INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

VIRTUAL SYNTHESISERS FOR MECHANICAL GEAR SYSTEMS

Alex C Starling and Kristina Shea

Keywords: virtual product development, virtual design synthesis, grammatical design, functional modelling, simulation-based design, multi-objective design optimisation

1 Introduction

This paper presents a grammatical method for parametric synthesis of mechanical gear systems. The goal of this research is to enhance innovative design, as discussed by Dym [1], through the development of new computational synthesis tools. Aiding conceptual and embodiment design has been identified as a key, yet difficult, research goal in advanced computer-aided design systems for mechanical engineering. While most virtual product development tools are targeted at supporting later design stages, e.g. detail design, the benefits of supporting earlier stages are potentially greater as it is generally considered that most "added value" is contributed to a product earlier in its development. However, in these earlier design stages, synthesis tasks can be difficult to formulate computationally.

Synthesis research, in the form of computational production systems, can be traced back to the advent of the computer [2]. Synthesis is a fundamental task of engineering and can be thought of as creating form to fulfil desired behaviour and function. Existing synthesis research can be categorised into computational and non-computational, i.e. "paper-based", methods.

From a computational viewpoint, Raphael and Smith [3] describe synthesis as being the antithesis of analysis, "the reverse of analysis, where target behaviour is used to infer a physical configuration within an environment". Antonsson and Cagan [4] provide a more abstract definition of engineering design synthesis, describing it as "the creative step itself: the conception and postulation of possibly new solutions to solve a problem". Design synthesis is commonly considered a manual task: "In most engineering design, this step is performed by creative human minds" [4]. Further, they put forward that engineering design synthesis can be referred to as being *formal* engineering design synthesis when it is "computable, structured and rigorous, not *ad hoc*" [4]. The aim of the research presented here is to develop formal, computable methods for virtual, simulation-driven synthesis to aid designers in developing better products through rapid generation of a wide range of alternative mechanical solutions.

As yet, computational methods and tools have influenced our ability to model and analyse potential designs, but they have not contributed as greatly to design *synthesis* tasks. Despite the proliferation of computer tools for all aspects of engineering design, it seems that the task of synthesising new ideas and creating innovative and original concepts has remained almost the exclusive burden of human designers. Commercial tools offering limited computational assistance can provide some design assistance, e.g. TRIZ[1] and the Invention Machine[2]. However, mapping design specifications to tangible embodiments remains mainly a manual task. The core question addressed in this paper is whether computational methods can be developed that stand alongside an engineer as a "virtual synthesiser" to improve rapid generation of high quality design alternatives that can be further developed into innovative designs.

To validate the simulation-driven synthesis method that is presented here, a case study on the configuration of automotive gearbox designs is considered. It is thought that current vehicle transmission technology requiring gearboxes will be used until at least the year 2020 [5]. Computer modelling and simulation in this design domain has been increasing in recent years and the incorporation of new capabilities for computational synthesis is considered the next step towards shorter gearbox development and implementation cycles [5].

In Starling and Shea [6,7,8] the authors introduce a parallel grammar, implemented for a mechanical clock design case study, as a basis for mechanical synthesis of gear systems, including, but not limited to, linear gear trains. This work is now extended to include simulation-based evaluation of designs, in addition to previous geometry-based evaluation and multi-objective search to enable generation of Pareto optimal sets of design alternatives. First, a general overview of the method is presented illustrated by example of a camera winding mechanism. Second, applicability of the method to beneficial industrial problems is investigated through an automotive gearbox redesign problem, which serves to validate the method.

2 Background

Chomsky [9] was the first to use the word "grammar" in the technical context of production systems [2] while developing string grammars to generate valid, i.e. "grammatically correct", linguistic sentences. Other grammars have since been developed, e.g. shape grammars [10]. The use of grammars to assist design is conceptually simple. In the same way as a natural language is based on rules (termed a grammar), it is also possible to develop a language of designs via design grammars. Starting with a legal construct, repeated application of different grammar rules generates new designs. The sum of different designs produced by exhaustive application of all valid sequences of grammar rules to a starting symbol is termed the design language. Early spatial design work using shape grammars can be found in the field of architecture and visual arts [11].

Development of shape grammars for generation of mechanical designs has been carried out for diverse products such as Harley-Davidson motorcycles [12], coffeemakers [13] and MEMS devices [14]. Earl [15] discusses the use of shape grammars for generation of form as well as function. McCormack and Cagan [16] have developed a two-dimensional parametric shape grammar interpreter and demonstrated its use for the design of vehicle inner hood panels. While the previous methods have focused on shape generation, Finger and Rinderle [17,18] have developed a grammar based on the manual manipulation of form-behaviour

---

[1] http://www.triz-journal.com/ (last accessed 27 May 2005)

[2] http://www.invention-machine.com/ (last accessed 27 May 2005)

diagrams using bond graphs [19] for conceptual design of topological configurations from a part-based element library. Schmidt et al [20] present a graph-grammatical approach to synthesis of mechanisms with less of an emphasis on quantitative performance evaluation as the work in this paper.

3 Method

A framework for design synthesis research and methods (Figure 1) has been proposed [6] based on a review of general approaches [4] and previous, extensive work in the area of structural synthesis, e.g. [21].

