Article:

Hambley, C.J., Jones, B.L. orcid.org/0000-0002-7465-1389, Griffin, I. et al. (2 more authors) (2018) Optimized synthesis of cost-effective, controllable oil system architectures for turbofan engines. Systems Engineering, 21 (5). pp. 417-431. ISSN 1098-1241

https://doi.org/10.1002/sys.21430

Optimized Synthesis of Cost-Effective, Controllable Oil System Architectures for Turbofan Engines

Christopher J. Hambley*[1], Bryn L. Jones[1], Ian Griffin[2], Andrew R. Mills[1] and Visakan Kadirkamanathan[1]

[1] *Rolls-Royce Control and Monitoring Systems University Technology Centre, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, United Kingdom*
[2] *Rolls-Royce Plc, Derby, United Kingdom*

ABSTRACT

Turbofan oil systems are used to provide lubrication and cooling in the engine. There is an increasing interest in oil system architectures which utilise electric pumps and/or valves to give optimized control of flows to individual oil chambers, leading to improved thermal management of oil and lubrication efficiency. The challenges here lie in the tradeoff between increasing controllability and minimising the addition of new components, which adds unwanted production and maintenance costs. This paper formulates the oil system architecture design as a constrained, multi-objective optimization problem. An architecture is described using a graph with nodes representing components and edges representing interconnections between components. A fixed set of nodes called the architecture template is provided as an input and the edges are optimized for a multi-criteria objective function. A heuristic method for determining similarities between the different oil chamber flow requirements is presented. This is used in the optimization to evaluate the controllability objective based on the structure of the valve architecture. The methodology provides benefits to system designers by selecting cheaper architectures with fewer valves when the need to control oil chambers separately is small. The effect of manipulating the cost/controllability criteria weightings is investigated to show the impact on the resulting architecture.

Key words: SEE06 Architectural Design; AS01 Transportation & Storage

1. Introduction

A system architecture is a definition of the system structure including the major components and the way in which they are connected in order to meet the system requirements [Firesmith, 2008]. Crawley [2015] defines system architecting as "the embodiment of concept, the allocation of function to elements of form, and definition of relationships among the elements and with the surrounding context". This high-level decision making process is a key part of the systems engineering discipline [INCOSE, 2006]. The way these decisions are made can vary greatly between different applications. Selva [2017] shows 6 main patterns of architectural decision making including: combining, downselecting, assigning, partitioning, permuting and connecting. Regardless of the pattern followed for a particular application, it is important for the architectural decisions to be made using a well-defined process. The impact of effective architecture design is highlighted in a recent survey of 46 industrial defense contractors, which shows a strong positive correlation between increasing system architecting activities and improved product development cost, schedule and scope goals [Elm, 2008]. Despite these clear benefits, Gustavsson [2011] shows a marked need for an improvement in the uptake of system architecting activities within industry. The research found that even a world-leading industrial automation company with more than 50% of the global market share still has no formal architecting process. Georgiadis [2013] also discusses this problem, highlighting a U.S. Government Accountability Office review which found that 10 defence programs out of the 32 investigated pursued a pre-selected solution without performing any analysis of alternative architectures. This motivates the need for more research into system architecting activities, as wider uptake by companies will rely on the continued development of effective tools and frameworks.

There are various types of architecting processes with different levels of formality. Improved project performance can be gained even through less formal approaches whereby the majority of systems architecture decisions are carried out using the knowledge and expertise of engineers, but in a well-defined task-flow [Fröberg, 2014]. More formal approaches attempt to automate some of these tasks and use optimization to produce optimal architectures that maximize satisfaction of some objective function and ensure system requirements or constraints are met [Bajaj, 2015; Finn, 2015; Ham-

* Author to whom all correspondence should be addressed (e-mail: cjhambley1@sheffield.ac.uk).

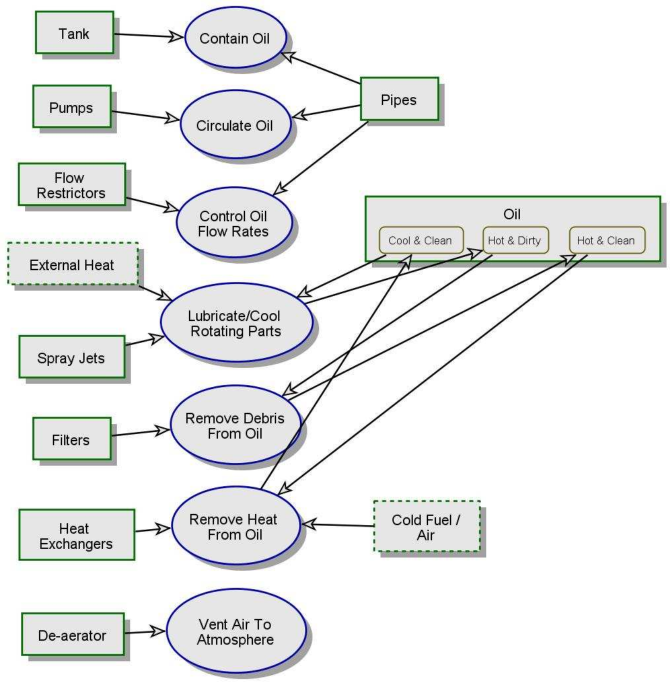


Figure 1: An object process diagram [Dori, 2002], showing the main objects and processes of the oil system. Blue ellipses represent processes, green rectangles represent objects and brown, rounded rectangles represent states of the oil. Arrows indicate which objects are consumed by which process, and how the processes change the state of the oil.

mami, 2014; Hammami, 2015; Nuzzo, 2014; Subramanian, 2016; Thompson, 2015; Wichmann, 2015]. Use of optimization also ensures a fuller exploration of the search space and limits the effects of cognitive biases. This paper follows a more formal approach, presenting a framework for optimization of controlled flow network architectures. The method is demonstrated on a turbofan oil system architecture case study.

The oil system is a vital part of a turbofan engine, providing the dual functions of lubrication and heat removal in the bearings and gearboxes. Components within an oil system architecture consist of: tanks to contain oil; pumps to move oil around the system; filters to remove debris; heat exchangers to remove heat; pipes and flow restrictors to control flow rates; oil chambers with jets directing flow to bearings or gears; deaerators and breathers to vent air to the atmosphere [Linke-Diesinger, 2008; Rolls-Royce, 2005]. This is shown graphically in Figure 1. The pumps and flow restrictors that determine the amount of oil provided to the bearing chambers are typically not actively controlled in Rolls-Royce Trent [Rolls-Royce, 2005], GE, CFM or Pratt & Whitney engines [Linke-Diesinger, 2008]. The pumps are driven by a fixed gear in the accessory gearbox, providing an output flow proportional to the speed of the high pressure shaft [Linke-Diesinger, 2008; Rolls-Royce, 2005]. This lack of oil flow controllability can lead to problems such as exceeding oil temperature constraints, which leads to oil degradation and higher maintenance costs. This is a particular issue during transient manoeuvres. For example, when reducing thrust the shaft speed slows more quickly than the temperature in the oil chambers due to the thermal capacitance of the metals. With a reduced oil flow, but sustained high temperature, the maximum allowable oil temperature can be exceeded [Rolls-Royce, 2005]. Challenges such as this are likely to be even more evident in the new generation of geared turbofan engines such as the Pratt & Whitney PW1000G and the Rolls-Royce Ultrafan$^{\text{TM}}$. The power gearbox in these engines creates substantial new demand for lubrication and cooling. The 22MW power gearbox on the PW1000G engine generates huge amounts of heat despite being highly efficient (e.g. 1% inefficiency produces 220kW of waste heat to be absorbed by the oil system) [Jones, 2015]. This motivates research into novel oil system architectures. Of particular interest is the ability to utilise electrically driven pumps and variable flow restrictor valves, to provide optimal flow to the individual oil chambers at all stages of the flight cycle. This removes the need to constantly oversupply oil during transients and thus reduces parasitic losses on the system efficiency. In addition, better thermal management of oil means that properties such as viscosity can be more closely controlled, improving lubrication system performance and increasing component life.

Choosing the controlled oil system architecture is a multi-objective problem. It is desirable to increase the controllability of oil flows around the system, but at the same time the production cost and weight of the system has to be kept low. This presents a *tradeoff* which must be handled by the system designer in some kind of multi-criteria decision making environment. In addition to this there are safety, reliability and power consumption constraints which cannot be violated. This paper presents a method for handling all of these considerations in a multi-objective optimization framework.

The remainder of this paper is organised as follows: Sections 2.1 to 2.3 present the related work on systems architecting and architecture optimization. Section 2.4 shows how system architecture design fits into a wider multilevel framework for designing complex engineered systems. The main contribution of this paper comes in Section 3. This formulates the oil system architecture design problem as a multi-objective optimization problem, with a heuristic algorithm for quantifying architecture controllability. Section 4 presents the resulting architectures and explores the tradeoffs between cost and controllability. Concluding remarks and future research opportunities are presented in Section 5.

