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

TOWARDS AN EVOLVABLE CHROMOSOME MODEL FOR INTERACTIVE COMPUTER DESIGN SUPPORT

Zhun Fan, Mogens M. Andreasen, Jiachuan Wang, Erik D. Goodman, Lars Hein

Keywords: chromosome model, evolutionary synthesis, interactive evolutionary computation, domain theory, morphology

1 Introduction

In traditional function based evolutionary design synthesis, the encoding of the design is decided by the designer, and no formal process of encoding is provided in most research in this line. As a result of limitations in the encodings typically used, evolutionary design synthesis often produces only small-scale designs – that is, relatively few components are used. And because they fail to cover the full spectrum of user needs, and frequently do not consider manufacturing processes, costs and constraints, the design results are often not very useful to industry and only end up as 'new ideas' generated to satisfy academic curiosity. This paper tries to extend the algorithmic design process of traditional evolutionary synthesis approaches, integrating the product development process and broader customer needs. The method is to adopt a chromosome model based on the domain theory described in [1] to facilitate defining of encodings, e.g. to decide the most important parts and characteristics of the designs to be evolved. From the viewpoint of the chromosome model, this approach is an extension of the model towards an evolvable chromosome model within the domain theory.

An evolvable chromosome model is reasonable because designs are intrinsically evolutionary and no design should be static. The evolvable chromosome model can also facilitate computer-aided conceptual design in an interactive evolutionary design system, thus pushing the research in functional-based evolutionary design synthesis a further step towards industrial-oriented applications.

2 Evolutionary engineering design

Highly automated function-based synthesis methods have emerged in recent years [2]. Among them approaches based on evolutionary computation (EC) appear to be one of the most promising groups. During the past ten years, the effort of integrating evolutionary computation with engineering design has rapidly increased, taking advantage of EC’s search ability to explore the design space. Many important research advances and results of evolutionary engineering design have been reported, including [2]-[8], [11].

2.1 Generation of morphology using evolutionary approaches

In the design community, one of the most powerful systematic methods for creating conceptual designs is Morphology. The core idea of the Morphology Method is that there exist sets of important characteristics that are believed to be common in all desired solutions. Each characteristic can be varied and a certain number of alternative solutions for satisfying the characteristic can be established. Then if we can identify that set of characteristics, any combination of each sub-solution will be a potential solution or design candidate.

EC refers to a class of general-purpose search algorithms based on (admittedly very incomplete) abstraction of principles of biological evolution and natural selection. These algorithms implement biologically inspired computations that manipulate a population of candidate solutions (the "parents") to generate new variations (the "offspring"). At each step (or "generation") in the computation, some of the less promising candidates in the population are discarded and replaced by new candidates ("survival of the fittest").

In summary, EC is very relevant to the core principles of design methodology, namely, to create several concepts, and to select the best one based upon criteria that mirror what is believed to represent high quality in a solution. Design models based on evolutionary computation start, like the Morphology Method, with generating a population of design concepts candidates, and then according to evaluation criteria set forth by the designer, each design candidate in the population will be evaluated and assigned a value that represents its 'goodness' / fitness for the design. With this, EC uses certain mechanisms, such as crossover and mutation that are analogous to mechanisms Nature uses to evolve its creatures, to gradually evolve / reconfigure the population of design candidates so that in each offspring generation, the population of design concepts as a whole is superior to its parental generation, according to evaluation criteria set forth by the designer. In this way, the population of design concepts is guided towards better designs in each generation, and after a number of generations of improvement / evolution it will converge to one or a set of 'good' candidates that the designer can select from or use to make further trade offs. Figure 1 shows an overflow of the design process based on evolutionary computation.

2.2 Topology exploring and parameter optimization in engineering design

Two most widely used types of evolutionary computation techniques include genetic algorithm (GA) and genetic programming (GP).

