Model-based gearbox synthesis

Kristof Berx1*, Klaas Gadeyne1, Michiel Dhadamus2, Goele Pipeleers2, Gregory Pinte1

1Flanders’ Mechatronic Technology Centre
  Herzele, Belgium
  kristof.berx@fmtc.be, klaas.gadeyne@fmtc.be,
  gregory.pinte@fmtc.be

2Department of Mechanical Engineering, Division of
  Production Engineering, Machine Design and Automation
  KU Leuven
  Herzele, Belgium

Abstract—An important step in the design of a mechatronic system is the synthesis of an architecture that optimizes an objective function and satisfies constraints, originating from user requirements, design rules and physical laws. Such an architecture fixes the topology, consisting of the number and type of components and their connections, as well as their key properties. This architectural design problem becomes very complex if (i) the design space of possible architectures is large due to a high number of components and/or high number of ways to combine these components and (ii) lots of constraints have to be satisfied, which makes it hard to distinguish feasible from infeasible architectures.

An example is the architectural design of an automatic gearbox, where the lightest architecture of shafts, clutches and gears is searched. This article describes a computer supported synthesis approach for these automatic gearboxes. First, a formal declarative model of the design space exploration problem is created using SysML and OCL. The creation of this model accelerates the process of finding a clear and concise formulation of constraints and objectives. Based on this formal model, a constraint programming algorithm is developed which generates all possible gearbox topologies, satisfying the topological constraints. This algorithm generates dozens of gearbox topologies per second. In order to find the topology with the lowest achievable weight in this group, a procedure is created to calculate the optimal properties (locations and dimensions) of all components in a given topology. By the reformulation of several constraints into a convex form and an intelligent variable selection, a significant reduction in the computational complexity of this optimization problem is obtained. Because of the large number of possible topologies, it is impossible to perform this property optimization for all of them. To this end, clustering techniques were implemented to limit the number of topologies for which the property optimization is performed.

The developed procedure has been successfully applied to derive the optimal architecture for automatic gearbox design problems with up to six transmission ratios.

I. INTRODUCTION

It is generally accepted that a large fraction of the cost of designing a new mechatronic system originates from design decisions made in the earliest stages of the design [1]. One important stage is the embodiment or architectural design stage [2, 3, 4], in which the architecture of the new system is chosen. This architecture fixes the topology, consisting of the number and type of components and their connections, as well as some key properties (size, material...) of the different components. The key properties are only those properties that affect the optimality of a topology. All other properties are fixed in the later design stages.

For some design problems, typically with a limited number of possible components combinations, the selection of the optimal architecture can be trivial. In other cases, however, many components can be combined in numerous ways and lots of constraints have to be taken into account. An example of the latter is the architectural design of an automatic gearbox, where the gearboxes with minimal weight is sought that (i) realizes a given set of discrete transmission ratios, (ii) is able to handle a given torque at the ingoing shaft; and (iii) fits within a given bounding box. Because of the large number of complex constraints, along with the potentially very large design space, finding the optimum in an automated way could help a designer to come up with innovative architectures, or confirm the current designs. This process is often referred to as Design Space Exploration (DSE). In the field of vehicle gearbox design, a manual, experience-led approach is generally used to generate gearbox architectures. This kind of approach suffers from design fixation [5]: in case of new user requirements always gearboxes are designed with similar topologies present in existing gearboxes.

At topology level, a gearbox is described as the connection of a number of shafts, clutches and gears. Figure 1 shows a schematic representation of one of the hundreds of possible gearbox topologies that realizes four discrete transmission ratios.

*Corresponding author
Some of the design constraints or objectives are purely determined by the system’s topology. An example of such a constraint is the number of transmission ratios that has to be realized, which is only determined by the connections between the different components. Other constraints or objectives however can only be checked by assigning properties to the different components. An example of such a constraint is that all components have to fit within the bounding box. In order to evaluate the latter type of constraints or objectives, all components are approximated as a cylinder with the dimensions (length & diameter) and the location (x, y & z coordinates) of its centre as key properties. These properties are sufficient for evaluating the design objective, i.e. the total weight, and the different constraints at property level, such as the bounding box or the torque limit per component. Other component properties, such as the number of teeth per gear, do not affect the optimality, i.e. the minimal achievable weight, of a gearbox topology and are thus not considered as key properties.