2. Related Work

Guidelines for system architecting are given by INCOSE [2006] covering a range of activities from obtaining customer requirements to verification of the design.

Firesmith [2008] presents another framework for systems architecting called the Method-Framework for Engineering System Architectures (MFESA). This covers the entire systems architecture process for an industrial company, which includes activities like assigning engineering effort. The areas of interest for this paper focus on MFESA tasks T5 to T8 which look at generating a list of suitable architecture candidates and selecting the best choice. A variety of approaches for carrying out these tasks are presented in Sections 2.1 to 2.3.

2.1. Informal Approaches

It has been established by Gustavsson [2011] that some substantial multinational companies still do not have a formal process for designing complex system architectures. The reasons for this can be due to lack of understanding or knowledge of more formal processes such as optimization techniques. However, improvements can still be made from use of more simple techniques. The INCOSE systems engineering handbook [INCOSE, 2006] does not provide specifics about how the systems architecting tasks must be accomplished. Therefore tasks such as "evaluate alternative design solutions" could be implemented in a straightforward fashion, simply by using subject-matter experts (SMEs) to rank candidate architectures against various criteria in a Pugh matrix [Pugh, 1991]. The same is true of many of the activities within the MFESA framework [Firesmith, 2008].

Another SME-driven architecting procedure is the 9-step method presented by Fröberg [2014]. This lists activities from stakeholder requirement ilicitation to validation of a chosen architecture. In this approach architectural candidates are listed via brainstorming sessions and a final architecture chosen through use of SMEs. The method is demonstrated on an automotive telematics system case study.

A big disadvantage to SME-focused approaches is that they can sometimes lead to design fixation [Jansson, 1991] with decisions echoing the engineers' previous experiences and neglecting novel ideas. This means the resulting architectures are often evolutionary rather than revolutionary, potentially missing out on the benefits of new technological advancements. However, by following an SME-driven architecting procedure rather than none at all, companies are still likely to see noticeable improvements in project performance [Elm, 2008]. It is also worth noting that SMEs can be effective in removing unsuitable architectures from consideration early in the design, where unnecessary lower-level analysis would be costly or time-consuming [Rekuc, 2006].

2.2. Architecture Design as a Component Selection Problem

A more formal way of system architecting is to have a *library* of components with their models and perform architecture design as a composition of these library elements. This naturally involves two stages as highlighted in [Agarwal, 2012; Calvert, 2011]:

1) Modeling phase: where the library of component models is populated either from first-principles, system identification or legacy models. It is noted that when designing complex systems there is a need to address the systems architecture problem at higher levels of abstraction, where simpler models facilitate more rapid analysis of alternatives. Therefore the modeling phase may also contain a bottom-up approach whereby high-level abstract models are derived from their high fidelity descriptions [Agarwal, 2012].

2) Component selection phase: where the architecture is constructed as a composition of components. A set of rules define the minimum and maximum number of different components of each type and the permitted interconnections. The components are then composed according to these rules until a set of formal system requirements are met.

In [Agarwal, 2012; Calvert, 2011] three different algorithms are used to implement the component selection phase: two "Greedy" algorithms and one simulated annealing approach. The effectiveness of the algorithms is demonstrated on a Network on a Chip (NoC) case study. The problem with the three component selection approaches here is that they only iterate until the system specification/architecture constraints are met. Therefore the algorithm may miss better architectures that satisfy the performance specification more robustly, or for cheaper cost. The reason the authors state for not following a more exhaustive optimization approach is due to the size of the search space [Agarwal, 2012; Calvert, 2011]. If the system design problem could be posed as a convex optimization, this large search space would not be prohibitive. However, it is known that system design is an NP-complete problem, as proved by Chapman [2001]. This means that optimal systems cannot be designed with deterministic, polynomial-time procedures. Fortunately system design does not need to be carried out in real-time and architecture optimization (see Section 2.3) can be performed in a reasonable timeframe for smaller systems. For example, the approach presented in this paper produces oil system architectures in less than 10 seconds on a standard desktop PC.

Albarello [2012] solves the architecture design problem using a genetic algorithm (GA) approach. It starts with the functions that must be performed by the system and a library of components. Each function is assigned to a component from the library. If the function cannot be met by a single component alone, then increasingly large chains of components are investigated until the function is met. Using this method a population of potential candidates for the architecture is generated. The GA is then an iterative algorithm which generates a population of new architecture candidates based on the best individuals from the previous population. The method is applied to an aircraft cockpit design case study. The downside to this method is that GAs are not able to guarantee finding a globally optimal solution. Several methods for improving this are suggested including reducing the design space, considering the architecture performance in the synthesis of candidates and using constraint programming [Albarello, 2012].

Design of system architectures as a composition of elements from a component library is also presented in [Bajaj, 2015; Nuzzo, 2014]. In these approaches optimization is used to ensure that the resulting architecture is *optimal* according to some objective function, rather than just satisfying the system requirements (constraints). This is discussed in more detail in Section 2.3.

2.3. Architecture Design as a Constrained Optimization Problem

As discussed in Section 2.2, it is possible to have multiple different system architecture candidates that satisfy the system requirements. Constrained optimization is a method for determining the best architecture from a set of candidates by finding the solution which maximises some desired objective such as minimising cost. The advantage of this is that it enforces a fuller exploration of the search space. In addition, the objective nature of optimization limits the cognitive biases inherent to SME-driven techniques (Section 2.1). Hammami [2014] presents the *generic* systems architecture optimization (SAO) problem as comprising of three elements:

1. f - the objective function
2. R - a set of constraints
3. A - a set of architectures built in a framework N

The SAO solution finds a subset of A which minimises/maximises f whilst satisfying R. This is a generic problem and the paper does not refer to any specific optimization schemes since often these need to be tailored to suit the application.

An application of the SAO problem is presented for optimal design of an aircraft electric power system in [Bajaj, 2015; Nuzzo, 2014]. In this framework an architecture is defined as a directed graph with components represented as nodes $\{N_1, \cdots, N_n\} \in \mathcal{N}$ and interconnections between nodes N_i, N_j represented by edges $e_{i,j} \in E$ where:

$$E := \begin{bmatrix} e_{1,1} & \cdots & e_{1,n} \\ \vdots & \ddots & \vdots \\ e_{n,1} & \cdots & e_{n,n} \end{bmatrix} \in \mathbb{B}^{n \times n}, \tag{1}$$

and $\mathbb{B} := \{1, 0\}$ is the Boolean set, with $e_{i,j} = 1$ indicating a connection between components i and j and $e_{i,j} = 0$ indicating no connection. Each node has different attributes which correspond to the design objectives. Therefore inclusion/exclusion of a node from an architecture will have an effect on the overall objective function score. The set of nodes can be partitioned into subsets of components of similar types. For example $\mathcal{N}$ is partitioned as $\{\text{Tank}, FP, HE, \text{Valve}, OC, SP\}$ in Figure 2. An architecture *template* is a set of nodes which are fixed. There may also be some connections between nodes which are fixed in the template as in Figure 2. Note that the architecture template represents the maximal node configuration. There is no requirement for every node in the template to be used in the final architecture. The architecture optimization problem is then to determine the optimum set of connections between components to minimise the objective function f whilst satisfying the system requirements/constraints R. Any nodes which are not connected to other nodes in the final architecture are discarded. Note that this is approach is an example of a *connecting* architectural decision making process as described by Selva [2017].

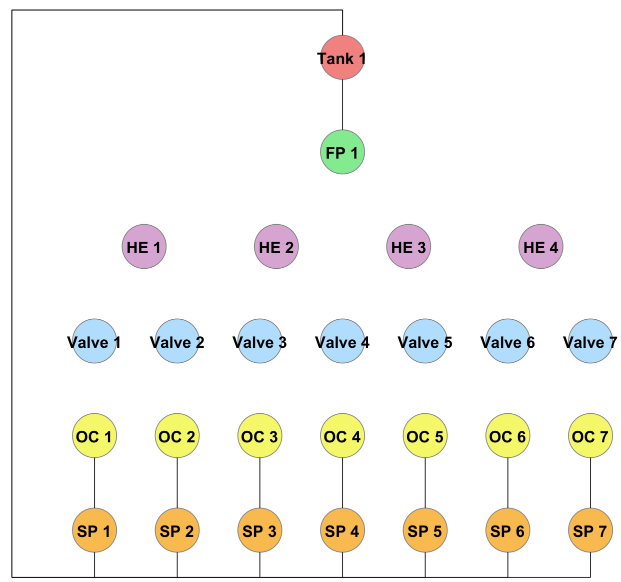


Figure 2: An architecture template for the actively controlled oil system. Connections between the tank, fuel pump (FP), oil chambers (OC) and scavenge pumps (SP) are fixed. The heat exchanger (HE) and valve connections are yet to be determined by the optimization algorithm. Any HE or valve nodes which are not connected to other nodes by the optimization algorithm are not included in the final architecture.

