INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

EXPLORING THE APPLICATION OF A MULTI-DOMAIN SIMULATION-BASED COMPUTATIONAL SYNTHESIS METHOD IN MEMS DESIGN

Francesca Bolognini[1], Ashwin A. Seshia[1] and Kristina Shea[2]
[1]University of Cambridge
[2]Technical University of Munich

ABSTRACT

Providing further computational support in the conceptual stages of the design process that focuse on design synthesis has been recognised as a key, yet difficult, research goal. Many theoretically successful computational synthesis methods have been developed to support different design domains, but the majority of these methods are developed for a single design domain, if not at a single design task, and rare are examples of multi-domain techniques, i.e. adaptable to different design domains. Straightforward adaptability of multi-domain synthesis methods creates instead the potential to provide customisable tools for automated design synthesis and optimisation. This paper presents the development of a multi-domain computational synthesis method and addresses issues related to generality and reliability. The method combines a generate-and test algorithm, called Burst, with an object-oriented, systems-based representation called Connected-Node System (CNS). The method has been successfully applied in the past to truss optimisation and bitmap synthesis and, in this work, is extended to the more complex domain of MEMS design synthesis. The first example presented is the automated synthesis and optimisation of a meandering microresonator considering two design criteria, device size and resonant frequency, and illustrates the basic capabilities of the method. The second example is a free-free beam microresonator that extends the automated MEMS modelling and simulation capabilities of the method to consider a realistic model such that the resulting solutions are relevant for MEMS designers. Both examples show how flexibility and generality of multi-domain methods are not necessarily an obstacle to finding precise and accurate solutions to complex design tasks, demonstrating that multi-domain computational synthesis methods create potential for use in everyday design practice to generate innovative solutions and explore design spaces.

Keywords: Computational synthesis, Generative design, Simulation-based design, Multiobjective design optimisation, MEMS

1 INTRODUCTION

Since the first CAD system in the 1960s computer tools have increasingly supported designers in a wide range of tasks within the design process and sophisticated tools for geometric modelling, analysis, simulation, and data management have become essential. The use of software support in the creative part of the design process, i.e. conceptual and embodiment design which focuses on synthesis of both functional solutions and product form, continues to be actively researched in order to advance method capabilities required for practical use and widespread take-up in industry. It is recognized by many articles that it is a difficult endeavour to create machines that can "think" and solve synthesis tasks in engineering [1]. This research area builds on fundamental work in artificial intelligence but is combined with the technical complexities found in product development within engineering [2]. The aim of computational synthesis tools is to assist and support designers by computationally describing and defining design spaces and performance criteria, rapidly generating design alternatives, searching for beneficial and optimised solutions, and generating new solutions that go beyond a designer's own insight and experience, thus promoting innovation. Benefits to the design process include reducing design time and costs, gaining an understanding of complex trade-offs among multiple performance criteria and their relation to product structure and form, and assisting decision-making. Many computational synthesis methods have been developed for different design domains, e.g. structural, mechanical and electrical design. In structures, topology synthesis has been investigated for many years and current methods use either continuous or discrete representations. A recent application of a discrete method is structural shape annealing, which combines a structural grammar with structural analysis and simulated annealing optimisation, and has been used to design a novel cantilever structure that has been built and permanently installed [3]. Starling [4] has automated the synthesis of gear systems using a parallel grammar based on a function-behaviour-structure representation and incorporating automated simulation to calculate behaviour. Lipson and Pollack achieved the automatic design and manufacture of robotic lifeforms by evolving electromechanical systems, controlling the generative process with a simple fitness function [5]. This is only a short overview of the many methods and applications within the area of computational synthesis. A complete description of the current state-of-the-art in formal design synthesis methods in a variety of domains can be found in Antonsson and Cagan [2, 6].

Current limitations of computational synthesis methods often include the necessity of supporting multiple design criteria and multidisciplinary considerations as well as the integration of automated simulation modelling to create simulation-driven synthesis tools. These limitations will be addressed in this paper. Further, in most situations, synthesis methods and tools have been developed with the aim of covering a specific design domain or even a single design case, making them not reusable for other design tasks. Koza [7] introduces the concept of ”routineness” as a way to assess synthesis techniques. A synthesis technique has a high degree of routineness if it is applicable to a wide range of problems, within a single domain and different domains, with minimal human effort required to adapt the method and implementation to new design scenarios. All the approaches mentioned above have been demonstrated to be successful on chosen theoretical benchmark tasks, but rare are the cases of multi-domain techniques, i.e. not oriented at a specific design domain. Campbell et al. [8] provide in GraphSynth a GUI-based graph grammar rule specification and a common engine for recognizing and applying rules to designs, represented by graphs, within a topology optimization process. Applications include MEMS devices and sheet metal parts. Rudolf et al. present a system called “Design Compiler 43” for creating and compiling engineering graph grammars, which can be linked to CAD models and FEM analysis tools. The system focuses on synthesis alone, providing a domain independent representation for conceptual design and has been applied to various aerospace applications [9].

The work described in this paper presents the development of a multi-domain method that contributes to knowledge in developing flexible and robust computational synthesis and optimisation methods, leading in the future to extended use of such tools in industrial applications. The modular and object-oriented approach adopted to make the method and its implementation flexible is explained. Two issues related to generality and lack of reliability of multi-domain methods are addressed. The first is concerned with the effectiveness of multi-domain synthesis techniques in complex design domains where it is essential to include sufficient details in both synthesis and optimisation models in order to generate practical results. The second deals with the perplexity of whether it is possible to obtain detailed and accurate solutions to design tasks with complex constraints and objectives. These two issues are here investigated through the application of the method to the complex design domain of microelectromechanical systems (MEMS). First an example of automated synthesis and optimisation of a meandering microresonator is given considering two design criteria, device size and resonant frequency, to illustrate the basic capabilities of the method. Next, a more complex example is presented for a free-free beam microresonator that extends the automated MEMS modelling and simulation capabilities to consider a realistic model such that the resulting solutions are relevant for MEMS designers. The paper finishes with conclusions and future work.