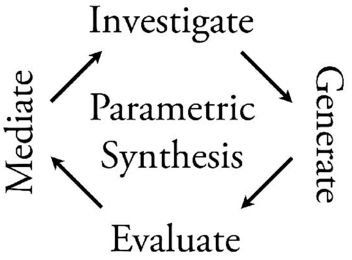


Figure 1: Parametric synthesis framework

The phases of the parametric synthesis framework provide a structure for design synthesis research. First, choose and *investigate* a particular design domain, e.g. mechanical gear systems. Second, create a production system capable of *generating* existing and novel designs within the design domain, called a “design language”. Third, create automatic mechanisms for *evaluating* the performance of designs to interpret and quantify the “goodness” of designs within the language defined. Fourth, *mediate*, i.e. reason about the evaluated design language so as to explore, compare and choose among alternatives. In practice, there can be overlap between the four phases of this parametric synthesis methodology [22]. The following sections provide an overview of the method created and implemented for gear systems. A detailed discussion of the method is outside the scope of this paper and can be found in [8].

3.1 Design Generation and Classification

The parallel grammar for mechanical gear systems consists of both function and structure representations where constraints ensure that the designs generated are both topologically and parametrically valid. Function is represented by a function graph where nodes represent spindles or axles and links between nodes represent gear pairs. Each function graph is translated into an embodiment consisting of a set of components, resulting in a parallel representation of function and form, shown in Figure 2.

Given a design specification, first an initial design is created using the grammar create rules or “C-rules” [7]. It is then modified using perturb rules, or “P-rules” [7], enabling design modification that maintains the integrity of both form and function, ensuring that no design constraints are violated. The design synthesis framework for design generation using a parallel grammar [6,7] is shown in Figure 3.

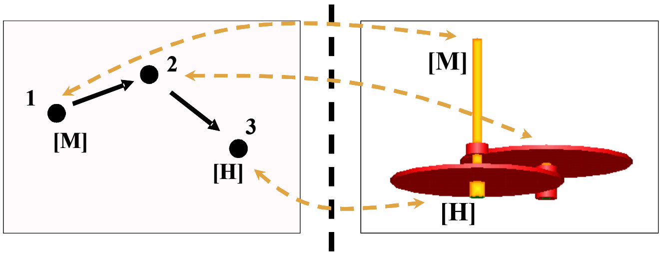


Figure 2: A parallel representation of function (left) and form (right)

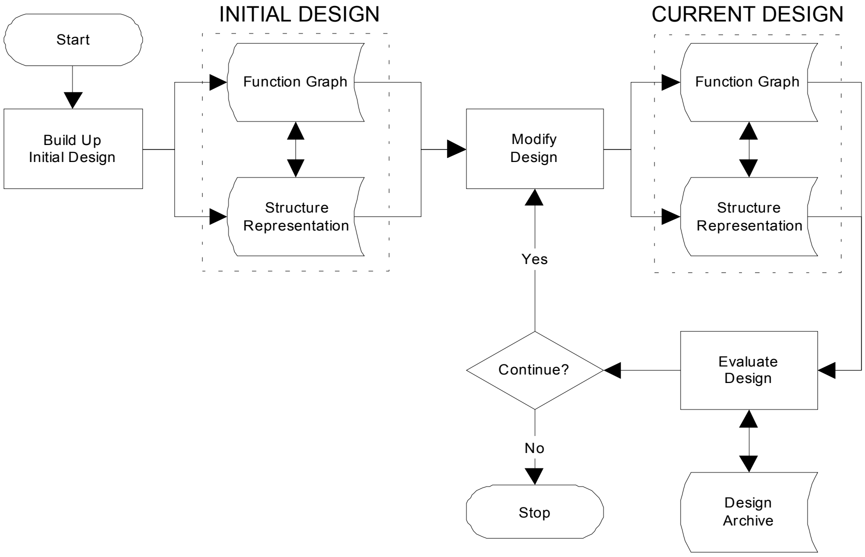


Figure 3: Generation of design alternatives

For a production system, or grammar, capable of generating designs for a particular domain, it is worthwhile investigating classification of designs within a design language. To what extent can two designs be considered "different" or "similar" and how can these differences be described quantitatively? Figure 4 depicts a categorisation of designs within a language into "clans" and "families" [7].

Designs that vary only in parameter values can be considered to be in the same family, i.e. they vary parametrically. Two designs can be considered to be part of different design families if they vary in component topology, i.e. architecture details. As an example, consider a 5-speed manual gearbox for a passenger vehicle. Adjusting gear ratios only, to provide different levels of torque, is an example of creating a new design within the same family. Varying the number of speeds provided by the gearbox, i.e. changing component topology from a 5- to a 6-speed gearbox design, is an example of creating a new family of designs. At a higher level of abstraction, clans represent distinctly different design concepts and alternatives. Continuing with the gearbox example, all manual gearboxes based on parallel shafts are considered members of the same design clan, whereas a gearbox with epicyclic gear trains[3] would be a member of a different clan. A clan can have many family sub-classes that have the same general system topology.

---

[3] Such devices find use in many areas of robotics and positioning systems rather than in the automotive industry: www.harmonicdrive.de (last accessed 21 March 2005).