In [Bajaj, 2015; Nuzzo, 2014] the methodology is applied to an aircraft electric power system (EPS) case study. Here the objective function is focused on minimising the cost of the architecture (number of nodes included) and complexity (number of connections amongst components). The interconnection constraints enforcing rules for how components should/should not be connected are expressed as inequalities on the edges $e_{i,j}$. In addition there are reliability constraints which are expressed as inequalities on combinations of the component reliabilities and the edges $e_{i,j}$. As the decision variable in this optimization problem is the Boolean matrix E (1), this is known as an integer program (IP). IPs

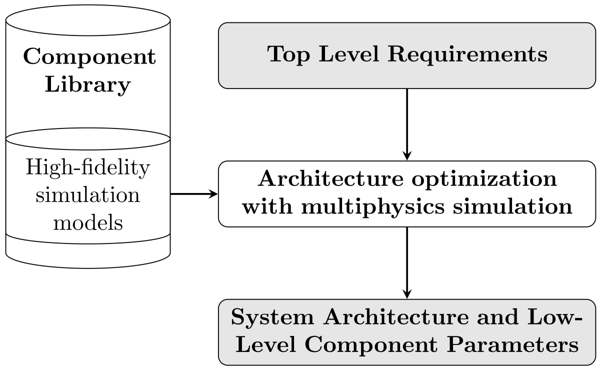


Figure 3: An overview of the multiphysics approach to architecture optimization. The cyclindrical bucket represents the library of components. The grey boxes are the inputs and outputs to the design flow. The white box indicates a design process. Here the design is carried out only at one high-fidelity layer of abstraction.

can be solved using software such as the MATLAB toolbox YALMIP [Löfberg, 2004].

In [Agarwal, 2012; Bajaj, 2015; Nuzzo, 2014] the architecture design is carried out at a high-level of abstraction using low-fidelity steady-state models for the components. Some research takes a different approach whereby the low-level, high-fidelity components models are simulated directly within the optimization algorithm [Hammami, 2015; Wichmann, 2015]. A schematic of this design flow is outlined in Figure 3. This has the advantage of being able to assess the low-level performance of candidate architectures visited by the optimization scheme. The performance of these high-fidelity simulations is the closest approximation of the real system performance and hence these methods should yield the best architectures. Unfortunately there are various downsides to these approaches. Firstly it can be impractical to simulate high-fidelity representations of more complex systems in a reasonable time-frame. In addition, when the system architecture is chosen at the start of a complex product development, these high-fidelity component representations may not have been developed. One of the key limitations for the actively controlled oil system is that the low-level performance cannot be evaluated without the controller, but the controller cannot be designed without the architecture of the system. This problem is common in any control system and hence multilevel approaches have been developed (see Section 2.4).

A recent attempt to address some of these problems through use of a two-level optimization scheme is presented by Finn [2015]. At the upper level the algorithm produces an architecture candidate using low-fidelity steady-state models. The architectural candidate is then passed to the lower level where sizing of the individual components is optimized using high-fidelity models. When no feasible component sizing can be found for a candidate architecture, a new set of constraints is added to the high-level optimization problem. For example, consider a high-level architecture that leads to a flow through a given component A which exceeds its upper bound in the low-level simulation. In this case, a new constraint can be added to neglect all architectures with upper bounds on flow which are smaller than the upper bound of A [Finn, 2015]. This process is repeated until an architecture is produced with a valid component sizing to meet all system requirements. Here the iterative mapping between the two levels is carried out automatically. This means the only inputs required are the system requirements and library of components with their interconnection rules. The algorithm will then run until a feasible architecture with optimum component sizings is reached.

Another multi-level approach to system design is presented by Miller [2015]. Here the design progresses sequentially as the set of potential solutions is narrowed down to a choice set, with increasing levels of fidelity in the models used to make decisions. Low fidelity models are coupled to higher fidelity models through use of bounding functions which specify the upper and lower bounds on variables. This allows information from the detailed models to be considered at the conceptual design stage, without the need for complex analysis or simulation of the high-fidelity details [Miller, 2015].

Subramanian [2016] presents another multi-level optimization framework for systems-of-systems (SoS) architectures. Here the framework follows a hierarchical structure with three levels resembling a tree of optimization problems. The method is applied to a noise-optimal aircraft design problem with: optimization of aircraft trajectories at the SoS level; optimization of aircraft designs at the system level; and optimization of turbojet thrust and airfoil shape at the sub-system level. Thompson [2015] also considers SoS architecture optimization. This approach uses multi-objective optimization of all the SoS decision variables in a single Mixed Integer Non-Linear Program (MINLP), with links to dynamics and performance models to evaluate candidate solutions.

2.4. Systems Architecting Within a Platform-Based Design Framework

The concept of platform-based design (PBD) was presented by Sangiovanni-Vincentelli [2007]. In this framework system design is carried out at a series of abstraction levels called *platforms*. The system design at each platform is termed a *platform instance* and this is achieved through composition of platform library components. As in the component selection approaches discussed in Section 2.2 this naturally involves a bottom-up abstraction phase, whereby high-level platform models are developed, as well as the top-down design phase. A key aspect of PBD is the concept of *mapping* between system designs at adjacent platforms. For example, Nuzzo[2014] carries out the mapping through use of assume-guarantee contracts [Sangiovanni-Vincentelli, 2012]. The concept of contract refinement is used to ensure that designs at lower platforms are a valid implementation of the more abstract design at the platform above.

The number and type of platforms will vary between applications. For example, in Sangiovanni-Vincentelli [2007] the platforms chosen for a wireless sensor network application are: the sensor network service platform (SNSP), to specify the functions of the network without dealing with details of the implementation; the sensor network ad hoc protocol platform (SNAPP), which defines the communication protocols available and their link to the SNSP; and the sensor network implementation platform (SNIP), which contains the physical nodes which implement the network. Nuzzo [2014] uses a different set of platforms: the top-level requirements platform, which sets out the key constraints and system behaviours; the static/non-functional model platform, used for architecture optimization; the discrete event platform, used for controller synthesis; and the continuous time/hybrid model platform, used for simulated-based verification and sizing of components. This closely resembles the ideal platform-based design flow for the oil system case study discussed in this paper (see Figure 4). Note that the method and results presented in Sections 3 to 4 are just concerned with the upper architecture optimization platform. However, in a full PBD flow the resulting architecture would be used for controller design and high-fidelity simulation at the lower levels.

2.5. Architectural Drivers / Decision Criteria

Whatever systems architecting approach is taken, a key task is to identify the *architectural drivers*. Fröberg [2013] presents a 5-step method for identifying the architectural drivers by analysing and refining stakeholder requirements. These drivers are the motivating features of an architecture which correspond to either the constraints or decision criteria in more formal optimization-based design. For example, in Bajaj [2015] the architectural drivers are cost and complexity (decision criteria in the objective function) and reliability (one of the constraints).

In the case of the oil system architecture problem presented in Section 3 to 4, the architectural drivers/decision criteria are:

1. Increasing controllability of oil flow to the individual oil chambers.

2. Minimising system architecture production cost.

These are both handled in the objective function described in Section 3.4. Minimising cost is common in almost all applications. The meaning of *increasing controllability* is less clear since the term controllability has many definitions. Some discussion of this is given by Skogestad [2005] who notes that often the term controllability is used to mean state-controllability (the ability to move a system from an initial state to an arbitrary point in the state space in finite time). If we consider the exit oil temperature at each of the oil chambers as states in our system, then the ability to arbitrarily move to any point in the state space requires uniquely controllable flow to each chamber. This would require a unique valve for each oil chamber. In this paper the term controllability relates more closely to (input-output) controllability which is linked to *performance* [Skogestad, 2005]. In the case of the oil system, good performance can be achieved when the flow to oil chambers can be controlled to manage oil temperature peaks during transients, without the need to oversupply oil during steady-state conditions. Therefore if two oil chambers share similar oil flow requirements, it may be possible to get good input-output controllability (good performance) without having full state-controllability. This would allow the production cost of the oil system to be reduced by using fewer valves.

3. Problem Formulation

This section formulates the turbofan oil system architecture optimization problem mathematically. The approach is based around the method presented by Bajaj [2015] whereby the architecture is described as a graph of nodes and edges. The resulting architecture is generated from a given template of nodes, with optimization used to determine which nodes to include and the interconnection structure between them. The novelty of this research and the main differences from the approach in Bajaj [2015] are:

1. **Use of a multi-criteria objective function** - this facilitates tradeoffs between the different objectives via selection of weights. The approach in this paper covers 2 criteria (cost and controllability) but can be easily extended to include others. The need for this arises from the fact that controllability cannot be handled as a constraint as reliability is handled in Bajaj. This is because there is no"necessary limit" for controllability since none (direct drive) or full (individual valve for each chamber) could both be acceptable depending on the priorities of the customer.

2. **Application to a new real-world problem** - increased controllability of oil flow leads to improved lubrication efficiency meaning reduced friction and decreased fuel consumption. Better management of temperature transients also improves the life of components and oil, leading to maintenance cost savings. However, there is also a strong pressure on engine manufacturers to keep the production costs low by minimising the number of additional components. This paper presents a new approach for handling these conflicting concerns.

