Forschung im Ingenieurwesen (2023) 87:1333–1346
https://doi.org/10.1007/s10010-023-00699-y

ÜBERSICHTSARBEITEN/REVIEW ARTICLES

Potentials and challenges in enhancing the gear transmission development with machine learning methods—a review

Stefan Sendlbeck[1] · Matthias Maurer[2] · Michael Otto[1] · Karsten Stahl[1]

Received: 6 January 2023 / Accepted: 23 July 2023 / Published online: 22 August 2023

Abstract

The electrification of vehicle powertrains and the expected engineering labor shortage are ongoing key challenges in the gear transmission development. Because traditional methods reach limits, the solution is further automating the design process while enabling flexible and optimal design solutions even with rapidly changing constraints and requirements. We therefore review the current design process, review state-of-the-art methods for automated gear transmission design, and evaluate their potential and the challenges in combination with using machine learning methods. In focus are grammars and graph grammars in particular, which offer an approach to represent and generate the relational structure of transmission topologies or shaft arrangements. Other potential approaches are knowledge-based engineering, which allows to choose various predefined expert design solution and combine them to new designs, and constraint programming for gear transmission generation. Combining these methods with latest advances in reinforcement learning, machine learning for inverse problem-solving, and graph neural networks offers promising capabilities for automatic topology generation and dimensioning of gear transmissions.

Potenziale und Herausforderungen bei der Verbesserung der Getriebeentwicklung mit Machine-Learning-Methoden – Ein Review

Zusammenfassung

Die Elektrifizierung von Fahrzeugantrieben und der zu erwartende Arbeitskräftemangel bei Ingenieuren sind zentrale laufende Herausforderungen bei der Entwicklung von Zahnradgetrieben. Da herkömmliche Methoden an ihre Grenzen stoßen, liegt die Lösung in der weiteren Automatisierung des Entwicklungsprozesses und der schnellen Bereitstellung flexibler und optimaler Entwurfslösungen, auch bei sich schnell ändernden Randbedingungen und Anforderungen. Es werden daher der aktuelle Entwicklungsprozess und die aktuellen Methoden für den automatisierten Getriebeentwurf untersucht und deren Potenzial und die Herausforderungen in Kombination mit dem Einsatz von Machine-Learning-Methoden evaluiert. Im Fokus stehen dabei Grammatiken und insbesondere Graphgrammatiken, die einen Ansatz zur Repräsentation und Generierung der relationalen Struktur von Getriebetopologien oder Wellenanordnungen bieten. Weitere mögliche Ansätze sind das Knowledge-based Engineering, das es erlaubt verschiedene vordefinierte Expertenlösungen auszuwählen und zu neuen Entwürfen zu kombinieren, und das Constraint Programming zur Generierung von Getrieben. Die Kombination dieser Methoden mit den neuesten Fortschritten im Bereich des Reinforcement Learning, des Machine Learning für inverse Probleme und der Graph-basierten neuronalen Netze bietet vielversprechende Möglichkeiten für die automatische Topologieerzeugung und Dimensionierung von Zahnradgetrieben.

✉ Stefan Sendlbeck
stefan.sendlbeck@tum.de

Matthias Maurer
matthias.maurer@tum.de

Michael Otto
michael.k.otto@tum.de

Karsten Stahl
karsten.stahl@tum.de

[1] Gear Research Center (FZG), Institute of Machine Elements, Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

[2] Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

1 Introduction

The latest challenges in drivetrain und transmission development come along with emerging advances in machine learning. According to the "2022 State of the Gear Industry" survey of the American Gear Manufactures Association [1], many see the electrification of vehicle powertrains as a technology disruption and one of the most important ongoing challenges. Additionally, many expect or are already facing a shortage of labor in general, but especially in smaller companies and also on an engineering or expert level. Due to the increased dynamic of the gear transmission development, companies demand for faster obtained, but high-quality solutions. Because traditional methods reach limits, automation and machine learning are key for future success. The latest advances in artificial intelligence (AI) and machine learning in particular offer widespread opportunities to learn from data. Applications in various fields show that machine learning can improve the performance of complex processes or can automate until recently impossible tasks.

With the present work, we consider methods for automated gear transmission design and evaluate their potential and the challenges in combination with the use of machine learning methods. The focus of the work is on methods, which are fundamental for machine learning algorithms in order to obtain flexible designs under various constraints and requirements. Examples are grammars and graph grammars in particular, which offer an approach to represent transmission topologies or shaft arrangements. With knowledge-based engineering (KBE), various predefined expert design solution can be chosen and combined to new designs. Another potential approach for gear transmission generation is constraint programming (CP).

The present paper first contains a review of the literature of gear design programs, of methods for automated design and transmission synthesis, and of optimization and machine learning methods for gear transmission design. Secondly, we provide the current state of gear transmission development before evaluating the potentials and challenges of a gear transmission development approach enhanced by machine learning methods. Lastly, we discuss the potentials and indicate key areas of further research.

2 Methods

Although nowadays the design process of gear transmissions is based on partially automated computer aided methods, expert knowledge and experience are still necessary to a large extent [2]. Computer-based applications as part of the design process can be distinguished into multiple groups. The first ones are the already widely used gear design and mainly analysis programs. The second ones are methods for automated design synthesis like grammars and KBE. Additionally, there are more general methods for optimization and machine learning as part of AI.

2.1 Gear design programs

Available programs can be divided with regard to their functionality into software for gear component design and software for overall system design.

Prominent programs as an example for gear component design are STplus (Research Association for Drive Technology (FVA) [3]), GearEngineer (GJW Technology [4]), LDP (Ohio State University [5]), Transmission3D (Ansol [6]) and STIRAK (FVA), which allow load capacity as well as geometric design calculations. STplus and GearEngineer both contain parameter-based tooth geometries representations and the calculation follows the standards DIN, AGMA, and ISO. In contrast, STIRAK is based on numerical finite element method (FEM) simulation and therefore it is possible to carry out more detailed tooth contact analysis under load [7]. Further programs are DZP (FVA) for the analysis and optimization of the dynamic excitation behavior as well as MDESIGN LVR for the load distribution calculation and optimization of the tooth flank [8, 9].

To achieve a more holistic design process, overall transmission development programs such as RIKOR (FVA), GAP (FVA), MASTA (Smart Manufacturing Technology Ltd [10]), Romax (Romax Technology Ltd [11]), WTPlus (FVA) and KISSSoft (KISSSoft AG [12]) are in application. The focus of RIKOR is on comprehensive deformation analysis and the resulting impact on the gears, shafts and bearings. By specifying common requirements such as the desired power and transmission ratio, GAP creates quick first drafts of transmission designs based on various predefined transmission topologies [7, 13–17]. WTPlus is an analysis tool for efficiency and heat analysis at system level [18]. Masta and Romax are programs for standard-based design, NVH analysis and optimization of shafts, bearings and gears in the overall transmission context [19, 20]. A broad software program is KISSSoft, which contains modules for the design of single machine components and extensions for overall gear transmission design. It also includes the partially automated creation of graded transmission variants based on user specified parameter ranges [21, 22].

Next to commercial software programs, there are many individual software solutions in companies that range from analysis tools to automated synthesis solutions for specific, well-known gear types and topologies. However, these in-house software solutions are difficult to cover, because they are mostly confidential.

In summary, software-supported transmission design already provides a wide range of options for automating individual areas of transmission development. However, the available programs are still predominantly limited to the detailed analysis of already defined geometries and cannot perform a detailed synthesis from scratch.

2.2 Methods for automated transmission design

Automatically designing gear transmissions is a complex task due to the infinite possible design solutions. Following the approach of computational design synthesis by Cagan et al. [23], the process includes the generation, representation, evaluation and guidance of designs. Especially guiding and generating new design drafts requires adequate methods to cope with extensive requirements and strong dependencies of properties. To address the shortcomings of today's software-supported transmission design, methods like grammars, KBE and others could be promising for automation, especially in combination with using machine learning.