2 A MULTI-DOMAIN METHOD: CNS-BURST

This paper presents an object-oriented, simulation-based, computational synthesis method for structural, mechanical and mechatronic systems. A synthesis task is formulated as a design optimisation task consisting of design parameters, constraints and objectives. The method combines a multicriteria generate-and-test algorithm, called Burst, in conjunction with a general design representation called Connected Node Systems (CNS) to create the CNS-Burst method. The CNS representation is a generalised representation for interconnected systems. The main idea of the search method is to iteratively modify an initial design (represented by a connected-node system) using a library of modification operators that generate new solutions by combining basic building blocks and altering their connectivity and internal geometry. New designs are simulated and evaluated using defined objective functions and constraints and, when appropriate, placed in an archive of Pareto optimal solutions. The method has been successfully applied to different design domains [10] and tested against benchmark case studies.

The real strength of the method is its modular implementation. The architecture of a synthesis tool requires as necessary components a design representation, a generative mechanism to generate alternative designs, integrated evaluation of performance criteria, and a search method to find feasible and optimised design alternatives. In a multi-domain tool all these features must be conceived in a way such that they can be adapted to different design tasks. In the method introduced each of these fundamental components has been implemented as a stand-alone module, independent from the others. This allows for desired changes in each module without affecting the rest of the code, enabling also straightforward inclusion of new domain knowledge. The modules are linked together through the main algorithm that directs the search and calls the required different modules (Figure 1). Inputs required to start the search include:

- the maximum number of design evaluations
- an initial design,
- an optimisation model expressed in terms of minimising objectives and any required constraints.

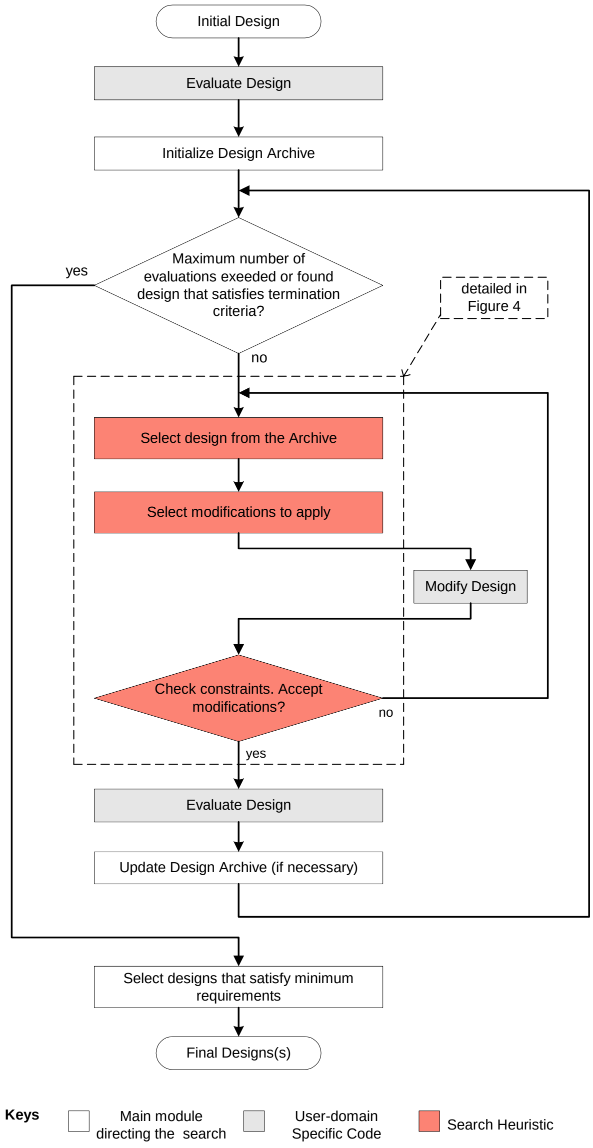


Figure 1. CNS-Burst method overview.

The first step of the search algorithm consists of validating (in terms of design constraints) and evaluating (in terms of design objectives) the initial design and creating an archive where the initial design and any other Pareto optimal design generated by the search will be stored. Next the main loop of the search method starts, and stops only if the maximum number of evaluations set initially has been reached.

The main module of the search is the 'search advisor' module (red steps in Figure 1), called Burst. This advisor acquires a design from the archive and selects modification operators to apply to this design. The design is then modified and validated. Each candidate design successfully generated by the design modifications is automatically modelled in the chosen simulation environment, where its behaviour is evaluated according to defined design objectives and constraints. Subsequently, evaluated designs are tested for inclusion in the design archive. Those that satisfy the Pareto criteria are stored to evolve a Pareto-optimal front of non-dominated solutions, according to all the solutions seen throughout the optimisation process. A solution is 'non-dominated' when, through pair wise comparison, it is superior to any other in the design archive for at least one objective [11]. The Pareto-optimal front develops throughout the synthesis and search process so that the outcome is a set of Pareto-optimal designs. This search loop will be repeated at each iteration until the maximum number of evaluations set initially is reached or the termination criteria are met. The method is entirely implemented in Matlab and is able to synthesise both 2D and 3D artefacts. Figure 2 shows a modular representation of the method where each module can be thought as a black box with its own inputs and outputs defined completely independent from the other modules. The picture also highlights the cascade order in which each module is executed during the search. In the following sub-sections a description of each module will be given. Further details of the method can be found in [10, 12, 13].