Note that this approach, like Bajaj [2015], is an example of a *connecting* architectural decision making process

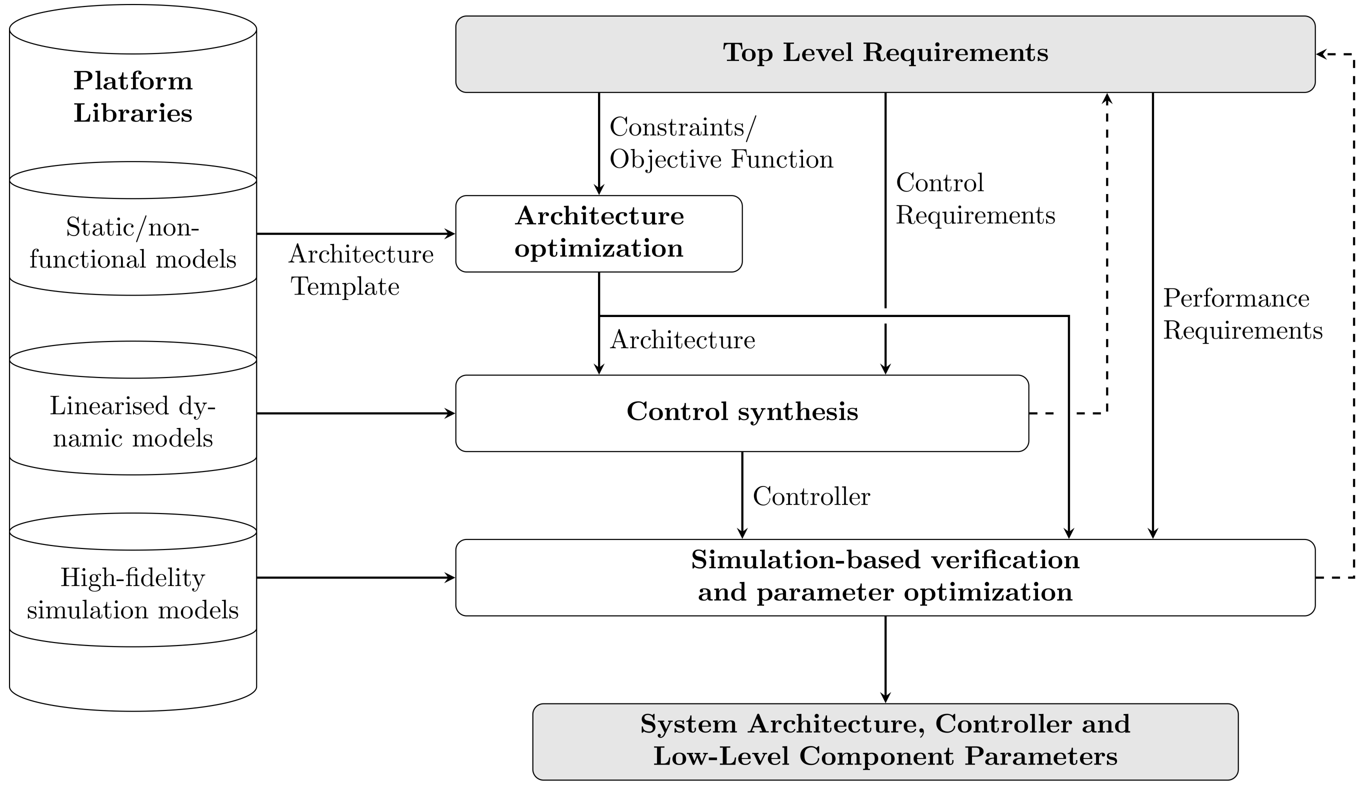


Figure 4: A platform-based design flow for the turbofan oil system. Cyclindrical buckets represent libraries of components/models at different levels of fidelity, compiled from bottom-up abstraction. Grey rectangles represent inputs and outputs of the top-down design flow. White rectangles represent design activities at the different platforms. Black arrows indicate information flows. Dashed arrows indicate a change in the requirements for upper platforms when no feasible solution can be found at a lower level.

[Selva, 2017]. However, it is also similar to a *downselecting* process due to the cost/controllability tradeoff leading to architectural solutions which are a subset of the original architecture template [Selva, 2017].

Section 3.1 defines the components and architecture template of the actively controlled oil system architecture. Section 3.2 discusses a heuristic approach to quantifying the similarities between different oil chamber flow requirements. Finally Sections 3.3 to 3.4 present the constraints and objective function for the optimization.

3.1. An Actively Controlled Oil System

Whilst geared turbofan designs motivate the architecture optimization techniques presented in this paper, the proposed methods are being validated against a baseline design of a conventional 3-shaft turbofan engine. The key components which make up a typical turbofan oil system architecture are outlined in Rolls-Royce [2005]. These include: tanks for storing oil; mechanically driven pumps for moving oil around the system; filters for removing debris from the oil; heat exchangers for removing heat from the oil; pipes for directing oil flow around the system; flow restrictors for changing the velocity and pressure of oil flows around the system; oil chambers with jets directing flow to bearings or gears; and dearators/breathers to vent air to the atmosphere. The main differences with an actively controlled architecture are the addition of variable restrictor valves and electrically driven pumps. These modifications allow the oil flow to be controlled independently of the engine shaft speed.

Following the approach taken by [Bajaj, 2015; Nuzzo, 2014] the oil system architecture is expressed as a graph with nodes $\{N_1, \cdots, N_n\} \in \mathcal{N}$ where $\mathcal{N}$ is partitioned into subsets $\{T, FP, HE, V, OC, SP\}$ corresponding to the 6 component groups outlined in Table 1. The interconnection matrix E is defined as in equation (1). The architecture template is given in Figure 2. In this template the connections between the tank-pumps and oil chambers-scavenge pumps are fixed, i.e. $e_{T,FP} = 1$ and $e_{OC_i,SP_i} = 1$, $e_{SP_i,FP} = 1$, $\forall \ i = \{1, \cdots, 7\}$.

3.1.1. Assumptions

The following assumptions have been made in the formulation of the problem:

- Some components such as filters and the breather are essential in any architecture and therefore these are taken out of this architecture optimization for simplicity. The remaining components which are considered in this problem are given in Table 1.

- Oil connections to oil chambers are parallel.

Table 1: The component groups, functions and maximum numbers of instances.

Component	Function	No.
Tank	Contain oil	1
Feed pump	Supply oil to the oil chambers	1
Heat exchanger	Remove heat from oil	4
Valve	Control the oil flows to the individual oil chambers	7
Oil chambers	Supply oil to engine bearings or gears	7
Scavenge pumps	Remove oil from oil chamber sumps	7

- Component sizes are fixed. Architectures are composed by connecting components according to rules defined in Section 3.3. Some components from the template may not be used in a given architecture.

3.2. Quantifying Similarities Between Oil Chamber Flow Requirements

The location of the seven oil chambers is based on a typical 3-shaft turbofan engine as outlined in Rolls-Royce [2005] and shown in Figure 5. As mentioned previously, the motivation for using an oil system architecture with valves is to better control the flow of oil to the individual oil chambers. There is also a need to keep the production costs and complexity of the architecture low. Therefore it is desirable to control multiple oil chambers with a single valve when their flow requirements are similar throughout the flight cycle.

The similarities between the oil flow requirements are contained in a matrix $C_{fr} \in S\mathbb{R}^{m\times m}$, where $S\mathbb{R}^{m\times m}$ is the set of real valued symmetric matrices of size $m \times m$ and m is the number of oil chambers.

$$C_{fr} := \begin{bmatrix} 0 & c_{1,2} & c_{1,3} & \cdots & c_{1,m} \\ c_{2,1} & 0 & c_{2,3} & \cdots & c_{2,m} \\ c_{3,1} & c_{3,2} & 0 & \cdots & c_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & c_{m,3} & \cdots & 0 \end{bmatrix}. \quad (2)$$

The elements $c_{i,j}$ are a measure of the independence of oil chambers i and j with a larger magnitude indicating a greater need to control their oil flows separately. The values of $c_{i,j}$ can be assigned through a variety of methods. For example, if there was a set of optimized flow conditions for each oil chamber over the flight cycle, these could be analysed to determine the statistical correlation between chambers. These correlations would be used to populate the matrix C_{fr}.