Design space exploration has already successfully been used within many subdomains of mechatronic design. Examples are software deployment on a distributed set of ECUs [6], software product line configuration [7], electrical circuit design [8] and mechanical system design [9-15]. Most research in the area of DSE for mechanical systems deals with either the property optimization [9] or topological optimization [7,10] only. This paper presents an approach to solve the problem of complete architectural optimization for a mechatronic system. Papers focusing on solving these kinds of problems are often based on graph grammars, and use a simultaneous optimization of both the topology and the component properties [6,8,11-13]. The use of graph grammars is however limited to mechanical systems consisting of only two types of mechanical components, e.g. a gear reducer consisting of just shafts and gears. To solve the gearbox design problem, which contains three component types (gears, shafts and clutches), an approach is presented in this paper that decouples the problem into a topological level and a property level. A similar split up has been chosen by others [14-16]. However, the solvers they use at topology level progressively make small variations to a given topology in order to create a new topology. These kinds of solvers are not suited in case of complex constraints as it becomes very difficult in this case to define topology variations that maintain the feasibility of the created topology variant. Our approach solves this issue by a combination of (i) constraint programming to generate all topologies satisfying the complex constraints, (ii) clustering techniques to intelligently search through these topologies and (iii) a gradient-based optimization to evaluate the topologies by optimizing the key properties of all the components. The presented use case focuses on the synthesis of the mechanical components in an automatic gearbox because of the high complexity of the mechanical design constraints. The hydraulic and electrical components in a gearbox are not taken into consideration in this use case.

This paper describes how an algorithm for the optimization of gearbox architectures was created and is organized as follows. Section 2 describes the formal model of the gearbox design space exploration problem. Sections 3 to 6 describe the solution strategy. Section 7 shows the results achieved and the last section presents some conclusions and directions for future research.

II. DESIGN SPACE MODEL

In order to deal with the complexity of a mechatronic design problem, a formal model of the design space exploration problem is created in SysML [17] and OCL [18]. This model intuitively and graphically describes the different components and their possible connections, and uses an object-oriented formulation of the constraints and objectives. Creating this model accelerates the process of finding a clear and concise formulation of constraints and objectives and helps in choosing an appropriate solution algorithm.

Firstly, the design space model describes the possible components in the system, their properties and the possible connections between components. These are the variables of the DSE problem. Secondly the objective function, in the case

Figure 1. Schematic representation of gearbox topology, consisting of shafts, gears and clutches and realizing 4 transmission ratios, indicated by arrows.

Figure 2. Diagram of the design space model, without any constraints.
of the automatic gearbox minimal weight, is added to the model. Figure 2 shows a block diagram with the variables and objective function. This diagram shows which component types make up a gearbox (indicated with the arrows ending in ♦) and shows which of these component types can be connected (indicated by the solid lines).

The constraints are also incorporated into the model. These can originate from:
- user requirements: for example a gearbox has to fit in a given bounding box,
- design rules: for example each gear can only be connected to one shaft, or
- physical laws: for example no interference may occur, meaning that two or more components cannot occupy the same space.

The complexity of the constraints can vary significantly. Simple constraints, for example restrictions on the number of connections per component, can be directly modeled in SysML. Complex constraints, affecting the complete system, are modeled in OCL. Some constraints even require the definition of additional concepts. For example, the most complex constraint in the gearbox architecture case is the prevention of mechanical loops.