GA is a very simple, straightforward, yet a powerful approach for global search of the parameter design space[6]. GA usually represents/encodes an individual design candidate with a string that concatenates parameters (both real and binary) considered important in the design. The design topology, in most cases, is fixed, so that the length of the string for each individual is the same, facilitating the crossover and mutation operations towards strings.

Genetic programming is an extension of the genetic algorithm, and it uses evolution to optimize actual computer programs or algorithms to solve some task, typically involving a tree-type (or other variable-length) representation, thus lending itself very well to explore topology design space [7]. Because GP (genetic programming) can manipulate variable-sized strings, it is especially useful for representing developmental processes and processing topological information. Most design methods based on GP require a preliminary design, or a design embryo, which need not contain all of the necessary components, or the necessary number of components, but only enough information to allow specifying the behaviors desired of the system (defining objectives and variables constrained, for example).

Genetic algorithm is widely used to optimize parameters, but lacks the ability in exploring topology search space. Compared with genetic algorithm, genetic programming is an intrinsically strong tool in open-ended topological exploration, because the tree structure of the GP chromosome is flexible in generation and reconfiguration, with constraints of maximum depth and maximum nodes only imposed by practical implementation considerations. In addition, functions used in GP, rather than rules used in specially designed GA, allow the designer to explore design regions (in the whole design space) with which he or she is not familiar. In practice, GP and GA may be used together in an evolutionary design system to explore both the topology and paramter design space.

It is important to point out that in the conventional design environment, designers' decision-making is biased by both the capabilities of simulation tools and the designer's experience and intuition [4]. It is hard for the designer to make an "imaginative jump or creative leap" from one design candidate to another. But design tools based on evolutionary approaches can free designers from this kind of "design fixation" and the limitations of conventional wisdom, allowing them to explore a huge number of possible candidates for a design problem, and increasingly, the probability to discover novel designs uncharted before by human exploration.