In the absence of these optimized flow conditions, a more heuristic approach has to be taken, as outlined in

Algorithm 1 Defining $c_{i,j}$ values

```
1: for all c_{i,j} do
2:   if i == j then
3:     Set c_{i,j} = 0
4:   else
5:     Set c_{i,j} = 1
6:     if i and j are in different parts of the engine
       (compressor/turbine/gearboxes) then
7:       c_{i,j} = c_{i,j} + 1
8:     end if
9:     for each shaft (HP/IP/LP) in i and j do
10:      if shaft is unique to either chamber i or chamber j then
11:        c_{i,j} = c_{i,j} + 1
12:      end if
13:    end for
14:  end if
15: end for
```

Algorithm 1. Lines 2 to 5 correspond to the fact that there will be at least some difference between oil flow requirements in different chambers. Lines 6 to 8 come from the fact that oil chambers are more likely to have similar flow requirements to other chambers in the same engine region, due to coupled temperature transients, pressures and flow rates. Lines 9 to 13 correspond to the fact that any oil chamber bearings or gears which do not have a physical connection can rotate at independent speeds and hence their optimum oil flows may vary more greatly.

Using Algorithm 1 for the 3-shaft civil turbofan example in Figure 5, C_{fr} evaluates to:

$$C_{fr} = \begin{bmatrix} 0 & 1 & 4 & 4 & 3 & 5 & 5 \\ 1 & 0 & 4 & 4 & 3 & 5 & 5 \\ 4 & 4 & 0 & 3 & 4 & 2 & 2 \\ 4 & 4 & 3 & 0 & 4 & 3 & 3 \\ 3 & 3 & 4 & 4 & 0 & 4 & 4 \\ 5 & 5 & 2 & 3 & 4 & 0 & 1 \\ 5 & 5 & 2 & 3 & 4 & 1 & 0 \end{bmatrix}. \quad (3)$$

Note that Algorithm 1 is a suggested set of rules for determining similarities, but some designers may be interested in other factors. For example, oil chambers with different bearing types (ball or roller) may have less similarity between flow requirements. Additionally, a designer may want to single out a specific oil chamber (OC i) to have an independent valve based on some experience or knowledge about particular oil flow challenges in that chamber. This could be achieved by adding a large number to the off-diagonal elements in the i^{th} row and column.

The multi-criteria optimization approach presented in this paper will work regardless of the method in which C_{fr} is populated. However, since the similarities matrix is used to generate the controllability objective scores, a sensible choice of C_{fr} values will be required to produce

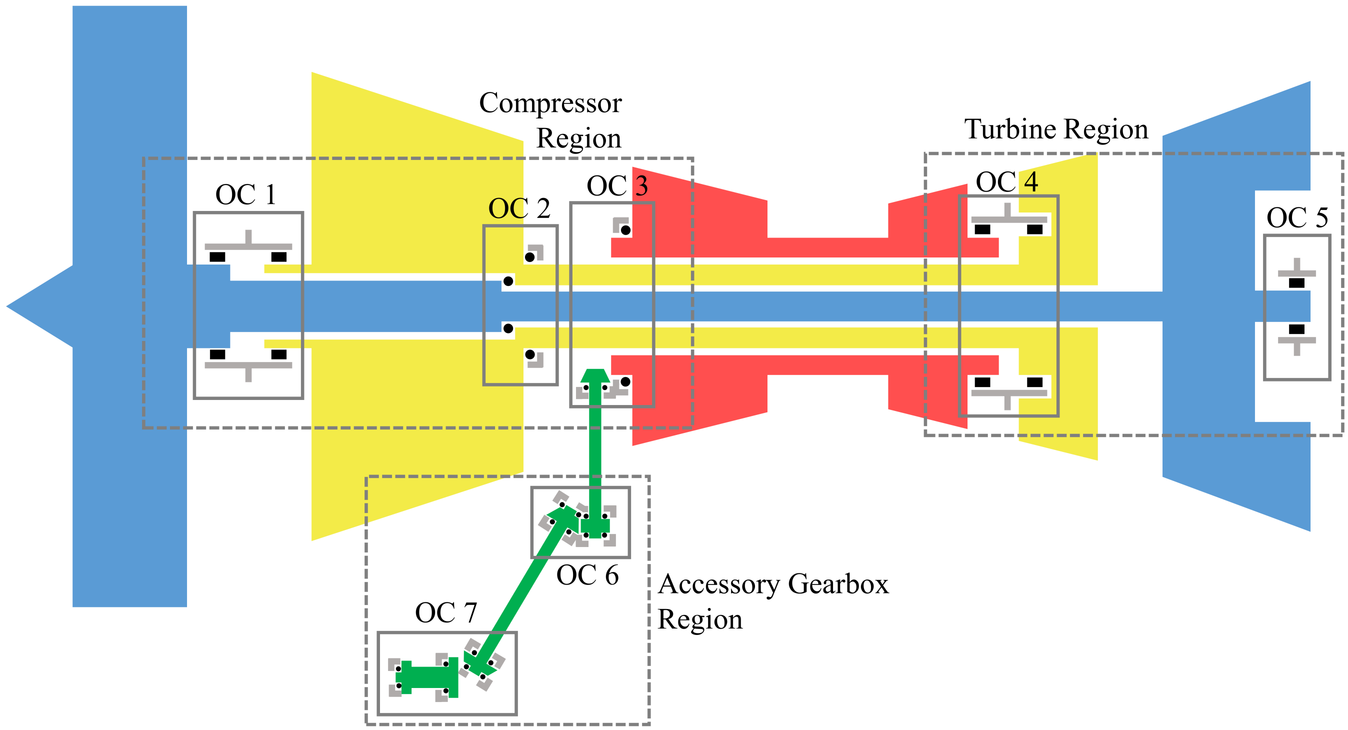


Figure 5: A schematic of a 3-shaft turbofan engine with low pressure shaft (blue), intermediate pressure shaft (yellow), high pressure shaft (red) and internal/step-aside/accessory gearboxes (green). Roller bearings (black rectangles) and ball bearings (black circles) are contained in the 7 oil chambers OC 1 to OC 7 (grey boxes). The engine regions (compressor, turbine, accessory gearbox) are highlighted by the dashed, grey boxes.

sensible architectures.

3.3. Defining Architecture Constraints

Constraints are introduced to the architecture optimization to ensure system requirements are met. These requirements may define either required/forbidden interconnections or some sort of energy balance that must be satisfied.

3.3.1. *Interconnection Constraints*

There are a variety of interconnection constraints which can be expressed formally as:

$$\sum_{i=1}^{|G1|} e_{G1_i,G2_j} \diamond c \quad \forall j \in \{1, \cdots, |G2|\}, c \in \mathbb{N}, \tag{4}$$

where $G1, G2 \in \{T, FP, HE, V, OC, SP\}$ are component partitions and $\diamond \in \{<, \leq, >, \geq, =\}$. For example, the requirement "each oil chamber shall be connected to exactly one valve" is defined as:

$$\sum_{i=1}^{|V|} e_{V_i,OC_j} = 1 \quad \forall j \in \{1, \cdots, |OC|\}. \tag{5}$$

For some component groups there may be constraints on connections with upstream components, depending on the connections made downstream. These can be expressed formally as:

$$\left(\sum_{i=1}^{|G1|} e_{G1_i,G2_j} \diamond c \right) \implies \left(\sum_{k=1}^{|G3|} e_{G3_k,G2_j} \diamond c \right), \quad \forall j \in \{1, \cdots, |G2|\}, \tag{6}$$

where $A \implies B$ indicates A *implies* B. For example, the constraint "if a valve is connected to one or more oil chambers, it shall also be connected to exactly one heat exchanger" is given by:

$$\left(\sum_{i=1}^{|OC|} e_{OC_i,V_j} > 0 \right) \implies \left(\sum_{k=1}^{|HE|} e_{HE_k,V_j} = 1 \right), \quad \forall j \in \{1, \cdots, |V|\}. \tag{7}$$

Likewise "if a heat exchanger is connected to a valve, it must also be connected to the feed pump" is expressed as:

$$\left(\sum_{i=1}^{|V|} e_{V_i,HE_j} > 0 \right) \implies \left(e_{FP,HE_j} = 1 \right), \quad \forall j \in \{1, \cdots, |HE|\}. \tag{8}$$

These interconnection requirements are contained in the constraint set R_I.

3.3.2. *Energy Balance Constraints*

In the component library there are 4 different off-the-shelf heat exchangers each with different maximum flow rates (in arbitrary units) contained in vector flow_{HE}.

$$\text{flow}_{HE} = \begin{bmatrix} 300 & 200 & 200 & 500 \end{bmatrix}. \tag{9}$$

The oil chambers have maximum flow demands given by:

$$\text{flow}_{OC} = \begin{bmatrix} 30 & 40 & 50 & 100 & 80 & 20 & 20 \end{bmatrix}. \tag{10}$$

One or more heat exchangers can be used in the architecture, so long as they meet the downstream maximum flow demand of the oil chambers. This is termed R_B, an energy balance constraint as in Bajaj [2015], and is defined formally as:

$$\sum_{i=1}^{|V|}\sum_{j=1}^{|OC|}(e_{HE_k,V_i})(e_{V_i,OC_j})(\text{flow}_{OC_j}) \leq \text{flow}_{HE_k}, \quad \forall k = \{1, \cdots, |HE|\}. \tag{11}$$

In this research, the interconnection and energy requirements have been manually converted from natural language to formal, programmable constraints. There is potential for a tool which allows requirements to be defined using a limited set of natural language expressions which are then automatically coded to formal requirements for the optimization problem. The challenges here revolve around getting a set of expressions which is large enough to capture any requirement that the user may wish to specify.