Grammar is defined by Merriam-Webster as "the set of rules that explain how words are used in a language" [24]. Equivalent to languages, the synthesis in technical development also follows fundamental rules. The first one to use grammar proceeding from linguistic science in a formal, technical sense was Chomsky who developed the string grammar [25]. Since then, a variety of other grammars have been developed, such as graph, array, tree, or shape grammars [26]. In general, every grammar consists of four parts, which are nonterminal vocabulary, terminal vocabulary, production rules and initial objects. Starting from the initial object, applying production rules modifies the object until it is only made of terminal vocabulary, or a user defined stop condition is met. The production rules usually follow a decision tree style and therefore have a causal structure with a left side ("If ...") and a right side ("Then ..."). If the left-side condition is met by the object, the right-side modification is applied to it, thereby deriving a new object [27]. Despite the large variety of different grammars, graph and shape grammars are the ones mainly used for technical product development [28].

The vocabulary of graph grammars consists of edges and nodes. Edges usually represent energy, material, or signal transmissions, while nodes, depending on the specific application, often represent individual product components. For gear transmissions, the nodes are usually used to represent shafts or gears and edges to represent the power transmission and the geometrical connection or contact. Since individual properties of the components are usually of great importance for technical development tasks, additional parameters are often assigned to nodes that can also be modified by the production rules [29]. In order to apply rules, it is necessary to find matches in the graph regarding the left-side conditions of the rules. Common algorithms for this search are Depth-first or Breadth-first algorithms [27]. However, machine learning on graph is getting traction recently and more sophisticated searches and evaluations of graphs become possible (see Sect. 2.3) [30]. For example, You et al. [31] proofed that latest advances in machine learning on graphs allow to generate new molecule topologies represented as graph structures.

A basic example of a graph grammar with rule application is illustrated in Fig. 1. The advantages of graph grammars are, for example, that they are well visualizable and that there are software programs to create graphs and apply rules.

Examples are GrGen, Design Compiler 43 with many interfaces, i.e. to CAD-programs, and GraphSynth with integrated search algorithms. A main disadvantage is that there is no commonly agreed language yet for creating graphs and therefore only limited transferability between systems is possilbe [28]. Graph grammars were already used to design satellites [34], an electric toothbrush [35] or epicyclic gear trains [36].

Shape grammars, in contrast, have a vocabulary consisting of shapes, which can be points, lines, surfaces, geometric bodies, or higher-dimensional hyperplanes [27]. Shape grammars were originally developed by Stiny and Gips to create paintings and sculptures [37]. They have already been used successfully in architecture to develop different variants of buildings [38]. Later they were also used successfully in areas of mechanical engineering such as the design of a robot arm [39], coffee machines [40], digital cameras [41], vehicles [42], or as an example for a commercial application for the design of airplane tubing [43]. Shape grammars are not based on a symbolic representation and therefore the production rules are applied directly to the shapes of the technical components [44]. A Disadvantage is that, in contrast to graph grammars, there are no comparable translation programs from grammar to CAD programs. Furthermore, the search process for the production rule application is significantly more complex; current approaches include computer vision algorithms and decomposition into sub-shapes [28].

Since the gearbox design is a complex inverse problem, ending the rule application depending on the terminal vocabulary does not make sense. Instead, optimization methods such as Simulated-Annealing [44], Genetic Algorithms [45], or Burst-Algorithms [29] are usually used.

Grammars have also already been used for the automated synthesis of gears transmissions. Lin et al. [44] automatically synthetized a 5-speed transmission by using a combined shape and graph grammar with 9 parametric and topological production rules. Königseder et al. [29, 46] present a similar approach for gear transmission generation with ten production rules and the goal of investigating differ-

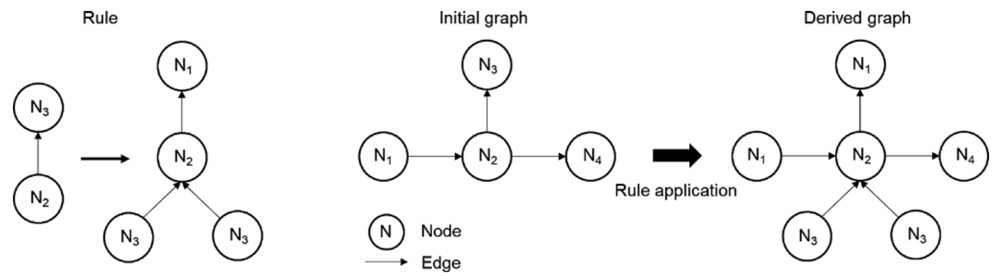

Fig. 1 A basic example of a graph grammar with rule application in the style of [32, 33]

ent strategies for rule application. For planetary gears, Wolf symbols are an approach to symbolize gear topologies [47].

Another method for automated transmission design besides grammars is KBE, which is a combination of several areas such as AI, CAD and object-oriented programming [48]. The focus is on capturing, saving and reusing the respective product knowledge with the aim of automating repetitive, non-creative tasks of the design process [49]. KBE is rule-based deterministic so that the system always generates the same results according to the expert definition. Furthermore, KBE programs are often strongly linked to CAD engines and can therefore directly create technical drawings or carry out analyzes [50]. Common commercial KBE systems are ICAD (Dassault Systèmes), AML (TechnoSoft), GDL (Genworks International), KnowledgeFusion (Siemens NX) and KnowledgeWare (Dassault Systèmes) [51]. KBE were used successfully for automotive body-in-whites analyzes [48], gear shaft of an aircraft engine [52] or ship structural design [53].

Berx et al. [54] proposed another approach to automated transmission synthesis with a CP algorithm. For this purpose, they defined many restrictions, which were derived from user requirements, design rules or physical laws. All possible combinations were then created in a two-stage process. With an increasing number of gear stages, the computational effort became too high despite many restrictions. This is why an additional clustering algorithm was implemented to independently add new restrictions.

Fauroux et al. [55] developed a method for the automated arrangement of transmission components. For this, an algorithm that has stored all common types of gear stages (e.g. planetary stages, conical stages etc.) replaces the stages with prismatic and rotary connection points. The former is introduced wherever a variable change in length, the latter where a rotational change is possible. An optimization problem is then solved that on the one hand minimizes the distance between the last gear shaft and the desired output position and on the other hand minimizes the length of the prismatic connection points ensuring that the shortest arrangement is found.

Stangl [56] uses graphs for topology representations of complex planetary gears. For finding concepts, he combines graph and shape grammars on the basis of Lindenmayer systems.

Kurth [57] describes a methods for complex planetary gear synthesis with Helfer diagrams and equivalent lever models for multiple operating conditions. He reduces the search space by an efficiency approximation method.

2.3 Optimization and machine learning in gear design

Unlike a forward or direct problem, where causes lead to effects, gear design can be seen as an inverse problem. For example, the geometry and material choice of a gearset cause its properties such as weight and load-carrying capacity. The inverse problem would be finding one or all suitable geometry material combinations for given properties. In contrast to the forward problem, inverse problems generally have various solutions or cannot be solved at all.