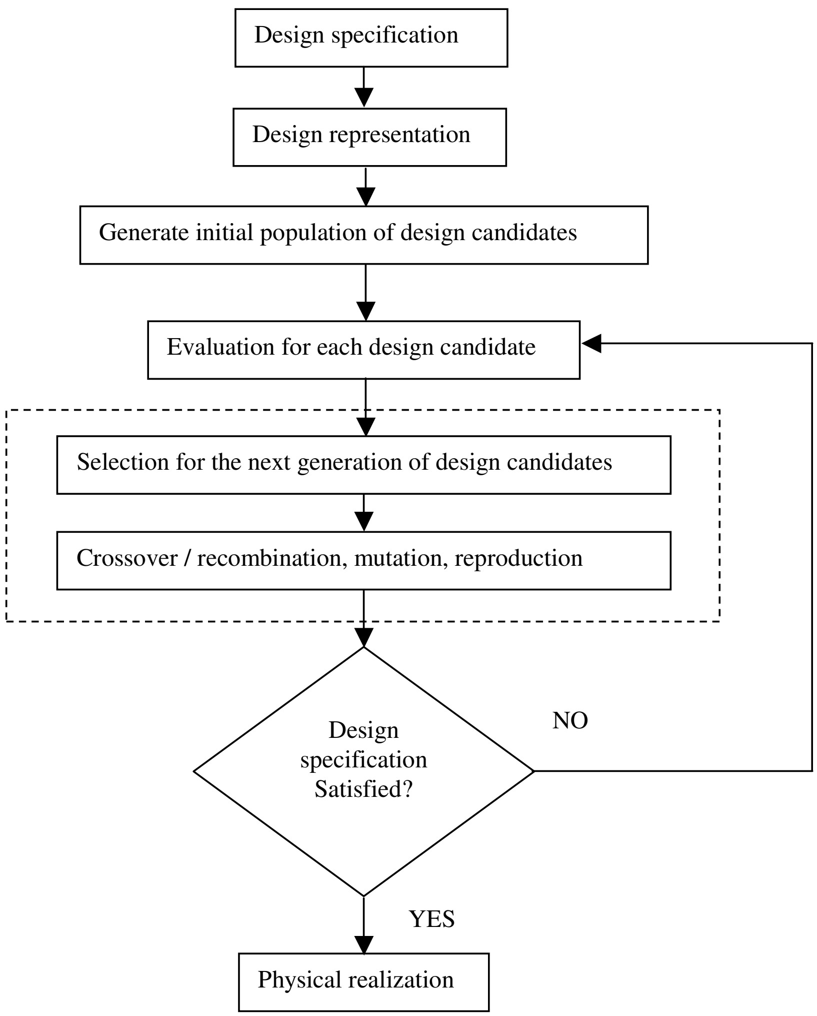


Figure 1. An overflow of the evolutionary engineering design process

3 Chromosome model

Unfortunately, despite the significance of research break-throughs in academia, the reported results of evolutionary engineering design are still not ready to be used widely in industry because they cannot link user aspects and design intent to a structural product model, so they only cover part of our current function vocabulary and/or design process, and generally lack the ability to generate realizable geometries and structural topologies. As a consequence, the identification of important characteristics is also an ad-hoc process. It is highly recommended that a richer representation language adding process, function, organ and geometric issues of product should be used to 'spell' the product so that storage and reuse of design knowledge becomes possible. A complete electronic product definition is necessary in supporting engineering designers in their design activities. According to [9], four attributes are relevant in defining a product: characteristics, inherent properties, relational properties, and qualities.

Characteristics are a class of design attributes that the designers can determine directly during design. They may include structure (both behavioral and physical), form, dimension, surface quality, material and so on.

Inherent properties describe the behaviors of a design and can be determined by the design characteristics and the environment. They can also be determined at high-level behavorial models by the designers in a top down design process.

Relational properties are design attributes that describe the behavior of the so-called meetings between the design and the life phase system. Relational properties are causal determined by the characteristics of the design, the life phase system characteristics, and the meeting characteristics. Examples on relational properties are costs, throughput time, flexibility, etc.

Quality, meaning pride of ownership, can also be considered as the stakeholder's reactions on inherent and relational properties. Determining quality requires a person observing and reacting, and there is no causality between properties and quality.

But how can these attributes be identified? A very promising approach is the so-called chromosome model, which has a structure in accordance with the domain theory [1]. The basic idea is to model the product from four hierarchical viewpoints, based on strict identification of structural and behavioral aspects of a product:

- A process view, with a structure of activities related to the product, for example the use process, the product life cycle etc. In this viewpoint, to understand how the transformation of materials, energy, and information of the product are related to their use or functions is central.

- A functional view, with a structure of the desired functions or effects. These functions must be able to facilitate the necessary transformations.

- An organ view, with a structure of functional carriers or solutions which create the desired functions or effects of the product. The result of design considerations is an organ structure.

- A part view, with a structure of parts and their assembly relations. By determine materials, form, tolerances and surface quality of each part and relations between the parts, the necessary conditions for the organs and their functionality are created.

The graphical representation of the chromosome model is show in Figure 2.

According to the theory of technical systems the design can be modelled from two constitutive viewpoints: organ and part viewpoints. The two constitutive viewpoints are necessary for explaining the behavior of a design and the physical realization.

The other two viewpoints, process view and functional view, provide a systematic way of analysing the design requirements and intentions and relate them to the organ and part viewpoints. An approach to carry out this analysis is called Function-Means tree.

In summary, the chromosome model provides an extended and hierarchical view of product configuration that relates a substantial part of all the data, information and knowledge about the product to the product model. This view can be used to extend previous research in evolutionary engineering design that only encoded the part domain or perhaps organ domain of the chromosome model.