3.3.3. *Safety Constraints*

A key constraint for a controlled oil system is safety. If any valves become blocked leading to an interrupt in oil flow there could be serious consequences. It is assumed here that appropriate safety measures are incorporated into the physical design of the valves. For example, they could be sized to ensure that the minimum oil flow rate is always maintained, with just the upper range of flow controlled to optimize flow.

Since these safety concerns relate to the design of the valves themselves rather than the system architecture, they are not incorporated into the optimization algorithm for the oil system. In other applications, such as those whereby redundant components need to be used to achieve a certain level of reliability, safety constraints will need to be programmed into the architecture optimization as presented in [Bajaj, 2015; Nuzzo, 2014].

3.4. Objective Function

As previously noted, there are two decision criteria in the objective function: cost and controllability.

$$f := w_{\text{cost}} f_{\text{cost}} + w_{\text{control}} f_{\text{control}}. \tag{12}$$

Since these are two opposing objectives, the tradeoff between them is handled through the introduction of weights w_{cost} and w_{control}. Section 4.2 investigates the effect of varying the weights on the resulting architecture. This section shows how the individual objective functions f_{cost} and f_{control} are constructed.

3.4.1. *Cost*

This is dependent on the production cost of the valves and heat exchangers which are used in the architecture and their interconnections:

$$f_{\text{cost}} := \sum_{i=1}^{|V|}\delta_{V_i}\left(C_{V_{\text{base}}} + \sum_{j=1}^{|OC|} e_{V_i,OC_j}\text{flow}_{OC_j} C_{V_{\text{add.}}}\right), + \sum_{i=1}^{|HE|}\delta_{HE_i} C_{HE_i}, \tag{13}$$

where,

$$\delta_i := \begin{cases} 1 & \text{if } \sum_{j=1}^{|\mathcal{N}|} e_{i,j} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

The production costs of the four potential heat exchangers (in some monetary unit) are contained in vector C_{HE}. This represents the fact that different off-the-shelf components will utilise different technologies and hence cost different amounts. There is also a rough correspondence between these costs and the flow capacities contained in (9).

$$C_{HE} = \begin{bmatrix} 4000 & 3000 & 3000 & 10000 \end{bmatrix}. \tag{15}$$

In this paper the valves are all assumed to be equal cost ($C_{V_{\text{base}}} = 5000$) which represents the basic cost of manufacturing a valve regardless of size. It is also assumed that there is an additional cost added for each oil chamber that is connected to a valve depending on the size of the maximum flow requirements to that chamber ($\text{flow}_{OC_i} \cdot C_{V_{\text{add.}}}$) where $C_{V_{\text{add.}}} = 100$. This represents the additional material cost in larger valves with greater flow capacity. Using this cost model, the cost of the valve part of the architecture is 69,000 units for a 7 valve system, and 39,000 units for a 1 valve system. This confirms that the more valves used, the higher the cost.

3.4.2. *Controllability*

The effect of the combination of operations in equation (16) is to extract and sum the relevant values from the flow interconnections matrix C_{fr} based on which oil chambers are controlled via the same valves.

$$f_{\text{control}} := \sum_{i=1}^{|V|} e_{v_i,OC} \left(\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \otimes e_{v_i,OC} \right) \bullet C_{fr}, \quad (16)$$

where $\otimes$ denotes the Kronecker product and $\bullet$ denotes the Hadamard product.

Consider an example with 4 oil chambers and 4 potential valves given in (17). In this example the first three oil chambers are controlled by one valve and the last oil chamber by another separate valve as indicated in $e_{V,OC}$.

$$e_{V,OC} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_{fr} = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{4} & 4 \\ \mathbf{1} & 0 & \mathbf{4} & 3 \\ \mathbf{4} & \mathbf{4} & 0 & 2 \\ 4 & 3 & 2 & \mathbf{0} \end{bmatrix}. \quad (17)$$

Evaluating f_{control} using equation (16) gives a sum of the bold values in C_{fr}:

$$f_{\text{control}} = (1 + 4 + 1 + 4 + 4 + 4 + 0) = 18. \quad (18)$$

Note that according to this equation, a higher value indicates *worse* controllability, so the objective is to minimise f_{control}.

3.4.3. *Normalization*

As shown in Sections 3.4.1 and 3.4.2, the criteria do not share the same units. Therefore they are normalized by dividing each criterion score by the maximum possible value for that criterion:

$$\hat{f}_{\text{cost}} = \frac{f_{\text{cost}}}{f_{\text{cost}_{\max}}}, \quad \hat{f}_{\text{control}} = \frac{f_{\text{control}}}{f_{\text{control}_{\max}}}. \quad (19)$$

The maximum value for the cost objective was found to be $f_{\text{cost}_{\max}} = 79,000$, when using all 7 valves and the most expensive heat exchanger (HE 4). The maximum value for the controllability objective was found to be $f_{\text{control}_{\max}} = 146$, when controlling all 7 oil chambers with a single valve. These normalized criteria are then used in the overall objective function as defined in equation (20).

4. Results

After defining the constraints R and objective function f the optimization problem is expressed as:

$$\begin{aligned} \min_{E \in \mathbb{B}^{n \times n}} \quad & f := w_{\text{cost}} \hat{f}_{\text{cost}} + w_{\text{control}} \hat{f}_{\text{control}}, \\ \text{subject to} \quad & R := \{R_I, R_B\}. \end{aligned} \quad (20)$$

Since the matrix variable E only contains values in the Boolean set $\mathbb{B} := \{1, 0\}$ this is a specific type of *integer program*. This has been solved using the MATLAB toolbox YALMIP [Löfberg, 2004] implementing a global branch-and-bound algorithm with upper solver FMINCON [The MathWorks Inc., 2015] and lower solver GUROBI [Gurobi Optimization Inc., 2016].

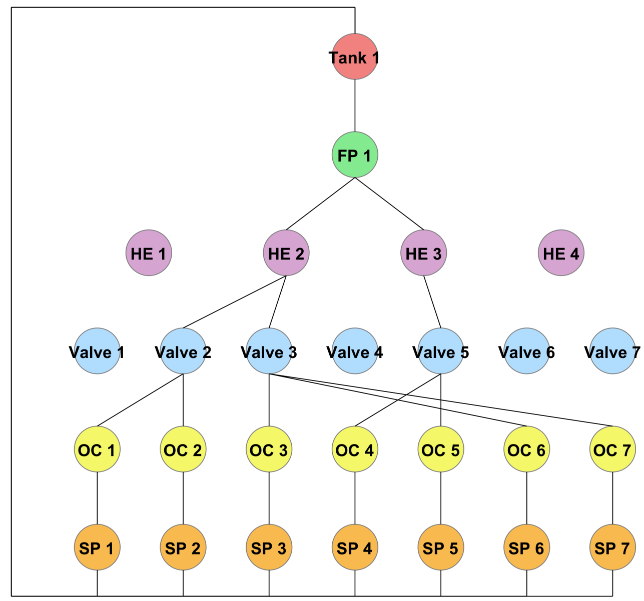


Figure 6: An example oil system architecture with 3-valves and two heat exchangers. The tank, fuel pump, oil chambers, scavenge pumps and their connections are fixed by the architecture template.

4.1. A 3-Valve Architecture Solution

The resulting architecture depends on the selection of weights w_{cost} and w_{control} as discussed in Section 4.2. Selecting a weight ratio that produces a 3-valve system results in the architecture shown in Figure 6.

There are two things to note here. Firstly the architecture contains two heat exchangers. Whilst the fourth heat exchanger has a flow capacity large enough to supply all of the oil chambers it is also more expensive (see (15)). Therefore the algorithm has chosen to use two cheaper heat exchangers (2 and 3). This was the case for the entire range of criteria weightings investigated in Section 4.2, apart from the architectures with a single valve. There is a requirement that "if a valve is connected to an oil chamber it shall be connected to *exactly* one heat exchanger". Therefore when there is a single valve controlling flow to all oil chambers only one heat exchanger can be used and heat exchanger 4 is the only one with sufficient capacity. A cheaper solution could be gained by allowing connection of multiple heat exchangers to a single valve in parallel. This has not been implemented because the physics of mixing multiple oil flows at the inlet to the valves would make it hard to quantify the state of the oil (e.g. temperature or viscosity) which is required for effective control. This

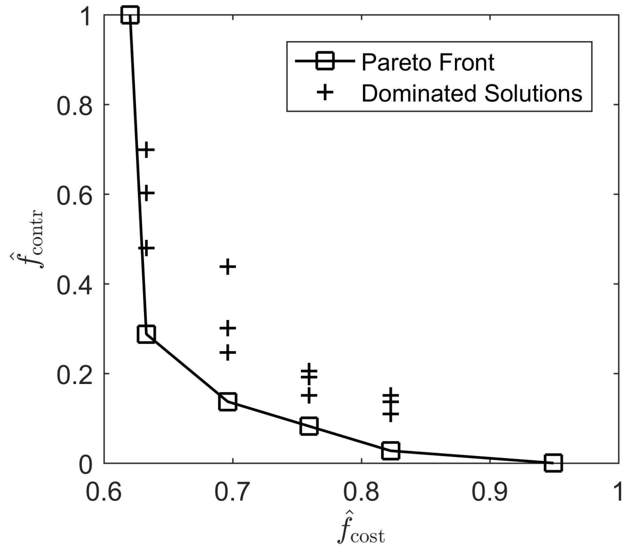


Figure 7: Architecture solutions generated via the optimization approach with the goal of minimising both criteria (squares). Moving from left to right on the x-axis, these represent solutions with 1, 2, 3, 4, 5 and 7 valves. Note that the optimization algorithm finds nondominated solutions on the Pareto front, meaning that improvement in one criterion cannot be achieved without producing a worse score for the other criterion. A few solutions have been generated randomly, without taking into account the objective function, to show the principle of dominated solutions (crosses).