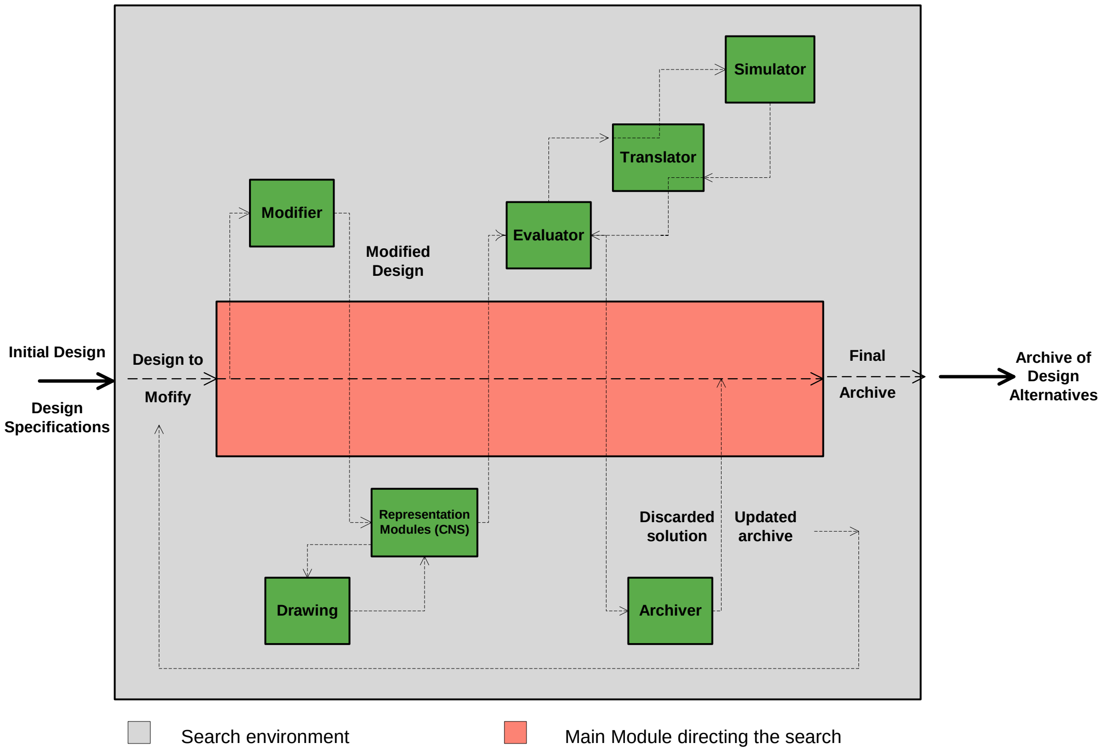


Figure 2. Modular structure of the method.

2.1 Representing design alternatives: Connected-Node System

The CNS design representation employs basic building blocks, called primitives, and nodes to build systems and subsystems of interconnected primitives. A system is represented as a graph that consists of nodes, primitives and subsystems (Figure 3). This representation is simple, well known in many design domains (for example circuit representation) and very effective for keeping the structure of the code modular. Each primitive can be thought of as an independent module and any new primitive can be independently implemented and “plugged-in” to the code. Primitives do not need to be of the same nature to be implemented, which offers the possibility to create a library of components to be used according to the design domain explored and offering the possibility to generate multidisciplinary designs, for example mechatronic devices. Nodes serve as connection points for primitives and subsystems that form the complete interconnected system. Nodes are defined by their position and degrees of freedom, i.e. floating or anchored nodes. Optional properties for the nodes are mass, voltage level, force applied to the node and possibility for interacting with primitives only, i.e. internal nodes, or with primitives and subsystems, i.e. called port-nodes. Primitives are basic elements that constitute a system, such as beams and resistors. Individual primitives within a connected-node system, such as the ones in Figure 3, are each instantiations of a specific primitive type.

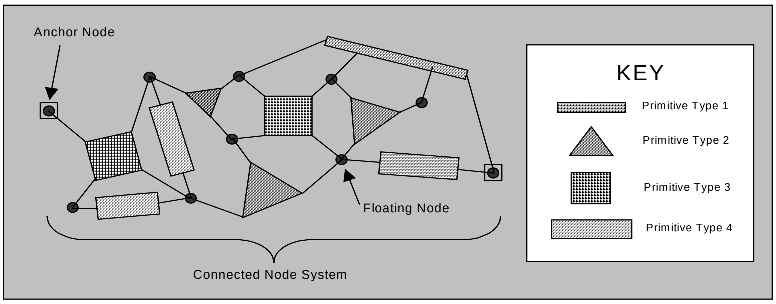


Figure 3. Example of connected-node system.

The essence of a primitive type is that it should be able to self-organise its internal structure and properties as a function of external specifications of:

- the nodes that the primitive instantiation is connected to in a system, and
- a list of primitive parameters (such as width for a beam, or resistance for a resistor), which can be either static in the synthesis process or variable.

Hard constraints for the primitive are defined within the primitive object class together with functions that check for violation of constraints specific to the primitive alone and with other primitives. For example, the beam primitive is fully defined when the two nodes it is connected to are defined. The links between primitives and nodes are implemented through a connectivity matrix.

2.2 Generating Design Alternatives: The Modifier Module

For the purposes of automatically synthesising a wide range of connected-node systems representing feasible design alternatives, it is necessary to modify CNS representations in a robust, consistent and purposeful way. The Modifier Module encompasses a set of modification operators that are applicable to all connected-node systems, subject to user-defined constraints provided in the CNS definition. The modification operators reflect valid modifications of a spatial graph. Since the modification operators are domain independent, the effort of producing modification operators for specific design synthesis tasks is removed. Five general modification operators have been defined:

1. the node property modification operator (modifies the location of the node)
2. the primitive parameters modification operator
3. the primitive addition operator (adds a primitive to the design)
4. the connection swapping operator (changes the node to whom a primitive is connected)
5. the primitive removal operator (removes a primitive from the design).

While modifying a design, the Modifier Module also provides built-in ways to check for constraints, assuring that not only is the design a valid CNS representation, but also that it is feasible.