A mechanical loop arises when two shafts are connected by two different series of gear pairs such that the gearbox blocks. A possible way to check this constraint would be to (i) count the number of possible routes between every pair of shafts using only gear connections and (ii) constrain this number to be one or less. This would require counting the routes between n∙(n-1) shaft pairs, with n the number of shafts, resulting in an overly complex constraint. To accelerate the process of finding a good formulation of this constraint, a simple gearbox editor, derived from the SysML model, was created. This editor allows validating constraint formulations on manually created gearbox instances. Figure 3 shows a gearbox instance, with two mechanical loops, created this way.

Based on these manually created gearbox instances, it is clear that mechanical loops are only found within groups of shafts connected by gear pairs, further referred to as geared sets. Preventing a loop in such a geared set can be achieved by simply constraining the number of gear pairs in the geared set to be exactly one less than the number of shafts in the geared set. Figure 4 shows a diagram of the final formulation of the mechanical loop prevention constraint. The geared set concept is added to the diagram, and the constraint is formulated as an OCL constraint on the number of shafts and gears associated with this geared set.

III. SOLUTION STRATEGY

To find the gearbox architecture with the lowest weight the optimization problem described in the SysML model needs to be solved. This section describes the developed solution strategy.

A first challenge is to generate the different topologies fulfilling the topological constraints. The approaches described in [14-16] use small incremental variations to search through the topological design space. However, because of the complexity of the constraints, this approach is infeasible for solving this problem. To deal with this, constraint programming is used to generate all topologies.

A second challenge is to find the optimal properties of a given topology. To this end, a property optimization algorithm is used. This algorithm searches the values for the dimensions and location of each of the components minimizing the weight while fulfilling a number of constraints using a gradient-based solver tailored for this problem.

Because of the high number of topologies and the non-negligible computational time of the property optimization, it is impossible to search the optimal properties for all topologies. So a third challenge is to limit the number of
When creating the formal design space model, the concept of geared sets, groups of shafts connected by gears, was defined. This concept is now used to split the constraint program into two phases. In the first phase all combinations of geared sets are generated and it is fixed how these sets are connected through clutches, such that the required number of transmission ratios is satisfied. The second phase, which is executed for every geared set combination generated in the first phase, populates the different geared sets with shafts and gear pair combinations, such that among others the mechanical loop constraint is fulfilled. By splitting the topology generation problem into two smaller and less complex problems, the total computational time can be greatly reduced. The process is shown in figure 5.

The two constraint programs are implemented in Gecode. The first phase is capable of generating about 1500 geared set combinations per second. The second phase can generate up to 400 topologies per second for a given geared set combination. However, because of the overhead in calling the second constraint program thousands of times, once per solution of the first CP, the complete algorithm only generates up to 20 topologies per second. However, this is still a significant improvement compared to the initial single phase constraint program which, because of the complexity of the problem, only generated a single topology per second on average.

B. Property optimization

To find the optimal gearbox architecture, an algorithm is needed to check the feasibility and optimality of the topologies generated by the constraint program. This requires determining the minimal achievable weight by optimizing the dimensions and location of all components in that topology taking into account the bounding box, torques and interference. If no solution is found, a weight of $\infty$ is returned.

Since all components are approximated by parallel cylinders, there are 5 variables for each component, namely their $x$, $y$ and $z$ coordinates and their length and radius, as shown in figure 6. In order to reduce the computational complexity, the total number of variables was reduced as much as possible by eliminating linear equality constraints. For example, when shafts are connected by clutches, they have equal $x$ and $y$ coordinates, while the $z$ coordinate of the second shaft follows from the shaft and clutch lengths, so for a series of connected shafts and clutches only a single $x$, $y$ and $z$ coordinate is used. In addition, the location of the ingoing and outgoing shaft was fixed.

The optimization problem has to take constraints on the properties into account. For some of these, a convex formulation was possible. E.g. "all components must fit inside a given bounding box" resulted in (1).

$$\forall \text{Component:} \begin{cases} x - r > x_{\text{Min}} \\ x + r < x_{\text{Max}} \\ y - r > y_{\text{Min}} \\ y + r < y_{\text{Max}} \\ z > z_{\text{Min}} \\ z + l < z_{\text{Max}} \end{cases}$$ (1)
The constraints enforcing the components’ torque limits, the transmission ratios and the fact that paired gears must touch, on the other hand, are non-convex.