2.3.1 Optimization in gear design

Common approaches for solving the inverse problem are optimization algorithms. Parlow [14] integrates an optimization with a global simulated annealing algorithm in the gear design process of the GAP (FVA) software. The optimization problem includes the weighted cost function of mass, efficiency and noise of a gearset or the whole gearbox consisting of gears and shafts. The constraint is fulfilling the minimum load-carrying capacity. The mass is considered as the ratio of the mass of all parts with respect to the input torque. The load-depended gear losses serve as a criterion for efficiency, while the characteristic value for noise is a combination of excitation level and response function. Fürst et al. [58] extend this concept in GAP (FVA) by a mixed integer nonlinear mass optimization including detailed shafts with shoulders, splines, catalogue bearings, and fixing elements. Each shaft section must have a diameter in accordance with the conjugating and chosen machine element. But they must also fulfill the requirements for load-carrying capacity and notch effects by considering the diameter transition from shaft section to section. The algorithm performs iterative searches along the shaft sections. The approach is based on two fixed shaft topologies of input/output shafts and intermediate shafts. Holder et al. [59] illustrate gear design automation with a graph-based design language and a package optimization. They also use GAP (FVA) to generate a gear design including synthesis and analysis and represent the transmission topology and parameters with a graph structure. In order to meet the requirements for input and output locations, the approach contains a nonlinear gradient-free optimization, which is constraint by a CAD kernel collision detection. The optimization parameters are manually chosen and bearings are with parametric rules next to the gearset. With the graph representation of the final gearset the approach contains a fully parametric housing description and forward generation. Brahim [60] approaches the mass optimization of a one-stage gear transmission geometry. The constraints of the inverse problem are the tooth root strength, the surface durability, the torsional stiffness of the shaft, and the shaft distance. First, an analytical model according to AGMA standards [61] is used as the basis for a global optimization to obtain a first level solution. The next step is CAD modelling the solution and deriving a surrogate model due to computational efficiency reasons. The result of the subsequent heuristic optimization with a genetic algorithm is the final gear design. Rai et al. [62] intend to minimize the gear volume of a helical gearset with a real-coded genetic algorithm. They use the profile shift coefficients, the module, the face width, the helix angle, and the number of teeth as design variables and constraint the problem by the specific sliding, the transverse contact ratio, the face width and strength requirements based on ISO 6336 [63] standards. Other works, for example, Simon [64] and Kohn et al. [65] include an optimization of the tooth flank properties and modifications and its manufacturing with load capacity and noise level constraints.

Often optimizing one objective such as the mass is not sufficient in gear design. Multi-objective optimization is a method to get design options and the Pareto front. Younes et al. [66] optimize a single-stage helical gear with respect to power loss and vibrational excitation. They calculate the power loss based on tooth friction and alternatively based on a thermal network with heat sources. The root mean square error of the transmission error oscillation represents the dynamic excitation. It originates from rigid body model calculations. To find the Pareto front, they use the Non-Dominated Sorting Genetic Algorithm NSGA-II with the bending stress in the tooth root and a minimum contact ratio as constraints. The optimization parameters are the macro geometry values of the pressure and helix angle in combination with the micro geometry values of the length and amount of tooth profile modification with crowning. The optimization approach is either using the macro and micro parameters simultaneously or subsequently. Maputi et al. [67, 68] use the volume and the center distance of a three-stage spur gear as objectives in order to obtain a maximum compact design. The problem constraint is the minimum required load-carrying capacity based on the bending and contact stress according to AGMA standard [61] calculation. With a bounded objective method in combination with teaching-learning based optimization the problem contains only the volume as a main objective, while the second objective, the center distance, is formulated as a constraint. Sabarinath et al. [69] use a similar approach to optimize the volume and overlap ratio of a helical gearset by applying a parameter adaptive harmony search algorithm. The optimization variables are the module, the helix angle, the gear width and the number of teeth. The constraints are minimum load-carrying capacity values according to ISO 6336 standard [63] in combination with geometrical restrictions. Artoni [70] describes another approach to find the global pareto optimal solution with a direct search algorithm for transmissions.

2.3.2 Machine learning in gear design

Advances in AI and machine learning in particular offer various possibilities to enhance the design process and to support solving the inverse problem. However, only few approaches have been published yet in the field of gear transmissions.

Lin et al. [44] and Königseder et al. [46] describe graph grammars used for computational design synthesis of transmission topologies, as illustrated in Sect. 2.2. Doumouchel et al. [71] and Masfaraud et al. [72] use a similar approach combined with clustering to generate detailed geared transmission designs including shifts. They describe component properties and interactions with an object-oriented design formulation. A rule database with a decision tree is used to perform an exhaustive graph generation of topology solutions. It contains components like shafts, gearsets, clutches, shifting elements, and power inputs/outputs. Subsequently, graph algorithms (depth-first search) are used to remove isomorphism and invalid topologies that do not meet the kinematic requirements. A wireframe modelling of the remaining subset of solutions adds spatial information and allows to detect further invalid solutions. The yet remaining subset is the basis for machine learning clustering. Each topology solution gets properties added by estimating, for example, the cost per element or the efficiency based on the number of tooth contacts. They use the cost, efficiency and the number of gear meshes for multidimensional scaling in a seven dimensional vector space. It is the basis for clustering and identifying groups of similar transmissions. Distinct solutions of each cluster, such as the one with favorable properties, are chosen for further 3D CAD refinement and optimization. Therefore, they continuously optimize the position and size of all components with a genetic algorithm and package constraints. Discontinuous values, i.e. the number of teeth, are obtained by discrete optimization. Urbas et al. [73] present an approach for predicting tooth root bending stresses of spur gears. There are standards, such as ISO 6336 [63] or AGMA 2101 [61], with analytical formulas for involute gears. However, the authors evaluate various machine learning methods in order to predict the values of polymer gears with a curved line of contact, for which these standards not apply. A dataset with the resulting maximum stress in the tooth root is obtained by geometrical variation and subsequent FEM simulation for various forces and Young's moduli. It is the basis for machine learning model training for stress prediction. Models evaluated are linear regression, support vector machine, k-nearest neighbor, neural networks, AdaBoost, and random forest regression. The latter two showed the best results in the provided work.

2.3.3 Advances in machine learning

Next to machine learning methods applied in gear design already (see Sect. 2.3.2), there are various advances in machine learning in general that are promising for product design. Many of these methods might be helpful to the application in gear design. Because the field of general machine learning is broad, we will not cover all aspects. But we will give an indication of promising and relevant advances in graph neural networks (GNNs), reinforcement learning and machine learning for inverse problem solving.

Research on GNNs has seen a lot of traction lately as summarized in [30, 74, 75]. From representing physical structures to knowledge, graphs are a versatile basis for analysis, generation, and problem solving. Latest GNNs are powerful machine learning methods and have proven their outperforming capability. Zhou et al. [75] gives an overview of current applications and challenges of GNNs. Many problems and applications have an inherent relational data structure, which makes the use of graph representations obvious, i.e. chemical molecules. Advances in GNN research allow to perform graph matching and clustering, biomedical and molecular engineering for interface and reaction prediction, and representation learning on knowledge graphs. Additionally, GNNs are used in generative models, for example, to create new chemical structures or knowledge graphs. In complex combinatorial problems, i.e. the traveling salesman problem, GNNs can improve the optimization. Furthermore, next to graph level and node level classification, GNNs are also suitable for link prediction and are therefore applied in recommendation systems. For some scenarios, a structural graph representation is not present directly. Applications with incorporated or assumed relational structure are, e.g., fast image classification, visual reasoning, text classification, or sequence labeling. Although GNNs are powerful, some challenges for research remain, which are the robustness, the interpretability, the possibility and extend of graph pretraining, and complex and dynamic graph structures [75].