2.3 Evaluating Design alternatives: the Evaluation Module

The evaluation of connected-node system designs synthesised by the modification operators is carried out within the Evaluator Module. The Evaluator Module is in charge of passing the design to a simulation module, i.e. a commercial software package able to provide quantitative feedback on the design behaviour. The Evaluator will then use these feedbacks to calculate a set of values for design objectives and “soft” constraints. “Soft constraints” are used here to describe design constraints that are transformed into design objectives using penalty functions. By minimising the penalty function throughout the search process, the constraint violation is minimised to zero and thus the constraint is satisfied. Constraint violation is calculated as the error from the desired value of a specific design constraint. In this implementation all design objectives and “soft” constraints are assumed to be minimised by the search algorithm and must be formulated accordingly. The simulation is performed using either SUGAR, a software developed at Berkeley for MEMS analysis [14], or COMSOL [15], a more general Matlab-based multiphysics simulation tool. Either simulation modules can be used according to the requirements of the synthesis task and other simulation tools can be integrated as needed. The evaluation is, by nature, a domain specific task, depending on the type of device to evaluate and the design optimisation model. Nevertheless, the Evaluator Module is expected to adhere to a set of simple conventions. Its essential elements include:

- decoding of the connected-node system and encoding it into a simulation model, e.g. translation into SUGAR or COMSOL objects,
- calling simulation to perform analysis,
- reading results returned from simulation, and
- calculating all design performance metrics, both design objectives and “soft” constraints.

Again, the possibility to easily modify the design objectives according to the design task and the possibility to plug into the code different simulation software according to needs demonstrate the flexibility of the approach.

2.4 Search for New Designs: The Burst Method

The search is driven by an iterative loop that drives the search for designs according to the design objectives and constraints defined in the optimisation model. The implementation has been constructed so that a great variety of options can be integrated in terms of search heuristics, from implementing standard search methods to experimenting with new approaches. Any generate-and-test type algorithm would be suitable and easily ‘plugged’ into the loop like any other module. However, since the synthesis tasks under investigation have multiple objectives, a simple yet effective search, termed the ‘Burst Algorithm’ is sufficient to test the effectiveness of the CNS design representation and generation, and will later be shown to produce good quality results. A flowchart of the search loop is shown in Figure 4. The algorithm selects a random design from the archive and chooses the modification operators to apply to in short ‘bursts’, evaluating the result after each modification, but always accepting the valid modifications. The maximum length of each burst is a parameter set by the user, usually on the order of ten. Should any design that emerges be a new non-dominated solution to the problem, the design is archived. After each “burst”, a ‘return-to-base’ is carried out by selecting a new starting design from the existing archive of non-dominated solutions for the next “burst” of design modifications. The CNS representation and the modification operators are easily integrated in the search process.

2.5 Benchmark Applications

So far the method has been successfully applied to discrete structural topology optimisation, bitmap synthesis problems and microcompliant mechanisms [16, 12, 13]. The technique proved, in each of these examples, to be easily adaptable to different design tasks and produce optimised solutions comparable to other published methods. In the next section it will be shown how not only innovative, but also sufficiently detailed optimisation models can be incorporated for design synthesis tasks in complex design domains such as MEMS.

3 METHOD ADAPTABILITY AND REFINEMENT: EXAMPLES IN MEMS

MEMS are the design domain chosen for applying the CNS-Burst method. MEMS are a difficult engineering domain with a complex multidisciplinary nature. Due to their complexity, there has been significantly less simulation-driven computational research than in other fields, and MEMS design is still most often carried out by hand. To date a range of synthesis and optimisation methods have been developed as part of research efforts in order to provide the basis for MEMS design and synthesis tools, but research in this area is still in its early stages. The first contribution to the field is the research carried out by Fedder et al. [17]. This work is the first successful implementation of an

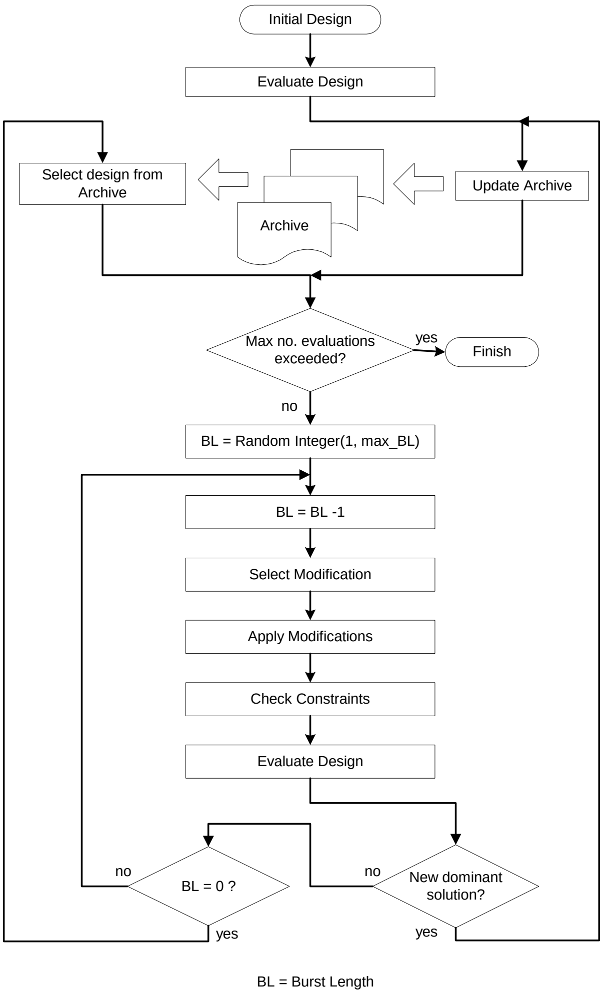


Figure 4. The Burst search.