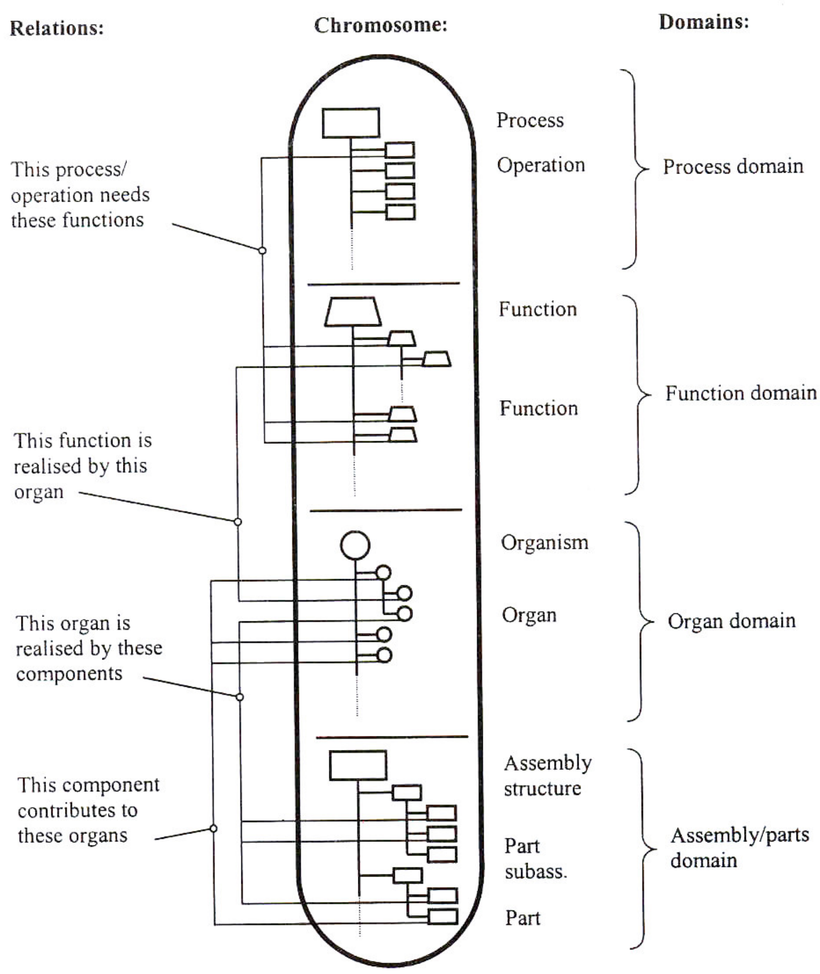
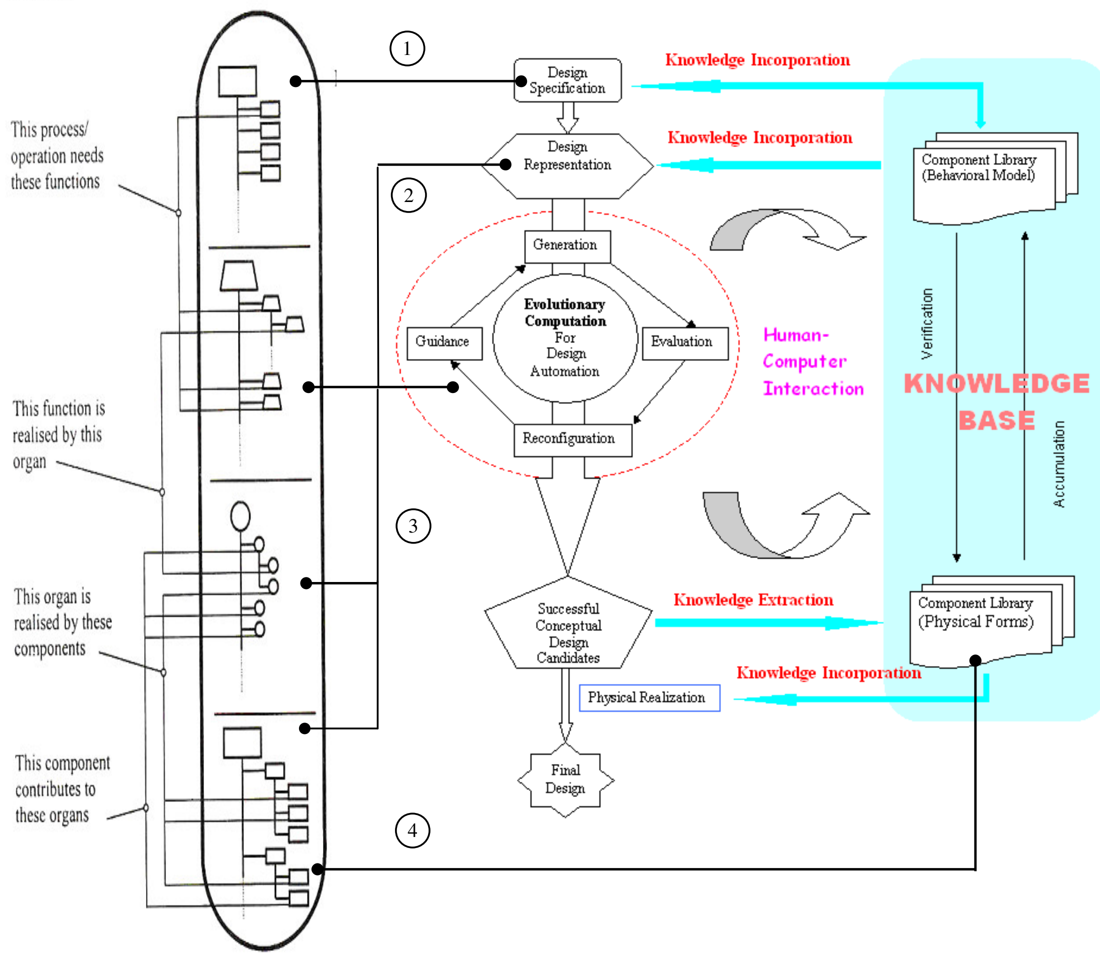