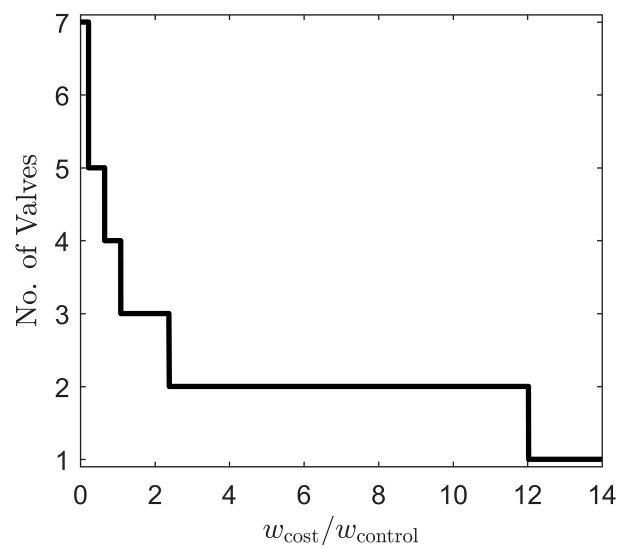


Figure 8: The effect of varying the cost to controllability weight ratio on the number of valves in the resulting architecture.

constraint only has a small effect on the overall size of the search space, since architectures with two or more valves are not constrained to only using 1 heat exchanger (as shown in Figure 6).

The second point to note is that this is a sensible coupling of the oil chambers for a 3-valve system. Referring back to Figure 5 it is clear that the two LP/IP compressor chambers are controlled by the first valve, the two gearbox chambers and the gearbox/HP compressor chamber are controlled by the second valve and the two turbine chambers are controlled by the third valve. This demonstrates the effectiveness of the approach presented in this paper.

4.2. Investigating the Trade-offs Between Cost and Controllability

The solution presented in Section 4.1 results in a 3-valve system. This is just one of a number of optimal solutions on the Pareto front, as shown in Figure 7. Note that there is a clear tradeoff between these criteria: improvement can only be achieved for controllability by increasing the cost and vice versa. All of the architectures shown in Figure 7 have some form of active control (1 valve or more), but none of the oil system architectures reviewed in the literature have controllable oil flows [Linke-Diesinger, 2008; Rolls-Royce, 2005]. This means they are cheaper to produce but have no controllability. Hence they would appear beyond the top-left corner of this tradeoff plot.

The solution from the Pareto front that is generated by the optimization will vary depending on the values of the weights w_{cost} and w_{control} in the objective function (20). Some discussion of how to choose weights is given by Shukla [2016]. In particular it presents a method for determining overall criteria weights from a set of weights given by multiple stakeholders. For this research, there is no access to multiple stakeholders to implement such a method. However, since the number of decision criteria is small it is possible to investigate the entire range of weight ratios $w_{\mathrm{cost}}/w_{\mathrm{control}}$ that produce architectures with 1 to 7 valves. This tradespace is represented in Figure 8. Since there are only 7 discrete possibilities for the number of valves in the architecture, the plot in Figure 8 shows a stepped line. One thing to note is the fact that there is a jump from a 7-valve architecture to a 5-valve architecture. The reason for this is clear when referring back to the matrix C_{f_r}.

$$C_{fr} = \begin{bmatrix} 0 & \mathbf{1} & 4 & 4 & 3 & 5 & 5 \\ \mathbf{1} & 0 & 4 & 4 & 3 & 5 & 5 \\ 4 & 4 & 0 & 3 & 4 & 2 & 2 \\ 4 & 4 & 3 & 0 & 4 & 3 & 3 \\ 3 & 3 & 4 & 4 & 0 & 4 & 4 \\ 5 & 5 & 2 & 3 & 4 & 0 & \mathbf{1} \\ 5 & 5 & 2 & 3 & 4 & \mathbf{1} & 0 \end{bmatrix}. \tag{21}$$

Note the 1s highlighted in bold in the top-left and bottom-right corners. The similarities between the flow requirements of OC 1 and OC 2 are identical to the similarities between the flow requirements of OC 6 and OC 7. Therefore as soon as the weight ratio $w_{\mathrm{cost}}/w_{\mathrm{control}}$ is great enough that it is worth controlling OC 1 and OC 2 with a single valve, the same is true for OC 6 and OC 7. This explains why the optimization never produces a 6-valve architecture.

Another observation is the fact that the stepped line shows a roughly exponential decrease. The reason for this can be explained through considering the effect of removing valves from the system on the values of the objective functions f_{cost} and f_{control}. Moving from a 7 to 6 valve architecture there is a decrease in the f_{cost} due to the removal of 1 valve. However, as two oil chambers become controlled by a single valve they *both* suffer a reduction in controllability. Similarly when moving from the 6-valve architecture to a 5-valve architecture f_{cost} continues to decrease in a linear fashion, whilst the controllability of all three oil chambers goes down. Since f_{control} decreases more rapidly than the reduction in f_{cost}, the weight ratio $w_{\text{cost}}/w_{\text{control}}$ has to increase exponentially to produce architectures with the smallest number of valves.

A sensitivity analysis has also been performed on the parameters of the cost model in equation (13). The net effect of increasing either $C_{V_{\text{base}}}$ or $C_{V_{\text{add}}}$ is that a smaller $w_{\text{cost}}/w_{\text{control}}$ ratio is needed to generate an architecture with the same number of valves. However, the pattern of the exponential stepped decrease shown in Figure 8 remains the same.

For simplicity this research has only considered the two decision criteria of cost and controllability. This allows a 2-dimensional plot to be used to visualise the tradespace. However, a more thorough optimization could consider other criteria such as weight, safety or reliability. In this case, a multi-criteria visualisation tool such as parallel coordinates [Fleming, 2005] would be needed to investigate the effects of varying the criteria weightings.

5. Conclusion and Future Research

This paper has presented a multi-criteria optimization approach to the design of high-level turbofan oil system architectures. A key development is the ability to analyse the impact of using common actuators for multiple oil chambers on the controllability and cost of the system. This has been achieved through use of a flow requirement similarities matrix which is used to identify which oil chambers should or should not be controlled together. In this research the matrix has been populated through use of a heuristic algorithm but the optimization framework would remain valid if the matrix was populated using other methods. The approach has produced sensible results and has demonstrated the ability for tradeoffs to be investigated through variation of weights in the objective function. The optimization yields more suitable architectures than other computational methods investigated, such as a random coupling of oil chambers to valves. The architectures generated also match with the best architectures determined subjectively by experienced engineers. This supports the method used and provides an additional objective evidence-base upon which to make decisions.

The techniques developed have been validated on a baseline 3-shaft turbofan oil system design. This motivates the use of the approach for future geared turbofan oil system designs. Other potential case studies are alternative controlled flow networks such as smart building water-heating control or smart traffic systems.

The optimization-based approach presented in this paper ensures that designs are verified and guaranteed to satisfy the formal specification. However, it is worth noting that there is a human element to formulating the problem in the choice of constraints and objective function. This means the resulting architecture will be sensitive to the problem formulation choices made by the system designers. Therefore a procedure for validation of the specification would also be required when using these techniques in practice.

This paper has only considered the high-level architecture optimization stage in a multi-platform design flow (see Figure 4). Future research will consider the lower levels in this framework such as control synthesis and parametrization of high-fidelity component models. The architectures produced should be suitable for implementation with any control strategy. However, intelligent choices of controller may provide opportunities to identify limitations of a given architecture, producing a feedback loop and reformulation of the architecture problem (see Figure 3). This idea will be explored further in future research.

Acknowledgment

The authors would like to thank Lixin Ren and Derek Wall of Rolls-Royce® for useful discussions and information. The authors also gratefully acknowledge the funding from the University of Sheffield Prize Scholarship which has made this work possible.