The hardest constraints to enforce in the property optimization problem are the so-called interference constraints, specifying that “no two components can take up the same point in space”. Stated otherwise, the distance between all components must be positive. A continuous formulation of this constraint was obtained from Lagrange duality applied to the distance determination problem [22] was used. After further simplifications, (2), an equivalent formulation of the interference constraints, is obtained.

\[
\forall \text{Component 1, Component 2} : 3\lambda, \alpha : \\
\begin{cases}
\lambda \cdot (z_1 - z_2 - l_2) + (\alpha - \lambda) \cdot (z_2 - z_1 - l_1) + \cdots \\
(1 - \alpha) \cdot ((x_1 - x_2)^2 + (y_1 - y_2)^2 - (z_1 + z_2)^2) > 0 \\
0 \leq \lambda \leq \alpha \leq 1
\end{cases}
\] (2)

This reformulation introduces two additional variables, \(\lambda\) and \(\alpha\), per combination of two components.

Adding these interference constraints between all possible combinations of two components was found to compromise the solvability of the optimization problem dramatically. To speed up the algorithm and to prevent the solver from getting stuck at infeasible solutions or inadequate local optima, a sequential optimization procedure is adopted. In the first step all interference constraints are omitted. Using the first step’s solution as starting point, a second optimization problem is solved where interference is circumvented by enforcing sufficient radial distance between shafts. In the last optimization problem, interference constraints (2) are added, but the \(z\)-order of the components is fixed from the second step’s solution. This allows reducing the number of component pairs for which interference constraints must be enforced.

The three-step optimization procedure is implemented in Python. The optimization problems are modelled with Casadi [23] and solved with ipopt [24]. Extensive numerical experiments revealed that the procedure finds an adequate optimum for the property optimization problem within a few seconds (5 seconds on average for 4 transmission ratios and 10 seconds for 6 transmission ratios).

C. Topology selection

A property optimization has to be performed for every topology generated by the constraint program in order to find the optimal gearbox architecture. However, the number of topologies that satisfy the topological constraints strongly increases with the required number of transmission ratios, resulting in more than 100,000 feasible topologies for just 6 ratios. Optimizing the properties for all of them would take weeks on a standard pc, which is unacceptably long. In order to reduce the computational time of the complete algorithm, a selection procedure is developed. This procedure limits the number of topologies for which a property optimization has to be performed before the optimum is found. The different steps are clarified, using the gearbox design problem with 4 transmission ratios as an example.

In a first step, a random sample of 10,000 topologies is chosen from the complete set of 284,996 topologies generated by the constraint program. For each topology in this set, a number of features is evaluated, which are expected to have a correlation with the minimal achievable weight of the topology. Ten features were taken into consideration, such as the number of shafts, the number of geared sets and the maximum number of shafts per geared set. Figure 7 shows 3 example topologies, with these feature values, from the set of 10,000 samples.

In a second step, a property optimization is performed for each of the sampled topologies, returning its minimal achievable weight. If no solution is found by the property optimization, the returned weight is \(\infty\). Figure 8 shows the results of the property optimization for the three example topologies with one topology, which could not be fitted in the bounding box, having a weight of \(\infty\).

Table 1 summarizes the results of the first two steps are, for the number of shafts feature. The third step consists of finding relations between the different features, evaluated in the first step, and the minimal weights, evaluated in the second step. These relations are used to add constraints to the topologies. Equation (3) shows an example of a constraint derived from Table (1).

\[8 < \text{number of shafts} < 11\] (3)

After performing this procedure, the properties of the topologies are only optimized if they satisfy the new constraints. Using this procedure, the number of property optimizations can be reduced by up to 75%, depending on...
how strict the resulting constraints are made. The implemented topology selection showed that the number of shafts and the maximum number of shafts per geared set were the most important features to distinguish good topologies.