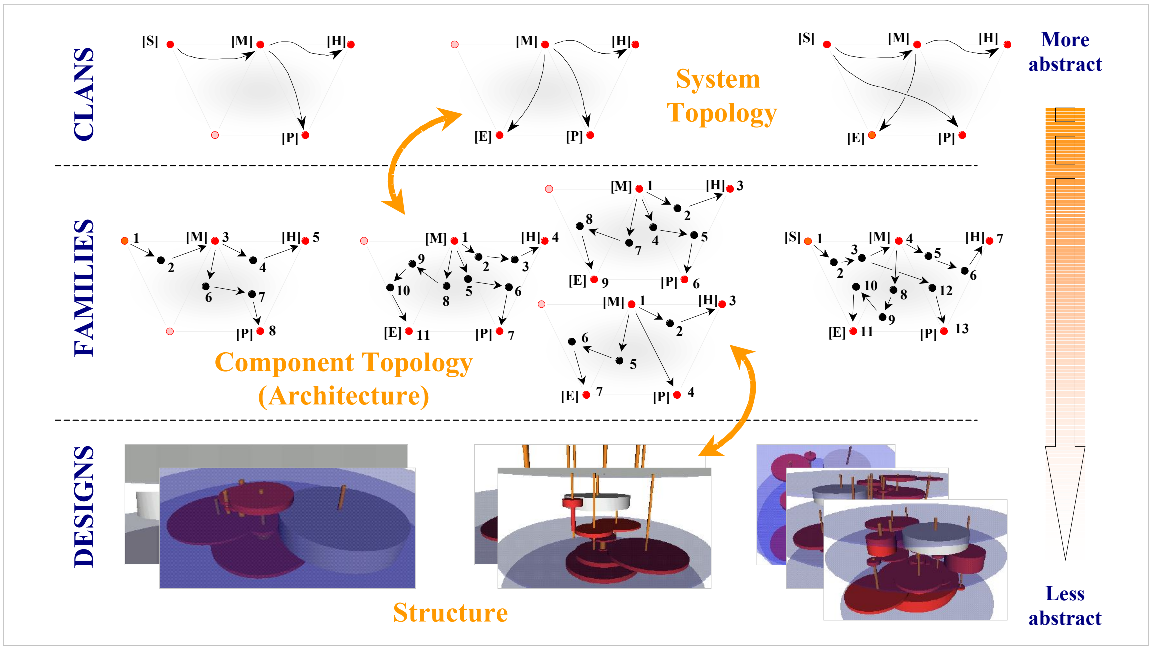


Figure 4: Clans and families of designs annotated with mechanical clock examples [8]

3.2 Simulation and Evaluation

In the case of most mechanical design problems, simple geometry-based performance metrics, such as a mass and volume, are usually not sufficient means of determining the quality of a design. Extension to using simulation as part of design evaluation provides the potential to capture richer and more meaningful performance data. The evaluation phase is carried out by automating the creation of a behavioural model in the native format of a simulation tool. The simulation is run and outputs quantitative information about the behaviour of a design, enabling a decision to be made on whether or not a new design is an improvement on previous designs. The challenges lie in constructing robust simulation models "on the fly".

To illustrate the proof-of-concept for incorporating mechanical simulation within an automated synthesis method, the cross-domain modelling language Modelica[4] is used (Figure 4). Modelica models can be simulated using Dymola[5], a simulation environment that employs Kron's method of 'tearing' [23]. This enables solving multidimensional systems without resorting to linear simplifications to enable the prediction of dynamic behaviour and interaction between components in designs. As an object-oriented modelling system, Dymola is suited to object-based synthesis and a large range of Modelica component libraries are available that cover different engineering domains, including mechanical systems. These basic building blocks can be used to create multi-domain design models, for example mechatronic devices.

---

[4] http://www.modelica.org/ (last accessed 11 November 2004)

[5] http://www.dynasim.se/ (last accessed 30 December 2004)

3.3 Search and Mediation

Based on deterministic pattern search algorithms [24], a prototype hybrid pattern search algorithm [8] is used to reward successful sequences of design modifications to both the function and structure representations towards optimally directed designs. A multi-objective approach allows generation of a palette of non-dominated solutions, i.e. Pareto sets [25], where each design archived is better for at least one performance measure than each other design when compared pairwise.

To illustrate the method described, the design of a simple mechatronic film camera is considered [8,26], in particular the design of a gear train used to transfer power from the motor of the camera to the winding mechanism (Figure 5). The spur gears are constrained by a bounding box (the housing of the camera) yet must transfer power efficiently. The overall gear ratio affects both the power usage of the camera and the winding time taken between film exposures, or frames. Ideally, the camera should have both low power consumption as well as fast winding. The initial goal of the study is to generate an optimised set of design alternatives that describe key performance trade-offs.

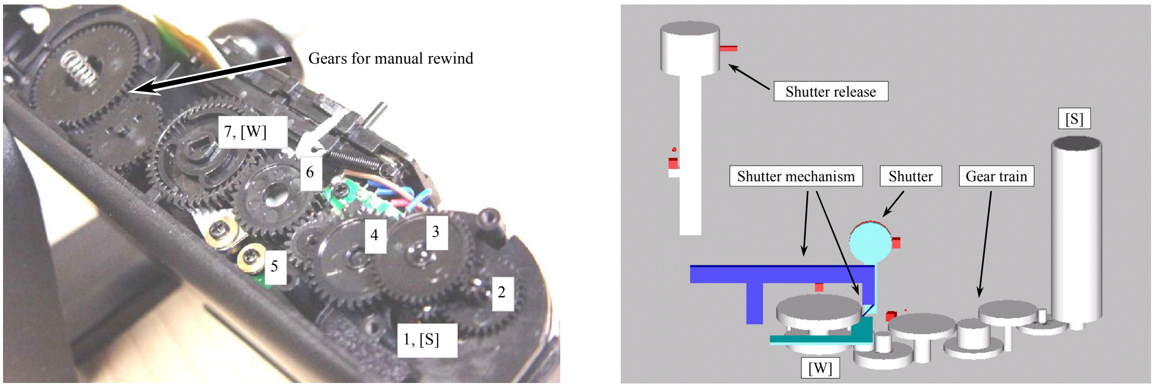


Figure 5: Winding mechanism of camera: actual (left) and Dymola simulation (right)

To investigate this application, the design clan of the winding mechanism is modelled in Modelica. Each different design family within the clan has a different number of nodes in its function graph, where each node represents a gear-carrying spindle in the gear train. A hybrid pattern search method [8] is used to generate camera designs using the parallel grammar with suitable bounding box constraints. Sample results are presented in Figure 5 that compare the trade-off between battery change ($Q_{battery}$) and winding time (t_{stop}) for different design families, referred to as x-noded designs where x is the number of spindles used in the gear train. Other performances considered include simple mass and volume metrics, as well as weighted thickness, aspect ratio and compactness.