Next to supervised learning, deep reinforcement learning is one of the important fields of machine learning and it has seen intensive progress in the past few years [76]. It is meant for sequential decision making with the goal to learn the best actions in various situations to fulfill a long-term goal. It is not based on training datasets, but on learning by exploration and exploitation. Choi et al. [77] suggest a deep reinforcement learning approach to assist in task-oriented product design. The goal is to change the diameters of a cylindrical pot at various heights in order to be able to catch as much water as possible when pouring it in, but simultaneously losing the least water when shaking the pot. Using a simulation environment for evaluating and rewarding the pouring and shaking scenario in combination with an actor-critic proximal policy optimization (PPO) algorithm, the reinforcement learning methods is able to create task-oriented pot designs. Hayashi et al. [78] show another application for optimum design of a plane frame. They combine deep reinforcement learning with metaheuristics of simulated annealing and particle swarm optimization. Using deep Q-learning, they approach to optimize the cross-section design of planar frame structures with respect to minimal volume and stress constraints. It is a combinatorial problem, because only discrete catalogue diameters are available. Jang et al. [79] use deep reinforcement learning for generative design. They show a case study of automotive wheel design. The goal is to generate 25 new wheels with maximum diversity. Therefore, they use a neural network to approximate and speed up the automotive wheel topology optimization. They combine that with a reinforcement learning PPO algorithm with autoencoder regularization in order to achieve the diversity by pixel difference or structural dissimilarity. Ruiz-Montiel et al. [80] use reinforcement learning in combination with shape grammars for design generation. They show a general approach based on Q-learning and demonstrate it on an architectural design example of automatic floor plan generation for two-person houses with various constraints. Mirhoseini et al. [81] show that reinforcement learning can improve the quality and time of chip placement. They place nodes of chip netlist, which is represented as a graph, into the two-dimensional space of a chip canvas. The goal is to optimize the power, performance and area while preventing infeasible solutions. In the course of which, they use supervised representation learning for predicting the placement quality.

Solving the inverse problem is an important task in engineering and various machine learning approaches try to increase quality and speed. For example, Ardizzone et al. [82] suggest invertible neural networks for inverse problems. The underlying idea is to use learning of the forward problem and thereby learning the inverse problem implicitly. They show the validity, i.e., by applying the approach on functional state analysis of biological tissue. Holl et al. [83] propose a different approach utilizing higher-order gradients for applications such as fluid flow prediction. While most state-of-the-art machine learning algorithms rely on first order gradients, second order gradients are often decisive to physical or engineering problems. Therefore, they use a hybrid optimization approach with so-called physical gradients, the embedded inverse-problem solver, replacing the gradient of the physical process.

2.4 Methods conclusion

When it comes to automating the gear design, there is various methodical support such as grammars, KBE, or CP. However, most approaches today rely on classic parameter optimization to solve the inverse gear design problem with given or restricted topologies. They range from single-objective to multi-objective approaches in order to find the pareto optimal solutions and target preferably macro geometric design parameters such as gear size and shape to micro geometric gear profile modifications. Only rarely, flexible design solutions by detailed topology variation are optimized to its fullest potential. There are approaches based on graph and shape grammars. However, these are mostly combined with exhaustive searches and are therefore time consuming at execution with high computational costs. Machine learning is not widespread in gear transmission discovery and, for example, used predominantly for particular property predictions including common feature engineering (e.g. material features [73]), model training and evaluation steps. However, advances especially in deep learning offer promising opportunities to utilize graphs due to the GNNs and their potential for graph analysis and handling, reinforcement learning for combinatorial optimization and sequential decision making based on rules, as well as approximation learning and neural network architectures for tackling inverse problems.

The question arises, what the potentials are of using grammars, KBE, and CP as a basis for machine learning methods in order to improve gear design and explore solutions spaces for diverse topologies.

3 Current state of gear transmission development

In the current transmission development, the knowledge of experts is still important and only few parts of it are automated [14]. The gear development process follows the iterative and stage-wise synthesis and analysis of the product development V-model according to VDI/VDE 2006:2021 [84]. It can be divided into the stages of target and requirement definition, topology design, component design, structural design, and system evaluation.

In the first step, the definition of targets, the requirements, boundary conditions and target values that the gearbox should achieve must be specified. Usually, these goals are set manually by the client and/or the contractor [85]. A variety of methods such as financial analyzes, interviews, and patent research are used for this task. In addition, the targets are often inspired by existing transmissions and their properties [86].

The following step is the generation and analysis of the topology with the basic gear type definition. In addition, the number and type of components is determined as well as the interactions between them [87]. The choice of topology still largely depends on the knowledge of responsible engineers, which makes it very person-dependent and often hard to find optimal solutions. For this reason and to simplify the process, simulation-based, model-based and mathematical optimization methods are used in addition (see Sect. 2.3.1 and 2.3.2). In summary, it can be stated that the selection of the topology has already been simplified to a small extend by software-based applications (see Sect. 2.1). But it is still mostly a manual task with geometrical and functional constraints such as the axis position or the number and type of gearsets.

The goal of component design is to iteratively define parameters for the individual gear components (especially gearing, shafts and bearings) so that they meet the requirements. Usual requirements for the components are ensuring the load-carrying capacity, a high efficiency, a minimal dynamic excitation, etc. [14, 88]. In general, the gear dimensioning is started first, then a shaft dimensioning and finally a dimensioning of the bearings is carried out. It is common to use various software for component design, but mostly restricted to component analysis. Depending on the program, the analysis of the gears, shafts and bearings is completely automated or at least highly software assisted. Next to integrated tools, it is often carried out with additional, external analysis programs such as FEM or multibody simulation tools [89].

In the structural design, the components are spatially arranged in relation to one another and the remaining transmission components such as the housing, seals, circlips and other machine elements are added. Like the topology design, the procedure is still largely based on the knowledge of experts. Following the generation of the structure, analyzes such as FEM are usually carried out for validation [90]. For this purpose, simulation and analysis software programs are used.

In the final step, an overall system evaluation is carried out in which the developed gearboxes are verified against the requirements specified in the target definition. For this purpose, the properties of the transmissions need to be analyzed [91]. This is often also done with software assistance. A final evaluation based on the analysis results is then usually carried out by the development team or responsible stakeholder.

All in all, the current development process of gear design seldomly contains KBE. Graphs are used to find suitable topologies for planetary transmissions with given transmission ratio constraints (for example, see Kurth [57]), but are not widespread used as grammars for design generation. This also is the case for commonly used shape representations.

4 Potentials and challenges in gear transmission development

Various possible topologies ranging from combinations of spur, bevel and planetary gears to machine element choice and their order on the shafts lead to challenges in gear design. Parametric scaling is only a small part of the solution, but much of it is combinatorial design and optimization. Therefore, the presented methods of grammars, the KBE, and the CP could make a crucial contribution to the automation of the transmission design in the future. For this reason, we analyze the potentials and disadvantages of the respective procedures including challenges in the machine learning application and data collection.

4.1 Grammars

Grammars have a great potential to enable automation in the gear topology selection and in shaft design. The reasons are, on the one hand, that they have already been successfully used for this or a similar application and, on the other hand, that they have a lower computational effort compared to other methods like CP. Regarding the type of grammar, graph grammars have the advantage over shape grammars of a lower computational effort, already proven compilers for conversions into CAD models and programs to create them (e.g. GraphSynth, GrGen or Design Compiler 43) [28, 92]. Since shape grammars enable a spatial arrangement, a hybrid form of both types could be of advantage.

Despite the potential of grammars, they also have considerable disadvantages. One of them is the current lack of uniform standards for creation and use of grammars, which means that transferability or interoperability between different systems is difficult [28]. However, this can be considered a minor problem, as newly emerging methods are often affected by a lack of standards at first. Attempts are also being made to remedy this disadvantage by using established modeling languages such as SysML for support [93].

Another disadvantage is the complexity of grammars. In existing gear design applications, in which grammars are used, only simple stage transmissions were designed without a large variety of production rules. In these design processes, only the components gear, shifting elements and shaft were considered with few rules in total [29, 46, 71]. Additional rules lead to an exponential increase in the model complexity of grammars. To limit this complexity, unfavorable topologies must be sorted out early by using suitable optimization methods (see Sect. 2.2). Due to the mostly random application of rules, many invalid or unfavorable variants are created. It would be an advantage if topologies are not only sorted out after creation, but rather avoided before production rule application. A possible solution is the combination of grammars with reinforcement learning. An example of this is a method published by Stump et al. [94], in which a shape grammar was combined with a char-RNN to modify the rule application. After the learning process, the char-RNN was applied by assigning the production rules that were most promising in the training to a higher probability than the others. With this method, the number of unsuitable concepts could be significantly reduced. It should be noted that the use of reinforcement learning to the production rule selection can introduce a bias. Furthermore, GNNs as described in Sect. 2.3.3 offer new possibilities in high quality and fast graph analysis. They should be considered for further research in combination with graph generation as shown by You et al. [31] or graph analysis and comparison in general.