optimisation tool that, starting from a parameterised layout, allows rapid exploration of MEMS devices through rapid modification of the design parameters. Argawal and Cagan's research [18], focusing on shape-based representation methods behind automated synthesis, offers a unique example of using a MEMS grammar for the generation of resonating structures. Antonsson, Li and Ma [19] orient their effort more towards synthesis of mask layouts, which are the patterns used to build and replicate the design on a silicon wafer. Another approach to computational synthesis of MEMS is that developed by Campbell [8]: an automation technique based on a shape grammar and the software SPECTRE. Other successful techniques, although applied exclusively to the synthesis of micro-compliant mechanisms, are homogenisation techniques [20]. While the most recent method in the area, developed at Berkeley by Zhou et al. [21], proposes the use of Multi-Objective Genetic Algorithms to automate synthesis of MEMS devices. A comprehensive review of synthesis methods for MEMS can be found in Ananthasuresh [22].

The specific case study examined here are micromechanical resonators. Silicon micromechanical resonators are a recent application that will enable the integration of resonators on silicon chips, leading to advances in new miniature-scale oscillators for wireless communication and mobile technologies. The fabrication process used for microresonators is SOI (silicon-on-insulator) MEMS process (Figure 14). Mechanical resonance is widely applied in high-precision oscillators: a typical example is a quartz-crystal resonator, used in a multitude of timing and frequency reference applications. However, a major drawback of quartz-crystals is their macroscopic size. The compact size and integrability of micromechanical resonators appear to open exceptional possibilities for creating miniature-scale wireless communication devices. The first case study examined is an example of how the method can be applied efficiently to solve topology optimisation tasks and help designers find innovative and optimised design alternatives in a complex design domain. This very simple application is introduced to illustrate the main aim of the method, to automatically generate spaces of alternative designs that tradeoff multiple design, and its potential. The solutions found, although topologically and spatially innovative, are highly conceptual, far from being ready to manufacture, and omit several important behavioural aspects of MEMS. The second example increases the complexity of the optimisation models to illustrate that computational synthesis can also support later design stages, where a greater level of detail and accuracy is required in order to transfer designs to the manufacturing phase. Further behaviours in MEMS design must be taken into account in order to consider the multiphysics aspects of design tasks. The second example is analysed in detail to show how it is possible to obtain solutions that are accurate enough to be brought into the final stages of refinement and manufacturing.

3.1 The Meandering resonator Case study

The first example examined is a meandering microresonator, a case study widely examined in MEMS synthesis literature [21]. The design of the meandering microresonator consists of a centre mass supported by four springs, which are made of sequentially connected beams (beam primitives) extending from an anchor point to one of the four port-nodes of the mass primitive (Figure 5).

The design objective is to minimize the device area, defined as the bounding box around the microresonator, subject to a target natural resonant frequency constraint of 10000 rad/s and a minimum width of beam elements of 1µm. The initial design (Figure 5) used to start the search has a resonant frequency far from the target (f_0 = 93723 rad/s) and a design area of 30 µm^2.

Design variables considered are the number of beams, the connectivity of the beams (device topology), length and width of each beam (*w, l*), the number of nodes and the geometric position (x, y) of all the nodes. The optimisation model for this design task is:

$$\min\{|\Delta f|, A\}, \quad S.t. \quad w \le 1\mu m \tag{1}$$

where $\Delta f = (f_0 - f)$ is a “soft constraint” representing the error in natural resonant frequency and A is the area of the box surrounding the device. The only hard constraint considered, which must be met throughout the search, is the minimum width of the beam elements (w), due to fabrication limitations (1µm). Figure 6 shows a Pareto-front obtained from one run of the method using 10000 iterations. The coordinates of the plot represent the two objectives of the search to be minimised, the error in natural resonant frequency and the area of the device in the x- and y- direction respectively. Two of the designs obtained in this archive are also shown in Figure 7 and 8. Solution 1 presents a reduction of the area of the device of almost one-third compared to the initial design. Solution 2 presents a $\Delta f < 1\%$.

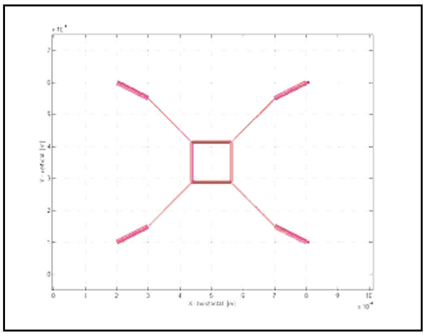


Figure 5: Initial Design, Δf = 83723, A = 30 µm^2.

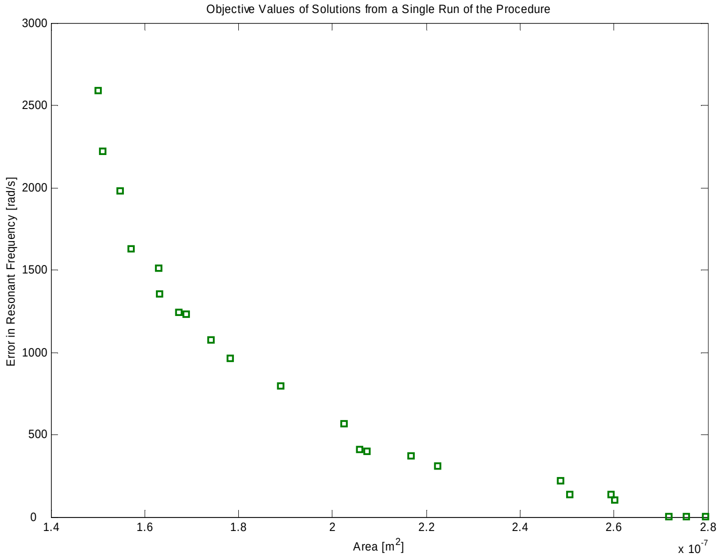


Figure 6: Archive of solutions obtained from a single run of CNS-Burst using 10000 iterations.