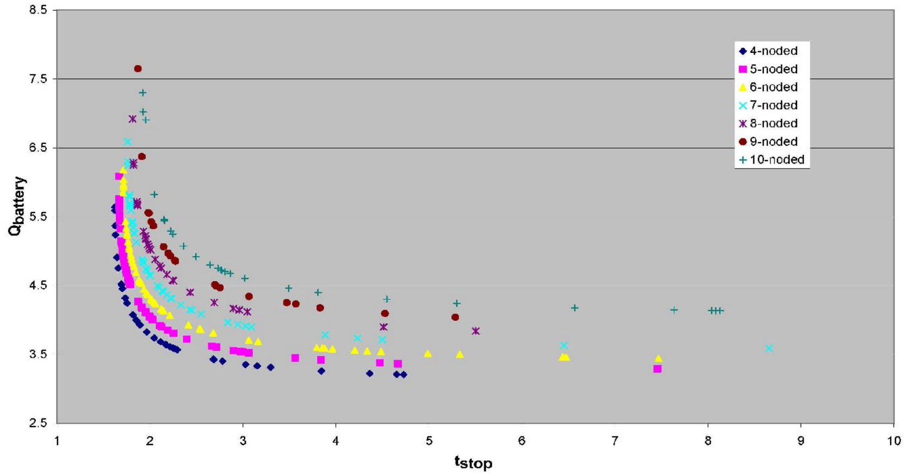


Figure 6: Multi-objective performance trade-offs for winding mechanism: winding battery charge used ($Q_{battery}$) is plotted against winding time (t_{stop})

The strength of combining a parallel grammar capable of generating a wide range of valid winding mechanisms and a multi-objective search approach is rapid generation of optimised design alternatives and their corresponding virtual 3D design models, via VRML[6]. Presenting designers with alternatives organised according to multi-objective trade-offs allows subsequent identification and exploration of beneficial performance regions.

4 Method Validation: Vehicle Gearbox Design

To validate the simulation-driven synthesis method presented, the design of automotive manual gearboxes is now used as an industrial case study. The example of a camera winding mechanism is limited to exploring families of designs within the same clan, i.e. gear trains, varying component topology only. This section will drive the need to expand the method to generating clans, in addition to their families, incorporating clutches and evaluating power flow paths within function graphs.

4.1 Investigating Gearboxes

Internal combustion engines used in vehicles have narrow operating ranges where torque and power are both at optimal levels. Therefore to provide a vehicle, such as a car, with a useful range of speeds, a gearbox is required. A gearbox contains a number of parallel gear trains of differing ratios that can be selected, one at a time, to transfer power from the engine to the driven wheels to suit driving conditions. The selection of these time-dependent connections is governed by user-controlled clutch mechanisms.

Figure 7 shows a simplified sketch of the forward portion of a passenger car. Front wheel drive is commonly used for non-speciality motor vehicles, which in most cases means that the engine is located at the front. Vehicle dynamics dictate that the heavy engine block is centrally mounted, resulting in restricted space on both sides of the engine. Sitting on one side of the

---

[6] Virtual Reality Modelling Language

engine, the gearbox takes a rotational input from the engine, converts this rotation by a given ratio and outputs the new rotation to the differential that then drives the front wheels of the car. The driver of the vehicle selects these ratios by moving a gear stick: it is standard practice to have a choice of five forward gear ratios and one reverse ratio. These are termed ‘speeds’, i.e. the first ratio is the first speed, the second ratio is the second speed, etc. This configuration is known as a ‘5-speed gearbox’.

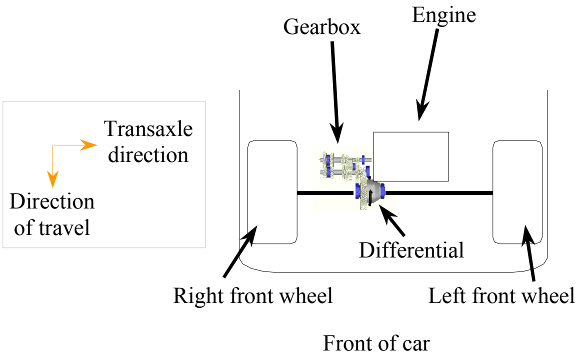


Figure 7: Sketch of front wheel drive passenger car layout, adapted from [27]

As more powerful vehicle engines have become more affordable and popular, there has been a trend for gearboxes to have more speeds. Older cars, such as pre-1980s vehicles, mostly used 4-speed gearboxes. 5-speed gearboxes are now considered standard, while 6-speed gearboxes are also becoming more common [28], providing drivers with more ratios to choose from to better adapt to varying driving conditions. Some high-powered vehicles, e.g. the Bugatti Veyron, have been designed with 7-speed gearboxes.

