2015; Volume 15.
25. Rand, D.A. Mapping global sensitivity of cellular network dynamics: Sensitivity heat maps and a global summation law. *J. R. Soc. Interface* **2008**, *5*, 59–69. [CrossRef] [PubMed]
26. Lu, K.; Zhang, K.; Zhang, H.; Gu, X.; Jin, Y.; Zhao, S.; Fu, C.; Yang, Y. A Review of Model Order Reduction Methods for Large-Scale Structure Systems. *Shock Vib.* **2021**, *2021*, 6631180. [CrossRef]

27. Rudolph, S. Übertragung von Ähnlichkeitsbegriffen. Habilitation Thesis, Faculty of Aerospace Engineering and Geodesy, University of Stuttgart, Stuttgart, Germany, 2002. Available online: https://www.academia.edu/57519499/%C3%9Cbertragung_Von_%C3%84hnlichkeitsbegriffen_ (accessed on 1 October 2026).
28. Freund, R.W.; Hoppe, R.H.W. *Stoer/Bulirsch: Numerische Mathematik 1*, 10th ed.; Springer: Berlin/Heidelberg, Germany, 2007. [CrossRef]
29. Zuguang, G. Complex Heatmap Visualization. *iMeta* **2022**, *1*, e43. Available online: https://onlinelibrary.wiley.com/doi/10.1002/imt2.43 (accessed on 13 December 2025). [CrossRef] [PubMed]
30. Zhu, J.; Wang, Z.; Shen, Z.; Wei, L.; Tian, F.; Liu, M.; Liu, S. ReorderBench: A Benchmark for Matrix Reordering. *IEEE Trans. Vis. Comput. Graph.* **2025**, *31*, 4126–4140. [CrossRef] [PubMed]
31. Chauhan, P.; Sah, K.; Kaushal, R. Design, modeling and simulation of suspension geometry for formula student vehicles. *Mater. Today Proc.* **2021**, *43*, 17–27. [CrossRef]
32. Design Cockpit 43® and DC43® Are Registered Trademarks of the Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH, Germany. Available online: https://www.iils.de/en/ (accessed on 18 December 2025).
33. Caro, S.; Wenger, P.; Bennis, F.; Chablat, D. Sensitivity Analysis of the Orthoglide: A Three-DOF Translational Parallel Kinematic Machine. *J. Mech. Des.* **2006**, *128*, 392–402. [CrossRef]
34. Wertz, J.R.; Wiley, J. *Space Mission Analysis and Design*; Springer: Berlin/Heidelberg, Germany, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.