D. Summary

The resulting procedure to find the optimal gearbox architecture comprises of the following steps:

1. Generate all topologies realizing the required number of transmission ratios and satisfying the topological constraints.
2. Choose a random sample set from the complete set of topologies and evaluate the features of each chosen sample.
3. Perform a property optimization for each of the sample topologies, collect the resulting weights and fill in a table such as table 1 for each feature being taken into consideration.
4. Determine suitable constraints on the value of the different features.
5. Perform an exhaustive search over all remaining topologies. Skip all topologies with unacceptable feature values. Perform a property optimization for all topologies with acceptable feature values.

IV. Results

To test the effectiveness of the developed algorithm, a problem with 6 transmission ratios and a realistic value for the torque and bounding box was solved. The resulting algorithm was run on a dell latitude e6420 laptop with a 2.6 GHz dual core and 3 GB RAM.

The constraint program generated 285,000 topologies in 8.5 hours. The topology selection procedure used a 10% sample to determine the optimal feature values and further optimized the 10% of the topologies fulfilling the resulting constraints on those features. The algorithm completed the search process in a couple of days and returned the solution shown in figure 9.

V. Conclusions

This work described how an algorithm for the optimization of gearbox architectures was created. Because of the complexity of the constraints, a formal design space model is used to describe the problem. This model helps in determining a good constraint formulation and in choosing a suitable solution strategy. This algorithm consists of three levels. At the top level, all topologies are generated using a two-phase constraint program. At the bottom level a gradient-based optimization selects the properties minimizing the gearbox weight for each topology. A selection procedure reduces the number of property optimizations needed to find the optimal architecture.

The presented use case was purely mechanical, however, the presented approach can be applied to any mechatronic design that consists of a limited set of component types that have to be combined such that they achieve a number of user requirements in an optimal way.

VI. Further work

This paper focuses on minimizing the weight of an automatic gearbox. Further research focuses on two extensions.

1. Using the total cost of ownership of the gearbox as objective function. This cost consists of both installation cost as well as fuel consumption of the vehicle using the gearbox. The installation cost is an analytical function of the key properties of the components. The fuel consumption on the other hand has to be calculated using a dynamic simulation of the vehicle containing this driveline. The optimization algorithm thus has to call an external simulation tool, which makes it a challenge to limit the computational time for the gearbox optimization. Another challenge is the automatic generation of a dedicated controller for any gearbox architecture and accumulator dimensions.

2. Hybridization of the driveline by adding energy storage in the form of a hydraulic accumulator. This will add an additional outgoing shaft of the gearbox to the pump/hydromotor, greatly increasing the number of possible topologies. The properties of the accumulator will be added to the property optimization, and to optimize all these properties the

<table>
<thead>
<tr>
<th>Number of shafts</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of topologies with this feature value [%]</td>
<td>1.8</td>
<td>12.4</td>
<td>24.2</td>
<td>32.7</td>
<td>29.0</td>
</tr>
<tr>
<td>Percentage of feasible topologies with this feature value [%]</td>
<td>10.7</td>
<td>44.3</td>
<td>36.6</td>
<td>7.6</td>
<td>0</td>
</tr>
<tr>
<td>Average weight of the feasible topologies [kg]</td>
<td>117</td>
<td>116</td>
<td>106</td>
<td>107</td>
<td>∞</td>
</tr>
</tbody>
</table>

TABLE I. EXAMPLE TABLE FOR THE NUMBER OF SHAFTS. THIS TABLE SUMMARIZES THE RESULTS OF THE FIRST TWO STEPS IN THE TOPOLOGY SELECTION PROCEDURE.

Figure 9. Visualisation of the gearbox topology and architecture found by the optimization algorithm.
total cost of ownership has to be minimized, requiring again a dynamic simulation of the vehicle containing the hybrid driveline. The model of the driveline will have to be expanded with the pump/hydromotor and the accumulator and the controller will have to be expanded to control the energy flows to and from the accumulator.

REFERENCES