As the axis of the front wheels is restricted to lie across the vehicle and orthogonal to the direction of travel, it is common practice to align the shafts of the gearbox in the same direction. This layout is termed a transaxle (“across the axle”) gearbox [27]. The differential, required to permit different rotation speeds of the front wheels to allow for steering, is then aligned between the front wheels at the output end of the gearbox. The simplest layout that can be considered for a 5-speed gearbox is shown in Figure 8.

This standard transaxle 5-speed gearbox has three main shafts. The output shaft from the engine is connected to an intermediate shaft by a set of different gear pairs, one for each speed available to the driver. If no speed is selected, the gear disks on the intermediate shaft do not grip the intermediate shaft. Hence the engine can run with the input shaft rotating and all the gear pairs on the intermediate shaft spinning freely. This is termed “neutral” speed. If the driver selects one of the forward speeds, a clutch mechanism activates to connect the intermediate shaft with the relevant gear pair. Hence an input rotation causes the intermediate shaft to rotate, resulting in power being transferred to the differential and thus also to the front wheels of the vehicle. Reverse gear works in a similar manner, except that there is another gear disk on a separate shaft that engages to rotate the wheels of the car backwards.

A main design issue with transaxle gearboxes is the space restriction between the engine and the outside of the vehicle, as the former is constrained to be centrally mounted. The gear pair for fifth speed in the standard layout, shown in Figure 8, is close to the right front wheel of the vehicle. A 6-speed gearbox based on this layout with another gear pair added to the left end of the shafts would be difficult to implement due to the aforementioned space restrictions. Another difficulty is that long shafts flex when loaded, resulting in substandard meshing characteristics and reduced performance. Using a greater number of intermediate shafts for alternative layouts could resolve some of the issues associated with extending the standard layout to 6 speeds.

Several alternative layouts already exist in current 5-speed cars. The layout of a new design, used in the Rover 75, is shown on the RHS of Figure 8. In this image, the differential is shown without its covering. Third, fourth and fifth speed work in the same fashion as in the standard layout: the input shaft drives intermediate shaft A via the relevant gear pair and power is transferred to the differential.

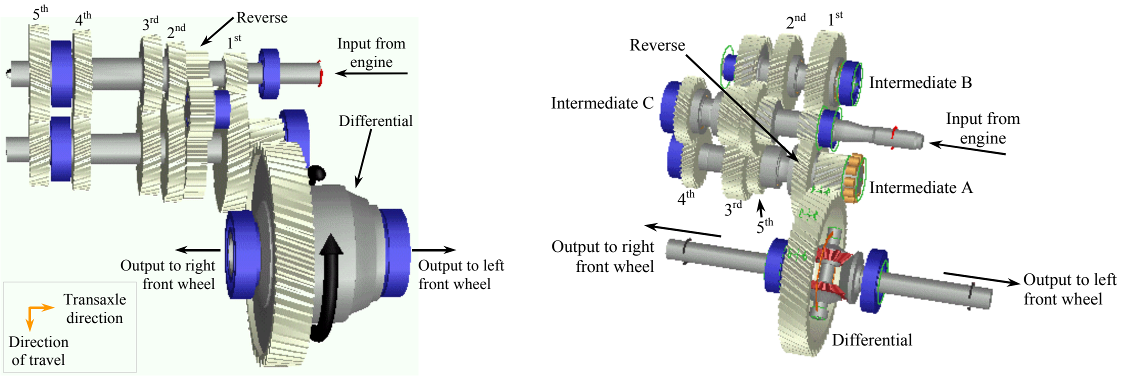


Figure 8: Standard (left) and alternative (right) 5-speed gearbox layout, adapted from [27]

First and second speed function in a different way. The input shaft actually consists of two concentric shafts, the original input axle that connects to the engine and a concentric sleeve (intermediate C). For third, fourth and fifth speed these are locked together and the fused entity acts as a single shaft, passing power from the engine directly to intermediate shaft A and then on to the differential. When first or second speed are selected, the concentric shafts (input and intermediate C) disengage, and power flows between these concentric, non-fused shafts via intermediate shaft B. The third speed gear pair is then used to transfer power to intermediate shaft A.

The differences between the standard and new gearbox layouts are summarised in Table 1. There are situations when the standard gearbox layout is not adequate for the type of car being designed. In these cases, being able to rapidly generate alternative layouts tailored to new design requirements would be beneficial. This would potentially increase the number of options considered by designers and therefore increase the possibility of finding a configuration with improved performance, e.g. decreased cost, in a shorter amount of design time.

The case study presented was carried out in conjunction with a UK-based company that produces gearbox design software and whose core competence is the complete modelling of gear-based systems. Their main product, a design tool, is used by many leading automotive companies to design gearboxes. The software models and simulates gearboxes to a high level of detail and allows detailed optimisation of features such as gear tooth profiles. The representation of detailed models makes it possible to analyse complex phenomena such as shaft misalignment and gear whine, which are important performance characteristics to gearbox manufacturers.