Another problem of grammars is that they are very time consuming to create and implement. The manual definition of suitable production rules represents a considerable development effort [95]. Consequently, grammars are very subjective with regards to the knowledge of their developer. A fundamental improvement by which significantly shorter development times and potentially better grammars could be achieved would therefore be the (partly) automatic creation and the self-learning of production rules. The method of Computational Evolutionary Embryogenesis (CEE) published by Yogev et al. [96] could be a possibility to implement this to grammars. CEE is strongly based on genetic algorithms from machine learning. The main difference is that they are not applied to the application sequence of the rules, but to the rules themselves. So, at the beginning, several rule sets are initialized with random rules and then structures are formed from them, which are then evaluated. The rule sets of the more suitable structures have a higher probability of passing their rules on to the next generation. The selected rule sets are also changed by mutation and recombination like by genetic algorithms. Another method for automated rule generation is the Statistically Derived

Table 1 Weaknesses and potentials for improvements of grammars

Weaknesses	Potential for improvements
Transferability and interoperability with other systems	Interpreter already available, so far only with very limited functionality Established modelling languages like SysML for support
Exponential complexity	Learning and adapted rule selection by combining with reinforcement learning Graph analysis with GNNs
Lack of standards	Definition of standards
Creation and implementation of production rules laborious	Machine learning, autonomous creation of rules Computational evolutionary embryogenesis Statistically derived shape grammar
Lack of rule analysis methods	Network-based rule analysis method

Shape Grammar created by Orborn et al. [95, 97] in which the vocabulary is determined by analyzing existing products (in this case vehicles). For this purpose, a principal component analysis was carried out in a database to identify similarities or differences, weigh them based on their variance in the vehicle designs and then automatically derive product rules from it. Since the method was developed primarily for the industrial design, it is not intuitively transferable to the creation of gear topologies. In summary, there is still research need in the field of automated rule creation, because the result will have a significant influence on the future of grammars in technical concept development [28].

Another drawback of grammars are the currently missing analysis and evaluation methods of rules. Even if the definition of the rules may be automated in the future, an additional, independent analysis should be available for checking and to ensure a deeper understanding of the grammar by the responsible engineers. An example of such an approach is the Network-based Rule Analysis Method (NRAM) published by Königseder and Shea [46], which can be used to analyze graph grammars. The functionality of NRAM is based on transition graphs, which represent an overview of all possible rule combinations. Analyzes of the graph grammar were carried out using the breadth-first algorithm, integrated rule analyzes from the GrGen.NET program and manual examinations based on clear visualizations. Even if NRAM is a very simple and only partially automated analysis method, it can be used to evaluate the most important properties of a graph grammar. Apart from that, there is still a need for research to develop further analysis methods, for example also for the evaluation of shape grammars. A summary of the weaknesses and potentials for improvements of grammars is given in Table 1.

4.2 KBE

When KBE came up, it was treated as a very promising method. This initial enthusiasm has lately decreased significantly, as KBE, except for a few automotive and aviation applications, could not be established in industry [49].

KBE is particularly suitable for automating repetitive, rule-based and parametric tasks as a forward problem that require little creativity. While certainly transmission development largely consists of such repetitive tasks, many areas, such as the topology generation, still require expert knowledge. Properties strongly depend on each other and many can only be evaluated sufficiently at a late stage of the design process. This is where KBE has its biggest drawbacks. In contrast to grammars, KBE always creates the same solution according to the expert definition without the possibility of exploring the design space broadly for unknown options. Therefore, KBE does not change the current iterative, experience-based design process, but represents a computer-based automation of it. Another problem of KBE is generalization. It is often adapted to a specific application and cannot be easily applied for similar products. New products often require time-consuming reprogramming of the KBs and, as a result, the loss of knowledge, misapplications, increased program maintenance costs and the danger of insufficient knowledge utilization [98]. Methodological approaches are a solution. The best-known example of this is MOKA (Methodology and Tools Oriented to Knowledge Based Engineering Applications), which supports the developer in creating a KBE system [50]. Apart from that, a (partially) automated, generalized rule generation could be a considerable improvement of KBE.

The potential of KBE for application in gear design is primarily the efficient scaling or modification of gear concepts for rapid adaptation of specific application cases. It is also an advantage that the KBE is the only one of the presented methods yet that can directly access the CAD engine and thus directly derive solids or technical drawings. Machine learning algorithms can help to make KBE more flexible in choosing the right design options.

4.3 Constraint programming

The model-based CP approach is an approach that tries to consider all possible topologies with the goal of determining the optimal solution. A central weakness, however, is that many constraints must be defined for this. As with the production rules of the grammars, these must be created by engineers, who thus have a strong influence on the resulting outcome. Furthermore, case-based overfitting may occur in the process.

Possible improvements for this include automated constraint generation with machine learning, where they are learned or adapted from training data. This method is called constraint learning or constraint acquisition [99]. With constraint learning, only the parameters of already existing constraints are changed, but not the basic formal structure. As an example, the initial constraint “gear diameter < 400 mm” could be given, and the machine learning algorithm would then only adjust the value 400 mm. For soft constraints that may be violated, if necessary, but which generate a negative reward for the algorithm, the weights can also be adjusted with machine learning. This is often implemented through clustering methods [54]. In addition to constraints, CP needs an efficient strategy on how to explore the search space. An approach to this as an example was presented by Cappart et al. [100], in which the CP was extended by a reinforcement learning algorithm. This enabled a much more efficient solving of the CP by ensuring that once a solution was found, new constraints were imposed to ensure that subsequent solutions must always be better. The application potential of CP in the transmission design process is mainly in topology selection. It therefore is a competing approach to grammars. However, due to the constraints, which all must be fulfilled, the CP has less flexibility compared to the grammars and is more complex to create.

4.4 Machine learning application

The application of machine learning in gear transmission development faces some general challenges. According to Liu et al. [101, 102], three major contradictions, the data structure and availability (high-dimensional data and small samples size), the machine learning model complexity and accuracy, and the knowledge source (learning results and domain knowledge) apply in machine learning research of materials design and discovery. They are reviewed in the context of gear transmission development.

- Data structure and availability

Only few data of whole gear transmissions are normally available due to the elaborate design process, diverse applications, and confidentiality restrictions. Additionally, the description of gear transmissions requires heterogeneous data such as functional relations between parts and machine elements and their respective properties. It results in continuous, integer, or binary data of varying size and structure. Data sources are a combination of existing products, experiments, and simulations with a varying level of detail. Machine learning and especially supervised learning therefore face the challenge of high-dimensional data with limited sample availability. Feature engineering for dimensionality reduction mostly means manual selection incorporating domain knowledge [58, 73]. Sample augmentation is limited to simulation results, for example, properties with FEM calculations [73] with exhaustive searches or topologies with generative search trees [71]. Additionally, Urbas et al. [73] show the application of ensemble learning, i.e. by applying random forest regression for property prediction. Because data are sparse, supervised learning seems challenging especially for topology discovery and therefore rule-based generation is an option [29, 46, 71, 72].