References

A. Agarwal, G. L. Hamza-Lup, and T. M. Khoshgoftaar, A system-level modeling methodology for performance-driven component selection in multicore architectures, IEEE Systems Journal 6(2) (2012), 317–328.

N. Albarello and J.-B. Welcomme, A model-based method for the generation and optimization of complex systems architectures, in IEEE 6th Annual International Systems Conference, 2012, pp. 1–6.

N. Bajaj, P. Nuzzo, M. Masin, and A. L. Sangiovanni-Vincentelli, Optimized selection of reliable and cost-effective cyber-physical system architectures, in Design, Automation & Test in Europe Conference & Exhibition, 2015, pp. 561–566.

C. Calvert, G. L. Hamza-Lup, A. Agarwal, and B. Alhalabi, An integrated component selection framework for system-level design, in IEEE 5th Annual International Systems Conference, 2011, pp. 261–266.

W. Chapman, J. Rozenblit, and A. Bahill, System design is an NP-complete problem, Systems Engineering 4(3) (2001), 222–229.

E. Crawley, B. Cameron, and D. Selva, System architecture: Strategy and product development for complex systems, Prentice Hall, Hoboken, NJ, 2015.

D. Dori, Object-Process Methodology: A Holistic Systems Paradigm, Springer-Verlag, Berlin, 2002.

J. P. Elm, D. R. Goldenson, K. E. Emam, N. Donitelli, and A. Neisa (NDIA SE Effectiveness Committee), A survey of systems engineering effectiveness - Initial results (with detailed survey response data), Tech. rep., Software Engineering Institute, Carnegie Mellon University, 2008.

J. Finn, P. Nuzzo, and A. L. Sangiovanni-Vincentelli, A mixed discrete-continuous optimization scheme for cyber-physical system architecture exploration, in 2015 IEEE/ACM International Conference on Computer-Aided Design, 2015, pp. 216–223.

D. G. Firesmith, P. Capell, D. Falkenthal, C. B. Hammons, D. T. Latimer IV, and T. Merendino, The method framework for engineering system architectures, CRC Press, Boca Raton, FL, 2008.

P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, Many-objective optimization: An engineering design perspective, Evolutionary Multi-Criterion Optimization LNCS 3410 (2005), 14–32.

J. Fröberg, S. Larsson, S. Dersten, and P. Å. Nordlander, Defining a method for identifying architectural candidates as part of engineering a system architecture, in IEEE 8th Annual International Systems Conference, 2014, pp. 266–271.

J. Fröberg, S. Larsson, and P. Å. Nordlander, A method for analyzing architectural drivers when engineering a system architecture, in IEEE 7th Annual International Systems Conference, 2013, pp. 711–717.

D. R. Georgiadis, T. A. Mazzuchi, and S. Sarkani, Using multi criteria decision making in analysis of alternatives for selection of enabling technology, Systems Engineering 16(3) (2013), 287–303.

Gurobi Optimization Inc., Gurobi Optimizer 6.5, 2016.

H. Gustavsson and J. Axelsson, Architecting complex embedded systems: An industrial case study, in IEEE 5th Annual International Systems Conference, 2011, pp. 472–478.

O. Hammami, Architecture frameworks, multiobjective optimization and multiphysics simulation: Challenges and opportunities, in IEEE 9th Annual International Systems Conference, 2015, pp. 546–553.

O. Hammami and M. Houllier, Rationalizing approaches to multi-objective optimization in systems architecture design, in IEEE 8th Annual International Systems Conference, 2014, pp. 407–410.

INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 3, INCOSE, 2006.

D. G. Jansson, and S. M. Smith, Design fixation, Design Studies 12(1) (1991), 3–11.

L. Jones, Battle for the skies, Engineering & Technology 10(5) (2015), 72–75.

A. Linke-Diesinger, Systems of commercial turbofan engines: An introduction to systems functions, Springer-Verlag, Berlin, 2008.

J. Löfberg, YALMIP: a toolbox for modeling and optimization in MATLAB, in IEEE International Symposium on Computer Aided Control Systems Design, 2004, pp. 284–289.

S. W. Miller, T. W. Simpson, and M. A. Yukish, Design as a sequential decision process: a method for reducing design set space using models to bound objectives, ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2A: 41st Design Automation Conference, 2015.

P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donze, and S. A. Seshia, A contract-based methodology for aircraft electric power system design, IEEE Access 2 (2014), 1–25.

S. Pugh, Total design: integrated methods for successful product engineering, Addison-Wesley Publishing Company, Harlow, 1991.

S. J. Rekuc, J. M. Aughenbaugh, M. Bruns, and C. J. J. Paredis, Eliminating Design Alternatives Based on Imprecise Information, SAE International, 2006.

Rolls-Royce, The Jet Engine, Rolls-Royce plc, London, 2005.

A. L. Sangiovanni-Vincentelli, Quo vadis, SLD? Reasoning about the trends and challenges of system level design, Proceedings of the IEEE 95(3) (2007), 467–506.

A. L. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, Taming Dr. Frankenstein: Contract-based design for cyber-physical systems, European Journal of Control 18(3) (2012), 217–238.

D. Selva, B. Cameron, and E. Crawley, Patterns in system architecture decisions, Systems Engineering (Early View), 2017.

V. Shukla, G. Auriol, and K. W. Hipel, Multicriteria decision-making methodology for systems engineering, IEEE Systems Journal 10(1) (2016), 4–14.

S. Skogestad and I. Postlethwaite, Multivariable feedback control: Analysis and design, 2nd edition, Wiley, Chichester, 2005.

S. V. Subramanian and D. A. DeLaurentis, Application of multidisciplinary systems-of-systems optimization to an aircraft design problem, Systems Engineering 19(3) (2016), 235–251.

The MathWorks Inc., MATLAB and optimization toolbox release 2015b, 2015.

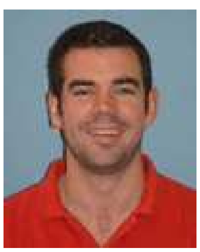
R. E. Thompson, J. M. Colombi, J. Black, and B. J. Ayres, Disaggregated space system concept optimization: Model-based conceptual design methods, Systems Engineering 18(6) (2015), 549–567.

A. Wichmann, S. Jäger, T. Jungebloud, R. Maschotta, and A. Zimmermann, System architecture optimization with runtime reconfiguration of simulation models, in IEEE 9th Annual International Systems Conference, 2015, pp. 660–667.

Christopher J. Hambley received an MEng degree in Systems and Control Engineering from the University of Sheffield in 2014. He is currently studying for a PhD in the Rolls-Royce Control and Monitoring Systems University Technology Centre, University of Sheffield. His research interests focus on applying formal methods and design automation techniques to reduce time-to-market and engineering effort in the development of industrial systems. Chris was awarded a University Prize Scholarship to fund his research.

Bryn Ll. Jones is a Lecturer in the Department of Automatic Control and Systems Engineering at the University of Sheffield, which he joined in 2011. He received his PhD from Imperial College London in 2010 and a MEng in Electrical and Information Sciences from the University of Cambridge in 2004. His research interests include the estimation and control of fluid flows and systems operating within fluid-flow environments. His research is focussed upon the development of efficient methods for applying control theoretic techniques to complex fluid mechanics-based systems, and is applied to a variety of problems in the Energy, Transport and Defence Sectors.

Ian Griffin received his PhD from the University of Sheffield in 2000 as well as MSc and BEng degrees from the same institution in 1997 and 1996. He is currently a Project Engineer for Thermo Fluid Systems at Rolls-Royce Plc. He joined the company in 2007 as a Control System Specialist and was promoted to Engineering Manager in 2010, leading a team of 23 Control System Architects operating across aerospace, marine and energy sectors. Prior to this he worked for 7 years as a Research Associate in the Rolls-Royce Control and Monitoring Systems University Technology Centre, University of Sheffield.



Andrew R Mills, CEng, PhD, is a Senior Researcher and Research Programme Manager at the University of Sheffield. He has worked in the defence industry on aerospace and automotive applications before his current post within the University Technology Centre supported by Rolls-Royce. Research interests are in a broad range of aspects related to control and health management systems including: signal processing algorithms, instrumentation, advanced processing technologies and model based design.

Visakan Kadirkamanathan obtained his B.A in Electrical and Information Sciences at Cambridge University Engineering Department and went on to complete his PhD in Information Engineering at the same institution. Following brief post-doctoral research positions at the Universities of Surrey and Cambridge, he was appointed Lecturer at the Department of Automatic Control & Systems Engineering at the University of Sheffield. He is currently Professor in Signal and Information Processing and is also the Director of the Rolls-Royce University Technology Centre for Control and Monitoring Systems Engineering at Sheffield, leading research activities in systems architectures, modeling and control and health monitoring of integrated power systems. His wider research interests include system identification and signal processing with applications. He has published more than 170 papers in peer reviewed journals and conferences. He was the recipient of the PNAS Cozzarelli Prize in 2013 for his seminal work on modeling the dynamics of conflicts. He is the Co-Editor-in-Chief of the International Journal of Systems Science.