Table 1: Summary of gearbox layout options

Gearbox layout	Advantages	Disadvantages
Standard	• Conceptually simple • Low part count	• Long shafts • Restricted gear ratios due to distance between shafts
New	• More gear ratio options • Requires less space in transaxle direction	• Complex • Higher part count

The aim of this case study is to investigate the feasibility of combining the method presented for gear system synthesis and optimisation with commercial simulation software for to enable automated, simulation-driven generation of new gearbox layouts. A detailed modelling tool complements the higher-level gear system representation used by the parallel grammar. The alternative layout for the 5-speed gearbox introduced in this section is not difficult to visualise, however, other vehicles can require even more complex layouts. For example, on-road heavy-duty trucks and off-road vehicles, such as tractors, feature gearboxes with up to 18 speeds. The design alternatives for such gearboxes are numerous making the case for using a computational synthesis approach to design generation compelling. The alternative 5-speed gearbox in Figure 8 has been successfully used in current vehicles to fit the gearbox into a constricted space, thus solving a major design issue. The growing demand for 6-speed gearboxes, where these layout problems are particularly acute, is a driver for incorporating a computation approach generating and exploring new design clans and families.

4.2 Generating Gearboxes

In order to combine automated synthesis of alternative gearboxes with simulation, two main extensions to the function grammar are required. First, a clutch function is represented by having parallel edges connecting nodes in the function graph. Second, additional rules are required to alter the system topology of graphs. A graph representation of the standard 5-speed gearbox configuration (Figure 8 LHS), not including reverse, is shown in Figure 9. This graph is relatively straightforward, containing five separate paths between vertices 1 and 2, corresponding to the five speeds of the gearbox, and a single connection between nodes 2 and 3, corresponding to the output of the system to the differential.

Figure 9 also shows a representation of the alternative 5-speed gearbox layout (Figure 8 RHS). The label [Z], corresponding to concentricity of node with its parent vertex, represents the dual function of such concentric shafts, i.e. they can be coupled and decoupled using a clutch mechanism as discussed above. The edge between nodes 1 and 4 is dashed to note that it is a speed-dependent connection between concentric shafts in the structure representation, e.g. input shaft and intermediate shaft C in Figure 8 (RHS).

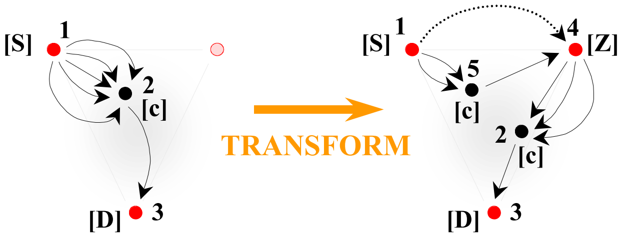


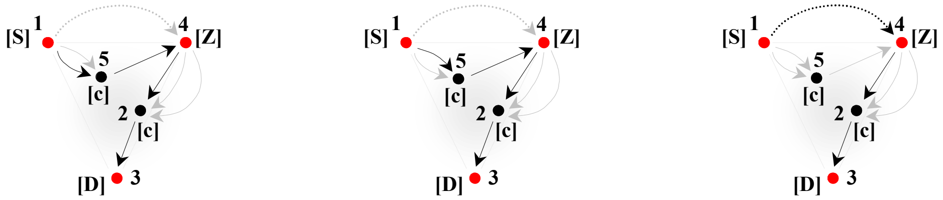
Figure 9: Function graphs for standard (left) and alternative (right) transaxle gearbox layouts

If it were possible to perform a transformation on the standard 5-speed gearbox representation to enable the generation of the function graph of the alternative design representation, similar transformations could be used to generate further design clans that fulfil the same functional specification as the initial standard design in a novel way. This work can be seen as a functional analogy to grammar transformations developed to follow style variations in architectural design [29]. The aim is to capture the language of gearbox designs for vehicle applications. The resulting function graphs from this exploration could then be used as an input for the parallel grammar to search for preferred designs.

Before this transformation can be studied in more detail it is necessary to distinguish between “active” and “inactive” edges in the graphs in Figure 9. In the gearbox example there are more possible combinations of power flow paths, that is paths from engine source [S] to differential [D], than there are speeds. Given a design specification for the number of speeds required, to automatically generate a gearbox it is necessary to have a mechanism for representing power flow paths in order to evaluate the number of speeds and to formulate topological constraints that maintain the validity of the power flow paths when modifying the design at a component level. These paths are included in Figure 10 for the alternative 5-speed gearbox configuration. Active edges, i.e. gear pairs that have been engaged, are coloured black and inactive edges, i.e. gear pairs that are not engaged, are shaded grey. Each separate gearbox speed can therefore be recorded by noting the sequence of edge traversals between input and output. The notation lists the sequence of vertices from input to output for each speed. If an ambiguous path is specified, i.e. there is more than one edge between two particular vertices, the vertex distinguishing identifier is noted in brackets.

To find new gearbox configurations, graph exploration is attempted considering the function graph alone for simplicity. The main aim is to generate design clans and families; parametric variations will not be considered here. An example sequence of modification rule applications is shown in Figure 11 to transform the graph representing the standard 5-speed gearbox into the graph representing the new layout (Figure 9) using the pre-existing function rules [8].

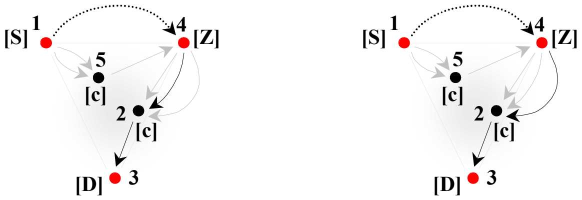
The graph modification rules also allow the creation of further configurations. An example of such a design, generated by applying the grammar rules by hand, is shown in Figure 12 with active edges for each speed shown in Figure 13. This layout has three concentric shafts, nodes 1, 3 and 5. Such a complex layout may well not be viable for a production vehicle, however, such an arrangement could be applicable if a closely packed transmission is required.