- Model complexity and accuracy

Common approaches in gear design mostly rely on standard calculations, i.e. according to ISO 6336 [63], which include property predictions based on (non)linear regression and decision trees. It is therefore an established interpretable and easy-to-use approach. More complex models such as artificial neural networks [73] are part of research and show promising accuracy for the high-dimensional data in gear design. It also applies to gear transmission discovery methods such as GAP (FVA), which are potentially interpretable, because they rely on decision trees with incorporated domain knowledge. However, flexible designs for diverse applications require more complex models such as artificial neural networks or generative machine learning models in general. But again, interpretability and ease-of-use are challenges for acceptance amongst engineers.

- Knowledge source

The lack of large sample sizes, various property dependencies and diverse structures of gear transmissions makes the use of domain knowledge in current design almost indispensable. Domain knowledge is incorporated manually in the problem and rule definition [44, 58, 71, 72]. Combining data-driven machine learning with domain knowledge directly, as for example reviewed in [103] for deep neural networks and shown by Liu et al. in feature selection for materials property prediction [104], is promising for flexible and high-quality designs, but is rarely addressed in current research of gear transmission design.

4.5 Data collection

Data collection is a crucial step in applying machine learning models successfully. We will briefly outline the common practices in gear transmission development today and refer to other works for general insights, i.e. [105]. Property predictions of gear transmission mostly rely on supervised learning such as (non)linear regressions [63], but there is also research with support vector machine or neural networks [73]. Data originate from experiments and simulations. Experiments include mainly component testing of diverse gears as well as system evaluations. Simulations such as FEM or MBS are a commonly used for data generation or experimental data augmentation. However, data collection is a tedious task in gear design, especially for systems or lager machines. Therefore, new gear transmission discovery often relies on rule-based generation [44, 58, 71, 72] with incorporated expert knowledge and physics. Digitalizing gear expert knowledge (design rules and principles) is key for such applications and requires common standards and guidelines.

5 Discussion and research focus

Grammars, KBE, and CP all still have disadvantages for the use with machine learning methods in automated gear transmission design. Grammars with machine learning have proven that they are able to reduce the subjective influence by experts' knowledge and reuse the knowledge. Examples in transmission development are currently limited to simpler designs, for example, few stage gearboxes with only gears and shafts, but applied successfully. The mutual dependency of gear transmission properties on various design parameters requires holistic approaches for more complex transmission designs. In contrast to today's often strictly iterative and sequential design process, an overall exchange of all analysis and synthesis methods is necessary for optimal design solutions. Grammars and in particular a combination of graph and shape grammar are promising methods to handle flexible topologies. The effort of implementation is not too excessive and especially the combination with deep learning methods such as reinforcement learning and GNNs offers a variety of new options for further exploring the solution space automatically in less time. Because such machine learning algorithms require very many samples and therefore analysis cycles, common analysis tools might take too much time and thus be the bottleneck for model training. Model simplification or value approximation with machine learning methods seems to be a viable approach if necessary. Next to topology search and combinatorial optimization, a transmission also requires spatial dimensioning. Inverse problem-solving algorithms as classic optimization algorithms are common, but are mainly very time consuming. Machine learning algorithms for inverse problem-solving offer new possibilities.

All in all, the gear transmission development still offers high potential for automation. The identified key areas of research are as follows:

- Reinforcement learning methods in combination with GNNs for topology generation of transmissions systems and shaft arrangements with grammars
- Machine learning methods for automatic grammar rule generation
- Transmission graph topology analysis with GNNs
- Common standards and guidelines for expert knowledge digitalization
- Speed-up of transmission analysis tools utilizing machine learning value predictions
- Spatial component dimensioning with methods of machine learning for inverse problems

6 Conclusion

Automated gear transmission design enhanced by machine learning methods is a challenge today. Therefore, we reviewed the current state-of-the-art and evaluated the potentials for the application of graph and shape grammars, KBE, and CP as a basis for machine learning methods. The work contains an overview of available software, to which extend it supports the transmission design process, and how grammars, KBE, and CP are used in transmission and product development in general. A review of optimization and machine learning approaches applied in gear design and relevant advances in the field of machine learning provides insight in current procedures and promising applications. The current state of gear transmission development serves as a basis for potential and challenge evaluation.

7 Nomenclature

The nomenclature is shown in Table 2.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest S. Sendlbeck, M. Maurer, M. Otto and K. Stahl declare that they have no competing interests.

Table 2 Nomenclature

AI	Artificial intelligence
CAD	Computer-aided design
CEE	Computational evolutionary embryogenesis
CP	Constraint programming
FEM	Finite element method
FVA	Research Association for Drive Technology
GNN	Graph neural network
KBE	Knowledge-based engineering
MBS	Multi-body simulation
NRAM	Network-based rule analysis method
NVH	Noise, vibration, harshness
PPO	Proximal policy optimization
RNN	Recurrent neural network