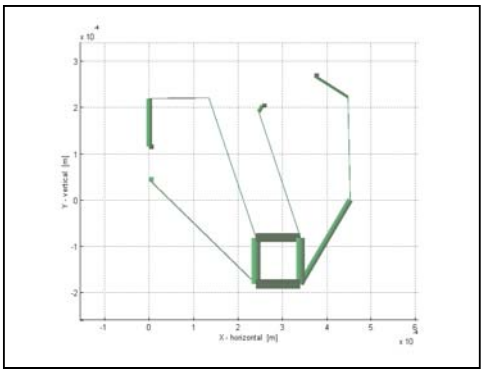


Figure 7: Solution 1, $\Delta f = 410$, $A = 20.02\ \mu m^2$

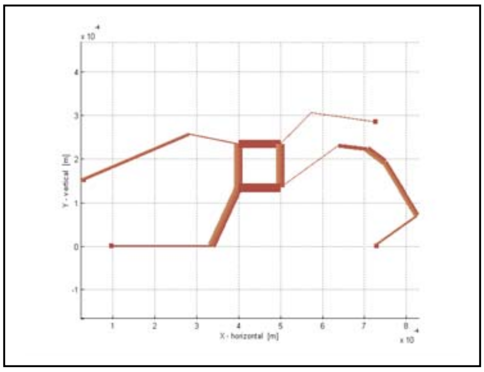


Figure 8: Solution 1, $\Delta f = 36$, $A = 25\ \mu m^2$

3.2 The FF-Beam Microresonator Case study

The aim of this case study is to design free-free (FF) beam microresonators with primary resonance in the 10MHz-3GHz range, to be integrated in low noise microelectromechanical-based reference oscillators for wireless communication and GSM/DCS mobile technologies. A typical FF-beam resonator includes a resonant structure part and an electrical drive part (Figure 9). The resonant structure is anchored in its centre, while its ends are free to resonate. At operational frequency, the resonant structure of the FF-beam resonator vibrates longitudinally, changing the dimensions of the gaps between the resonant beam and electrodes and, as a consequence, altering the capacitance sensed. Such vibration mode is also known as 'bulk' mode. An analysis of the parameters that determine the behaviour of FF-beam resonators has been widely described in literature [23] and will not be repeated here. In the next sub-sections a brief description of the parameters that are of interest to this case study will be given.

3.2.1 Key Design Parameters: Some Considerations

Microresonators design is not an easy task, due to the many behavioural considerations. The interdependency of the design parameters creates many trade-offs to be taken into account. Three design parameters have been here considered for the design of the FF-beam resonator: motional resistance R, longitudinal resonant frequency f and quality factor Q. Motional resistance is the parameter representing electrical loss and depends on many factors, including the fabrication process and the material parameters. The quality factor is the ratio between the maximum vibration energy stored by the system per cycle and the energy dissipated per cycle of vibration [24]. The quality factor will not be considered as a search objective in the following example. It is indeed difficult to take into account all the relations between design parameters, especially if the design is carried out by hand. Automated synthesis techniques incorporating multiobjective optimisation can help designers to compute and analyse trade-offs as well as providing innovative solutions to meet their preferences.

3.2.2 Results

In this section results obtained using CNS-Burst to synthesise and optimise designs for the FF-beam microresonators are reported. The design objectives of the search are:

- a target operational frequency of 20 MHz ("soft" constraint)
- a minimal motional resistance (design objective).

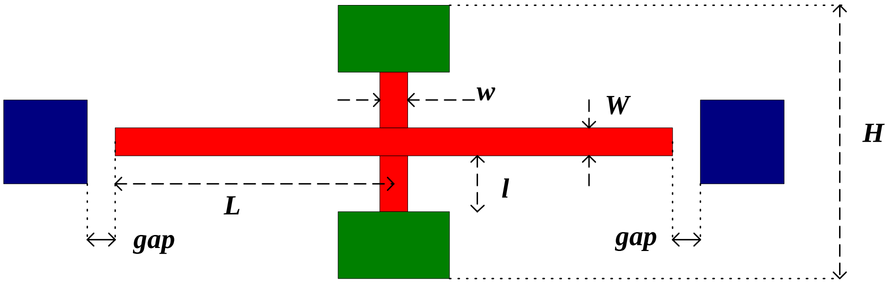


Figure 9: FF-beam resonator model (red: resonant structure; green: anchors; blue: electrodes).

The design variables are the length and width of the anchors (w, l) and the length and width of the resonant beam: W, L. The “hard” design constraints are:

- the minimum width of the beam elements (w, W), due to fabrication limitations (1μm)
- the height of the resonator (H), not to exceed 30.5μm.

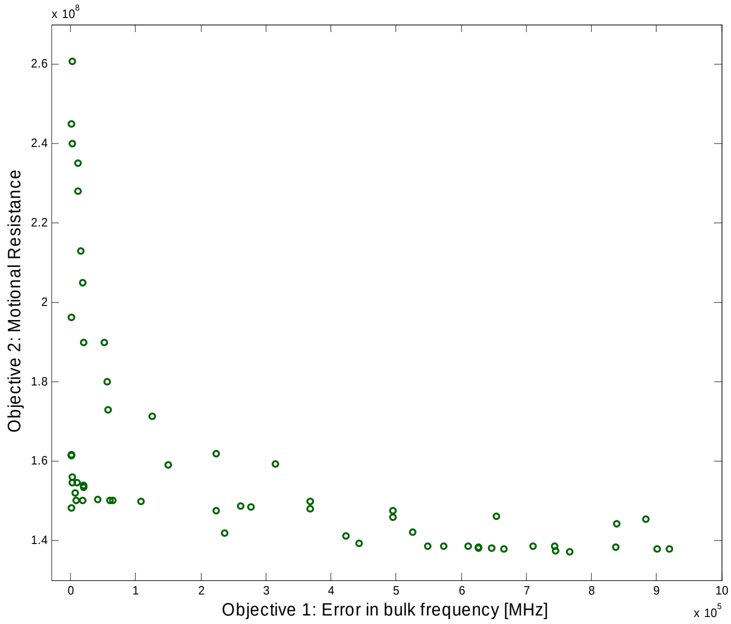
The initial design (Figure 9) used to start the search has a resonant frequency, f_0 = 20.396 MHz, and a motional resistance, R_m= 229 MΩ, and was obtained using the traditional hand calculation methods that are common practice among MEMS designers [23].