First speed
1 – 5(1) – 4 – 2(3) – 3

Second speed
1 – 5(2) – 4 – 2(3) – 3

Third speed
1 – 4 – 2(3) – 3



Fourth speed
1 – 4 – 2(4) – 3

Fifth speed
1 – 4 – 2(5) – 3

Figure 10: Active edges for different speeds in the alternative 5-speed gearbox graph (RHS of Figure 9) with path for each speed

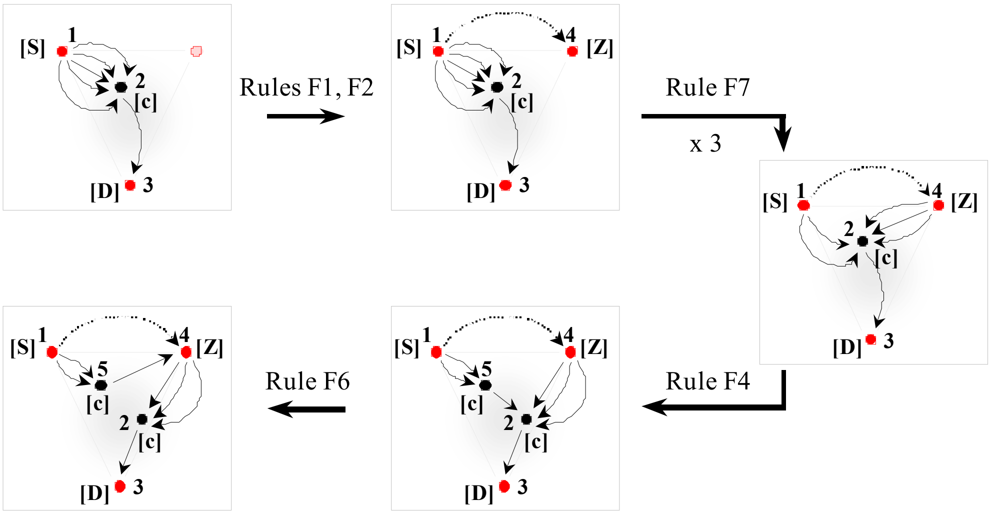


Figure 11: A sequence of function graph transformations used to generate the alternative 5-speed gearbox configuration from the standard layout (c.f. Figure 9).

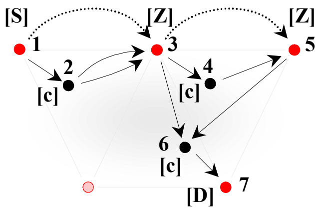
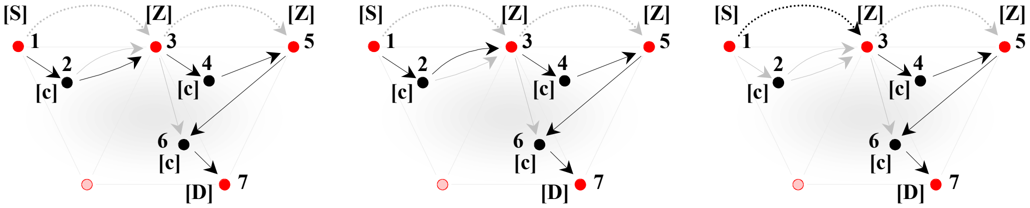


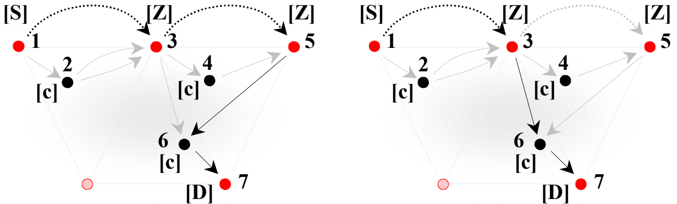
Figure 12: New 5-speed gearbox configuration synthesised with the function grammar



First speed
1 – 2 – 3(1) – 4 – 5 – 6 – 7

Second speed
1 – 2 – 3(2) – 4 – 5 – 6 – 7

Third speed
1 – 3 – 4 – 5 – 6 – 7



Fourth speed
1 – 3 – 5 – 6 – 7

Fifth speed
1 – 3 – 6 – 7

Figure 13: Active edges for different speeds in the novel 5-speed gearbox graph (Figure 13) with node sequence for each speed to indicate power flow

Further work is now required to extend the structure grammar to be able to embody the function graph representations of gearboxes. For example, the current implementation constrains shafts to lie parallel to each other whereas longitudinal gearbox layouts, require shafts to be aligned parallel to the direction of vehicle travel. The inclusion of bevel gears within the structure grammar would enable the analysis of gearboxes based on layouts other than the transaxle designs considered in this paper. Future extensions also include linking the generative method with commercial simulation software, in place of Dymola, that will provide detailed 3D modelling and simulation capabilities. Such extensions are targeted at providing the foundation for a robust, simulation-driven method for virtual synthesis of a wide range of gearboxes.

5 Conclusions

This paper has presented an overview of a simulation-driven grammatical method for generating mechanical gear systems. The method combines a parallel grammar, to create and modify both function and structure representations of designs, with automated simulation and multi-objective optimisation. Application of the method results in the generation of sets of design archives that present a wide range of optimised design alternatives to designers and highlight performance trade-offs. The approach was verified using a camera winding mechanism consisting of a linear gear train. Method validation was demonstrated by an extension to synthesis of vehicle gearboxes, requiring representation of clutches, i.e. time-dependent gear connections, and power flow paths within the grammar. Based on initial results, the possibilities for synthesis of new gearbox configurations is shown to be of relevance and interest to industry. While the work presented considered applications to a camera mechanism and automotive transmissions, the structure grammar can be extended to use a larger library of parts to enable more classes of mechanical designs to be considered.