References

1. American Gear Manufacturers Association (2022) gear technology Jan/Feb 2022
2. Ortner M, Schörghuber C, Scheidel S, Hasenbichler G (2018) Selektion der optimalen Antriebsstrangkonfiguration für künftige Anforderungen an Nutzfahrzeuge. MTZ Motortech Z 79(10):30–37. https://doi.org/10.1007/s35146-018-0083-x
3. FVA Antriebstechnik Software. https://www.fva-service.de/de/software/. Accessed 7 Sept 2022
4. GWJ Technology GmbH (2020) User manual gearengineer software
5. Kahraman A Load distribution program. https://mae.osu.edu/gearlab/research. Accessed 3 Jan 2022
6. Ansol Transmission3D. http://ansol.us/Products/TX3/. Accessed 3 Jan 2022
7. FVA GmbH (2022) FVA Workbench v7.1: Modulbeschreibung
8. MDESIGN Vertriebs GmbH MDESIGN LVR und LVRplanet – Lastverteilung an Verzahnungen. https://www.mdesign.de/de/produkte/mdesign-lvr-lvrplanet/. Accessed 17 Apr 2021
9. Höhn B-R, Wirth C, Haefke N (2011) Design and optimization of automotive transmissions with the FVA-Workbench
10. SMT MASTA overview. https://www.smartmt.com/cae-software/masta/overview/. Accessed 7 Sept 2022
11. Romax software. https://romaxtech.com/software/. Accessed 7 Sept 2022
12. KISSsoft. https://www.kisssoft.com/en. Accessed 7 Sept 2022
13. Parlow JC, Otto M, Stahl K (2014) Vom Getriebeentwurf zur Getriebeauslegung. Konstruktion 6:73–79
14. Parlow J (2016) Entwicklung einer Methode zum anforderungsgerechten Entwurf von Stirnradgetrieben (Dissertation, Technische Universität München, Münche)
15. Bansemir G (2013) Konstruktionsleitsystem für den durchgängig rechnerbasierten Zahnradgetriebeentwurf (Dissertation, TU München, München)
16. Dyla A (2002) Modell einer durchgängig rechnerbasierten Produktentwicklung (Dissertation, Technische Universität München)
17. Jaros M (2006) Integration des STEP-Produktmodells in den Getriebeentwicklungsprozess (Dissertation, Technische Universität München)
18. Beulshausen J, Geiger J, Pischinger S, Höhn B-R (2013) Energieeffizienter Antriebsstrang durch Reibungsminimierung. ATZ Automobiltch Z 115(10):828–835. https://doi.org/10.1007/s35148-013-0286-x
19. SMT (2018) MASTA—Empowering transmission engineers to reach new levels of quality when developing
20. Roberts S (2005) Robust transmission design through automated optimization of virtual prototypes. Gear Technol. https://doi.org/10.1002/sia.740010312
21. Kissling U (2011) Optimierungsprozedur zum Auslegen von Stirnradgetrieben nach Gewicht, Kosten und Wirkungsgrad. Konstruktion 3
22. KISSSoft AG (2019) KISSsoft—Release 2019: Benutzerhandbuch
23. Cagan J, Campbell MI, Finger S, Tomiyama T (2005) A framework for computational design synthesis: model and applications. J Comput Inf Sci Eng 5(3):171–181. https://doi.org/10.1115/1.2013289
24. Merriam Webster Definition of grammar. https://www.merriam-webster.com/dictionary/grammar. Accessed 15 Sept 2021
25. Chomsky N (1957) Syntactic structures. Language 33(3):375. https://doi.org/10.2307/411160
26. Gips J, Stiny G (1980) Production systems and grammars: a uniform characterization. Environ Plann B Plann Des 7(4):399–408. https://doi.org/10.1068/b070399
27. Kirshnamurti R, Stouffs R (1993) Spatial grammars: motivation, comparison, and new results. In: Flemming U (ed) CAAD futures '93: Proceedings of the fifth International Conference on Computer-aided Architectural Design Futures, Pittsburgh, PA, USA, 7–10 July, 1993, pp 57–75
28. Chakrabarti A et al (2011) Computer-based design synthesis research: an overview. J Comput Inf Sci Eng. https://doi.org/10.1115/1.3593409
29. Königseder C, Shea K (2016) Comparing strategies for topologic and parametric rule application in automated computational design synthesis. J Mech Des N Y. https://doi.org/10.1115/1.4031714
30. Hamilton WL (2020) Graph representation learning. Synth Lect Artif Intell Mach Learn 14(3):1–159
31. You J, Liu B, Ying R, Pande V, Leskovec J (2018) Graph convolutional policy network for goal-directed molecular graph generation (https://arxiv.org/pdf/1806.02473)
32. Kirchner E (2007) Leistungsübertragung in Fahrzeuggetrieben: Grundlagen der Auslegung, Entwicklung und Validierung von Fahrzeuggetrieben und deren Komponenten. Springer, Berlin, Heidelberg
33. Fu Z, de Pennington A, Saia A (1993) A graph grammar approach to feature representation and transformation. Int J Comput Integr Manuf 6(1–2):137–151. https://doi.org/10.1080/09511929308944564
34. Schaefer J, Rudolph S (2005) Satellite design by design grammars. Aerosp Sci Technol 9(1):81–91. https://doi.org/10.1016/j.ast.2004.08.003
35. Kurtoglu T, Campbell MI (2009) Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. J Eng Des 20(1):83–104. https://doi.org/10.1080/09544820701546165
36. Schmidt LC, Shetty H, Chase SC (2000) A graph grammar approach for structure synthesis of mechanisms. J Mech Des. https://doi.org/10.1115/DETC98/DTM-5668
37. Stiny G, Gips J (1971) Shape grammars and the generative specification of painting and sculpture. IFIP Congress 2:1443–1447
38. Cagan J (2001) Engineering shape grammars: where we have been and where we are going. In: Antonsson EK (ed) Formal engineering design synthesis. Cambridge University Press, Cambridge, pp 65–92
39. Wells AB (1994) Grammars for engineering design. California Institute of Technology
40. Agarwal M, Cagan J (1998) A blend of different tastes: the language of coffeemakers. Environ Plann B Plann Des 25(2):205–226. https://doi.org/10.1068/b250205
41. Lee HC, Tang MX (2004) Evolutionary shape grammars for product design. The seventh international conference generation art (http://www.generativeart.com/on/cic/papersga2004/32.htm)
42. Orsborn S, Cagan J, Pawlicki R, Smith RC (2006) Creating cross-over vehicles: defining and combining vehicle classes using shape grammars. AIEDAM 20(3):217–246. https://doi.org/10.1017/S0890060406060185
43. Heisserman J, Mattikalli R, Callahan S (2004) A grammatical approach to design generation and its application to aircraft systems. Proceedings of Generative CAD Systems Symposium '04
44. Y.-s. Lin Shea K, Johnson A, Coultate J, Pears J (2010) A method and software tool for automated gearbox synthesis. In: Proceedings of the ASME International Design Engineering Technical Con-

ferences and Computers and Information in Engineering Conference—2009, pp 111–121
45. Garcia S (2016) Classifications of shape grammars. In: Gero JS (ed) Design computing and cognition '16. Springer, Cham, pp 229–248
46. Königseder C, Stanković T, Shea K (2016) Improving design grammar development and application through network-based analysis of transition graphs. Des Sci. https://doi.org/10.1017/dsj.2016.5
47. Müller HW (1998) Die Umlaufgetriebe: Auslegung und vielseitige Anwendungen, 2nd edn. Springer, Berlin
48. Chapman CB, Pinfold M (2001) The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. Adv Eng Softw 32(12):903–912. https://doi.org/10.1016/S0965-9978(01)00041-2
49. Verhagen WJC, Bermell-Garcia P, van Dijk REC, Curran R (2012) A critical review of knowledge-based engineering: an identification of research challenges. Adv Eng Informatics 26(1):5–15. https://doi.org/10.1016/j.aei.2011.06.004
50. La Rocca G (2012) Knowledge based engineering: between AI and CAD. Review of a language based technology to support engineering design. Adv Eng Informatics 26(2):159–179. https://doi.org/10.1016/j.aei.2012.02.002
51. Quintana-Amate S, Bermell-Garcia P, Tiwari A (2015) Transforming expertise into knowledge-based engineering tools: a survey of knowledge sourcing in the context of engineering design. Knowl Based Syst 84:89–97. https://doi.org/10.1016/j.knosys.2015.04.002
52. Corallo A, Margherita A, Pascali G, Turrisi G (2011) Optimising the design of gearshafts with Knowledge-Based Engineering (KBE): an application to aircraft engines. IJPD 13(3):185. https://doi.org/10.1504/IJPD.2011.040266
53. Cui J, Wang D (2013) Application of knowledge-based engineering in ship structural design and optimization. Ocean Eng 72:124–139. https://doi.org/10.1016/j.oceaneng.2013.06.013
54. Berx K, Gadeyne K, Dhadamus M, Pipeleers G, Pinte G (2014) Model-based gearbox synthesis. In: Mechatronics Forum International Conference
55. Fauroux JC, Sartor M, Paredes M (2000) Using the skeleton method to define a preliminary geometrical model for three-dimensional speed reducers. EWC 16(2):117–130. https://doi.org/10.1007/PL00007189
56. Stangl M (2007) Methodik zur kinematischen und kinetischen Berechnung mehrwelliger Planeten-Koppelgetriebe (Dissertation, Technische Universität München)
57. Kurth F (2012) Efficiency determination and synthesis of complex-compound planetary gear transmissions, 1st edn. Dr. Hut, München
58. Fürst M, Götz J, Otto M, Stahl K (2021) Automation of gearbox design. Forsch Ingenieurwes. https://doi.org/10.1007/s10010-021-00517-3
59. Holder K, Rudolph S, Stetter R, Salander C (2019) Automated requirements-driven design synthesis of gearboxes with graph-based design languages using state of the art tools. Forsch Ingenieurwes 83(3):655–668. https://doi.org/10.1007/s10010-019-00322-z
60. Brahim M (2019) Optimum design of a spur gear using a two level optimization approach. mech 25(4):304–312. https://doi.org/10.5755/j01.mech.25.4.18994
61. AGMA 2101-D04:2010-01 (2010) Fundamental rating factors and calculation methods for involute spur and helical gear teeth
62. Rai P, Barman AG (2019) An approach for design optimization of helical gear pair with balanced specific sliding and modified tooth profile. Struct Multidisc Optim 60(1):331–341. https://doi.org/10.1007/s00158-019-02198-7
63. ISO 6336-1:2019-11 (2019) Calculation of load capacity of spur and helical gears—Part 1: basic principles, introduction and general influence factors
64. Simon V (2017) Optimization of gear design and manufacture. dtetr. https://doi.org/10.12783/dtetr/icmme2017/9140
65. Kohn B, Utakapan T, Fromberger M, Otto M, Stahl K (2017) Flank modifications for optimal excitation behaviour. Forsch Ingenieurwes 81(2–3):65–71. https://doi.org/10.1007/s10010-017-0218-0
66. Younes EB, Changenet C, Bruyère J, Rigaud E, Perret-Liaudet J (2021) Multi-objective optimization of gear unit design to improve efficiency and transmission error
67. Maputi ES, Arora R (2019) Design optimization of a three-stage transmission using advanced optimization techniques. Int J Simul Multidisci Des Optim 10:A8. https://doi.org/10.1051/smdo/2019009
68. Maputi ES, Arora R (2019) Multi-objective spur gear design using teaching learning-based optimization and decision-making techniques. Cogent Eng. https://doi.org/10.1080/23311916.2019.1665396
69. Sabarinath P, Thansekhar MR, Jeganathan R, Saravanan R (2015) Multi objective design optimization of helical gear pair using adaptive parameter harmony search algorithm. AMM 813–814:1032–1036. https://doi.org/10.4028/www.scientific.net/AMM.813-814.1032
70. Artoni A (2019) A methodology for simulation-based, multiobjective gear design optimization. Mech Mach Theory 133:95–111. https://doi.org/10.1016/j.mechmachtheory.2018.11.013
71. Dumouchel P-E, Masfaraud S (2018) Electric gearbox optimization with artificial intelligence. In: CTI Symposium Berlin
72. Masfaraud S, Danes F, Dumouchel P-E, de Vuyst F, Vayatis N (2016) Automatized gearbox architecture design exploration by exhaustive graph generation. In: World Congress on Computational Mechanics (WCCM)
73. Urbas U, Zorko D, Vukašinović N (2021) Machine learning based nominal root stress calculation model for gears with a progressive curved path of contact. Mech Mach Theory 165:104430. https://doi.org/10.1016/j.mechmachtheory.2021.104430
74. Zhou J et al (2018) Graph neural networks: a review of methods and applications. arXivLabs. https://doi.org/10.48550/arXiv.1812.08434
75. Zhou J et al (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001
76. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge (https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6260249)
77. Choi J, Hyun M, Kwak N (2019) Task-oriented design through deep reinforcement learning. arXivLabs. https://doi.org/10.48550/arXiv.1903.05271
78. Hayashi K, Ohsaki M (2021) Reinforcement learning for optimum design of a plane frame under static loads. EWC 37(3):1999–2011. https://doi.org/10.1007/s00366-019-00926-7
79. Jang S, Yoo S, Kang N (2022) Generative design by reinforcement learning: enhancing the diversity of topology optimization designs. Comput Aided Des 146:103225. https://doi.org/10.1016/j.cad.2022.103225
80. Ruiz-Montiel M, Boned J, Gavilanes J, Jiménez E, Mandow L, Pérez-de-la-Cruz J-L (2013) Design with shape grammars and reinforcement learning. Adv Eng Informa 27(2):230–245. https://doi.org/10.1016/j.aei.2012.12.004
81. Mirhoseini A et al (2020) Chip placement with deep reinforcement learning. arXivLabs. https://doi.org/10.48550/arXiv.2004.10746
82. Ardizzone L et al (2018) Analyzing inverse problems with invertible neural networks. arXivLabs. https://doi.org/10.48550/arXiv.1808.04730
83. Holl P, Koltun V, Thuerey N (2021) Physical gradients for deep learning. arXivLabs. https://doi.org/10.48550/arXiv.2109.15048
84. VDI/VDE 2206 (2021) Development of mechatronic and cyber-physical systems
85. VDI 2221 (2019) Entwicklung technischer Produkte und Systeme – Modell der Produktentwicklung. VDI, Berlin