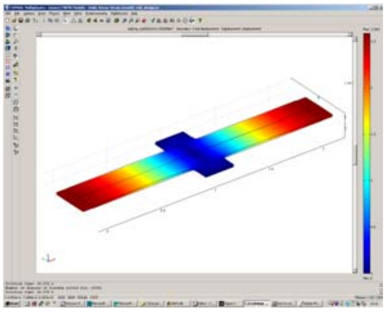
The optimisation model for this design task is:

$$\min\{|\Delta f|, R\}, \quad S.t. \quad W \leq 1\mu m, w \leq 1\mu m, H \leq 61\mu m \tag{4}$$

where $\Delta f = (f_0 - f)$.

Figure 10 shows an archive of solutions obtained from five optimisation runs at 5000 iterations each. Each run, resulting in an archive of non-dominated solutions distributed on a concave Pareto front, is overlaid on the same axes. The two coordinates of the plot represent the objectives of the search (the error in resonant frequency and the motional resistance in the x- and y- direction respectively). The results show that 52% of the solutions kept in the design archive have an error in resonant frequency, Δf < 1%, and that 93% of the solutions have a motional resistance, R, less than the initial value, R_m. The design archive presents a variety of solutions to the task, also illustrating performance trade-offs. Figure 11 and 12 show two interesting design solutions obtained. Solution 1 shows a Δf of 0.0013% from the target. Solution 2 has R = 137 MΩ (decrement of 40% from the initial value R_m). Solution 1 and 2 are compared with the initial design in Table 1.





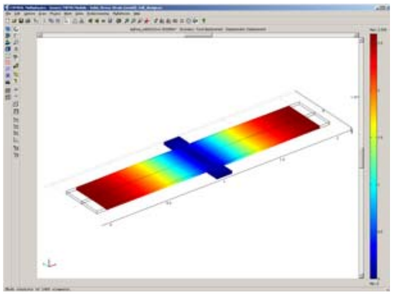


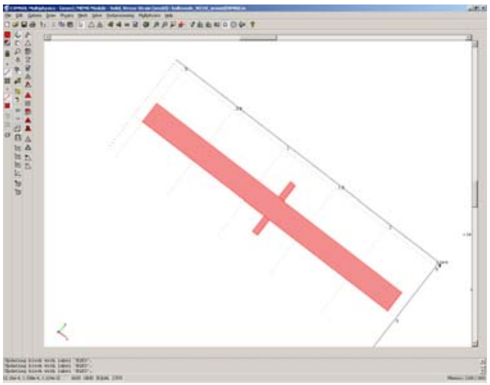
Figure 10. Five Archives of solutions resulted from 5 runs at 5000 iterations each and overlaid on the same axes. Figure 11 (top right): Solution 1, Δf = 269, R = 150. Figure 12: Solution 2, Δf = 665.859, R = 137.

4 CONCLUSIONS AND FUTURE WORK

This paper presented an investigation into the use and extension of multi-domain computational synthesis methods in design. The aim of the research was to demonstrate that a general computational-synthesis method not targeted at a specific design domain can nevertheless produce accurate and reliable solutions for a complex design domain. The CNS-Burst method combines a multicriteria generate-and-test algorithm (Burst) in conjunction with a general Connected Node System (CNS) design representation and provides automatic links to multiphysics simulation for quantitative evaluation of design performance throughout the synthesis process. The CNS-Burst method was shown here to be successfully extended to the design domain of MEMS, including both size and topology, multicritiera optimisation problems.

Two MEMS design tasks were presented to illustrate application of the method to multicriteria synthesis and extending the optimisation model to incorporate models representing more complex MEMS behaviour. The solutions obtained illustrate the success of the method in producing Pareto archives that meet the design requirements and trade-off the design objectives. Many solutions obtained show considerable improvement to solutions obtained by hand, which were used as initial solutions. The solutions were obtained in a short amount of time compared to the lengthy manual procedure where, after solutions are designed by hand, they are transferred manually into MEMS software for analysis. For comparison, using the computational method, about every minute a new solution is created and automatically simulated and evaluated. As a high number of solutions for MEMS design tasks are desirable, considering the rate of fabrication failures due to unpredictable stress in the material during the manufacturing phase of layer deposition, the method is able to provide many different design solutions with similar performance to enable selection based on criteria that can not be modelled directly. The high level of accuracy of the solutions obtained, compared to the important behaviours required, makes them ready to be analysed for fabrication tolerances and finally manufactured. The designs obtained are in fact improvable only with calculations of second order effects, which is not necessary in early design stages.

Research is currently underway to apply the method to more complex problems of topology and shape optimisation in MEMS, including further extensions to the behaviour models. A significant area of future investigation rests with the potential of the technique's multicriteria search capabilities and how the method can be refined to guide the search more effectively and efficiently, due to high computation cost of extended simulations. The search could be coupled with machine learning techniques, as previously done [10, 17]. Through the successful application of the method to the MEMS examples presented, the versatility of the approach was shown and capability to incorporate sufficiently accurate models and simulation software. The results obtained confirmed that the straightforward adaptability of multi-domain synthesis methods creates the potential to provide customisable tools for automated design synthesis. This novel use of the computer in design practice will not only help designers to achieve the most beneficial design alternatives, but also has the potential to boost innovation in many fields of design.