6 Acknowledgements

This work is supported by the UK EPSRC, the Leverhulme Trust (UK), the Newton Trust (UK) and St John's College, Cambridge, UK. The authors would also like to thank Barry James and Peter Poon for their discussions on this work.

References

[1] Dym C.L., "Engineering Design: A Synthesis of Views", Cambridge University Press, 1994.

[2] Post E., "Formal reductions of the general combinatorial decision problems", American Journal of Mathematics, Vol. 65, 1943, pp.197-268.

[3] Raphael B. and Smith I.F.C., "Computer-Aided Engineering", John Wiley & Sons Ltd, Chichester, 2003.

[4] Antonsson E.K. and Cagan J., "Formal Engineering Design Synthesis", Cambridge University Press, Cambridge, 2001.

[5] Poon S.Y., Romax Technology, Private Communication, 2003.

[6] Starling A.C. and Shea K., "A Clock Grammar: The Use of a Parallel Grammar in Performance-Based Mechanical Design Synthesis", Proceedings of the 2002 ASME International Design Engineering Technical Conferences, Montreal, Canada, 2002.

[7] Starling A.C. and Shea K., "A Grammatical Approach to Computational Generation of Mechanical Clock Designs", Proceedings of the International Conference on Engineering Design, ICED'03, Stockholm, Sweden, 2003.

[8] Starling A.C., "Performance-Based Computational Synthesis of Parametric Mechanical Systems", PhD Thesis, Cambridge University, Cambridge, UK, 2004.

[9] Chomsky N., "Syntactic Structures", Mouton, The Hague, 1957.

[10] Gips J. and Stiny G., "Production Systems and Grammars: a Uniform Characterization", Environment and Planning B, Vol. 7, 1980, pp.399-408.

[11] Stiny G., "Introduction to Shape and Shape Grammars", Environment and Planning B, Vol. 7, 1980, pp.343-351.

[12] Pugliese M. and Cagan J., "Capturing a Rebel: Modeling the Harley-Davidson Brand through a Motorcycle Shape Grammar", Research in Engineering Design, Vol. 13(3), 2002, pp.139-156.

[13] Agarwal M., "Supporting Automated Design Generation: Function Based Shape Grammars and Insightful Optimization", in "Department of Mechanical Engineering", Carnegie Mellon University, Pittsburgh, 1999.

[14] Agarwal M. and Cagan J., "On the use of shape grammars as expert systems for geometry-based design", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 14, 2000, pp.431-439.

[15] Earl C.F., Shape grammars and the generation of designs, in Principles of Computer-aided Design, Ed. J. Rooney and J.P. Steadman, 1987, Pitman Publishing, London, pp.297-315.

[16] McCormack J.P., Cagan J., and Vogel C.M., "Crossing the '63 Riviera with a Concept Cielo: Capturing, Representing and Generating the Buick Brand", Proceedings of the ASME 2002 Design Engineering Technical Conferences, Montreal, Canada, 2002.

[17] Finger S. and Rinderle J.R., "A Transformational Approach to Mechanical Design using a Bond Graph Grammar", Proceedings of the First ASME Design Theory and Methodology Conference, Montreal, Quebec, 1989, pp.107-116.

[18] Finger S. and Rinderle J.R., Transforming behavioural and physical representations of mechanical designs, in Engineering Design Synthesis, Ed. A. Chakrabarti, 2002, Springer Verlag, London, pp.303-317.

[19] Paynter H.M., "Analysis and Design of Engineering Systems", MIT Press, Cambridge, USA, 1961.

[20] Schmidt, L.C., Shetty, H. and Chase, S.C., “A Graph Grammar Approach for Structure Synthesis of Mechanisms”, Journal of Mechanical Design, Vol. 122, 2000, pp.271-376.

[21] Shea K. and Cagan J., "Innovative dome design: Applying geodesic patterns with shape annealing", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 11, 1997, pp.379-394.

[22] Shea K. and Starling A.C., "From Discrete Structures to Mechanical Systems: A Framework for Creating Performance-Based Parametric Synthesis Tools", Proceedings of the AAAI'03 Spring Symposium: Computational Synthesis, Stanford, CA, 2003, pp.210-217.

[23] Kron G., "Diakoptics: the piecewise solution of large-scale systems", MacDonald, London, 1963.

[24] Hooke R. and Jeeves T.A., "Direct Search Solution of Numerical and Statistical Problems", Journal of the Association for Computing Machinery (ACM), Vol. 8(2), 1961, pp.212-229.

[25] Pareto V., "Manual of Political Economy", A. M. Kelley, New York, 1906.

[26] Bolognini F., 2003, "The Performance Aspects of Mechanical Design Synthesis", Report No. CUED/C-EDC/TR125, Cambridge.

[27] James B., Romax Technology, Private Communication, 2003.

[28] Arzethauser K., "Gears galore", in "Watch International", G. Kern and H.A. Pantli, 2003, pp.56-59.

[29] Knight T.W., "Transformations in Design", Cambridge University Press, Cambridge, 1994.

Contact author:

Dr Kristina Shea
Engineering Design Centre, Department of Engineering, Cambridge University
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
Tel. +44 1223 332666 Fax. +44 1223 332662 Email. ks273@cam.ac.uk