86. Mattson CA, Sorensen CD (2020) Product development: principles and tools for creating desirable and transferable designs. Springer, Cham
87. Berx K, Pinte G, Gadeyne K (2013) Model-based optimal architectural design of mechatronic systems: application to gearbox synthesis. In: Proceedings of the 2nd Workshop on "Mechatronic Design, pp 1–5
88. Niemann G, Winter H, Höhn B-R, Stahl K (2019) Konstruktion und Berechnung von Verbindungen, Lagern, Wellen. Maschinenelemente, vol 1. Springer
89. Genender P, Wolff K, Govindswamy K, Hueser M (2002) NVH-Aspekte der Integration des Antriebsstrangs in das Fahrzeug. MTZ Motortech Z 63(6):470–477. https://doi.org/10.1007/bf03226633
90. Weis P, Kučera Ľ, Pecháč P, Močilan M (2017) Modal analysis of gearbox housing with applied load. Procedia Eng 192:953–958. https://doi.org/10.1016/j.proeng.2017.06.164
91. Kieninger D, Hemsen J, Köller S, Uerlich R (2019) Automatisierte Auslegung und Optimierung von Getrieben für Elektrofahrzeuge. MTZ Motortech Z 80(11):94–99. https://doi.org/10.1007/s35146-019-0129-8
92. Helms B (2012) Object-oriented graph grammarsfor computational design synthesis. Lehrstuhl für Produktentwicklung, Technische Universität München, München
93. Kerzhner AA (2009) Using domain specific languages to capture design synthesis knowledge for model-based systems engineering. Georgia Institute of Technolog, Atlanta
94. Stump GM, Miller SW, Yukish MA, Simpson TW, Tucker C (2019) Spatial grammar-based recurrent neural network for design form and behavior optimization. J Mech Des. https://doi.org/10.1115/1.4044398
95. Orsborn S, Cagan J, Boatwright P (2008) A methodology for creating a statistically derived shape grammar composed of non-obvious shape chunks. Res Eng Design 18(4):181–196. https://doi.org/10.1007/s00163-007-0035-9
96. Yogev O, Shapiro AA, Antonsson EK (2010) Computational evolutionary embryogeny. IEEE Trans Evol Comput 14(2):301–325. https://doi.org/10.1109/TEVC.2009.2030438
97. Orsborn S, Boatwright P, Cagan J (2008) Identifying product shape relationships using principal component analysis. Res Eng Design 18(4):163–180. https://doi.org/10.1007/s00163-007-0036-8
98. Sainter P, Oldham K, Larkin A (2000) Achieving benefits from knowledge-based engineering systems in the longer term as well as in the short term. Proceedings of the 6th International Conference on Concurrent Enterprising, June 2000
99. O'Sullivan B (2010) Automated modelling and solving in constraint programming. Proc Conf AAAI Artif Intell 24(1):1493–1497
100. Cappart Q, Moisan T, Rousseau L-M, Prémont-Schwarz I, Cire A (2020) Combining reinforcement learning and constraint programming for combinatorial optimization. arXivLabs. https://doi.org/10.48550/arXiv.2006.01610
101. Liu Y, Zhao T, Ju W, Shi S (2017) Materials discovery and design using machine learning. J Materiomics 3(3):159–177. https://doi.org/10.1016/j.jmat.2017.08.002
102. Liu Y, Guo B, Zou X, Li Y, Shi S (2020) Machine learning assisted materials design and discovery for rechargeable batteries. Energy Storage Mater 31:434–450. https://doi.org/10.1016/j.ensm.2020.06.033
103. Dash T, Chitlangia S, Ahuja A, Srinivasan A (2022) A review of some techniques for inclusion of domain-knowledge into deep neural networks. Sci Rep 12(1):1040. https://doi.org/10.1038/s41598-021-04590-0
104. Liu Y, Wu J-M, Avdeev M, Shi S-Q (2020) Multi-layer feature selection incorporating weighted score-based expert knowledge toward modeling materials with targeted properties. Adv Theory Simul 3(2):1900215. https://doi.org/10.1002/adts.201900215
105. Roh Y, Heo G, Whang SE (2021) A survey on data collection for machine learning: a big data—AI integration perspective. IEEE Trans Knowl Data Eng 33(4):1328–1347. https://doi.org/10.1109/TKDE.2019.2946162