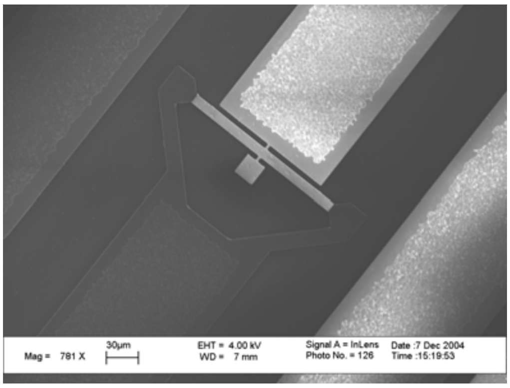


Figure 13: COMSOL model of solution; Figure 14: Manufactured solution.

REFERENCES

[1] Shea, K. 'Design Automation Tools for Early Phase Engineering'. To appear in the proceedings of *ProSTEP iViP Science Days 2007*, 25-26 September, Bremen, Germany, 2007.

[2] Antonsson E.K. and Cagan J. 'Formal Engineering Design Synthesis' (Cambridge University Press, 2001, Cambridge).

[3] Shea K. and X. Zhao. ' A Novel Noon Mark Cantilever Support: From Design Generation to Realization'. In *IASS 2004: Shell and Spatial Structures from Models to Realization*, 20-24

September, Montpellier, France, 2004.

[4] Starling A.C. and Shea, K. 'A Parallel Grammar for Simulation-Driven Mechanical Computational Design'. In *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, USA

[5] Lipson H. and Pollack J.B. 'Evolution of Physical Machines: Towards Escape Velocity'. In *6th International Conference on Artificial Intelligence in Design, AID'00*, Worchester Polytechnic Institute, Worchester, Massachusetts, USA, 2000.

[6] Cagan, J., Campbell, M.I., Finger, S. and Tomyama, T. 'A Framework for Computational Design Synthesis: Model and Applications'. In *Journal of Computing and Information Science in Engineering*, Vol.5, 2005, pp.171-181.

[7] Koza J.R., Keane M.A. and Streeter M.J. 'What's AI Done for Me Lately? Genetic Programming's Human Competitive Results'. In *IEEE Intelligent Systems*, May/June 2003, Vol.18, n.3, 25-31.

[8] Campbell, M. I., Nair, S. and Patel, J. 'A Unified Approach to Solving Graph Based Design Problems'. To appear in the *Proceedings of the ASME 2007 IDETC/CIE Conferences*, Las Vegas, NV, USA, September 2007.

[9] Rudolph S. and Kormeier T. 'On Self –Similarity as a Design Paradigm'. In *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, USA, October 2005.

[10] Vale C.A.W. 'Multicriteria Dynamic Synthesis via Machine Learning'. *MPhil Thesis*, 2002, University of Cambridge, UK.

[11] Pareto V. 'Manual of Political Economy'. Translation of the 1927 edition (A.M.Kelly, 1971, New York).

[12] Bolognini F. 'Computational Synthesis of Microsystems'. *MPhil Thesis*, 2004, University of Cambridge, UK.

[13] Bolognini F., Shea K., Seshia A.A. and Vale C.W. 'A Multicriteria Systm-based Method for Simulation-driven Design Synthesis'. In *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Philadelphia, Pennsylvania, USA, September 2006.

[14] http://www-bsac.eecs.berkeley.edu/cadtools/sugar/sugar/

[15] http://www.comsol.com

[16] Vale C.A.W. and Shea K. 'A Machine Learning-Based Approach to Accelerating Computational Design Synthesis'. In *ICED 2003 14th International Conference on Engineering Design*, Stockholm, Sweden, August 2003, 183-184.

[17] Fedder G.K., Iyer S. and Mukherjee T. 'Automated Synthesis of Microresonators'. In *9th Int'l. Conf.on Solid-State Sensors and Actuators (Transducers '97)*, Chicago, USA , June 16-19, 1997

[18] Argawal M., Cagan J. and Stiny G. 'A Micro Language: Generating MEMS Resonators using a Coupled Form-Function Shape Grammar'. In *Environment and Planning B: Planning and Design*, 2000, 27, pp.615-626.

[19] Ma L. and Antonsson E.K. 'Robust Mask-Layout Synthesis for MEMS'. In *2001 International Conference on Modeling and Simulation of Microsystems*, Chapter 5: Optimisation, Nanothech 2001, 1, pp.128-131.

[20] Bensdøe M.P. and Sigmund O. 'Topology Optimisation: Theory, Methods and Applications', 2003 (Springer, NY).

[21] Zhou N., Agogino, A.M. and Pister, K.S.J. 'Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS)'. In *ASME 2002 Design Engineering Technical Conference*, Montreal, Canada, Sept.29-Oct.2 2002.

[22] Ananthasuresh G.K. 'Optimal Design Synthesis for MEMS', 2003 (Kluwer Academic Publishers)

[23] Mattila T., Kiihamäki J., Lamminmäki T., Jaakkola O., Rantakari P., Oja A., Seppä H., Kattelus H. and Tittonen I. 'A 12 MHz Micromechanical Bulk Acoustic Mode Oscillator'. In *Sensors and ctuators A: Physical, 2002, 101(1), pp.1-9.*

[24] Hao Z. and Ayazi F. 'Support Loss in Micromechanical Disk Resonators'. In *18th IEEE International Conference on Micro Electro Mechanical Systems*, 30 Jan.-3 Feb. 2005, pp.137-141.

Contact: F. Bolognini
University of Cambridge
Engineering Department
Trumpington Street
Cambridge CB2 1PZ
UK
Phone: +44 (0) 1223-760562
e-mail: fb252am.ac.uk
URL: https://www-edc.eng.cam.ac.uk/