Figure 2. The chromosome product model, adapted from [1]

4 Integrated evolutionary engineering design

4.1 Integrating chromosome model with evolutionary design



(1) The design specification motivates analysis in the process domain of the chromosome model. This will then in turn decide the functions needed to realize the process in the function domain, and the organs needed to realize the functions in the organ domain, and components that contribute to the organs in the part domain.
(2) The design representation is collectively determined by the decisions made in the function domain, the organ domain, and the part domain.
(3) The function domain, the organ domain, and the part domain of the chromosome model can all influence how evolutionary design may be carried out, e.g. in the stage of encoding design characteristics, and defining proper EC operators like GP functions.
(4) Parts domain relates to the physical forms of the component library used in the design.

Figure 3. The integrated evolutionary engineering design framework with an evolvable chromosome model

The framework of Integrated Evolutionary Engineering Design (IEED) is shown in Figure 3. The overall procedure of IEED starts with the design specification, including design objectives, design constraints, and design preferences, etc. The design specification motivates analysis in the process domain of the chromosome model. This will then decide the functions needed to realize the process in the function domain, and the organs needed to realize the functions in the organ domain, and components that contribute to the organs in the part domain. With a complete chromosome model, we can make a decision on the how to represent design candidates. It is important to point out that in many engineering design cases, a design need to be represented in multiple levels of abstractions. The different levels of abstractions may correspond to the part domain, or the organ domain, or even the function domain in the chromosome model.

After we have design representations, we can move on to the next step of the Integrated Evolutionary Design framework – to run evolutionary computation for design automation. A preparatory step for this includes several more issues to be determined, e.g. the encoding of design characteristics, and defining of proper EC operators like GP functions if needed. Again, they will be collectively determined by previous decisions we made on part domain, organ domain, and even function domain.

The automated design loop of evolutionary computation includes four steps of generation, evaluation, reconfiguration, and guidance. Design candidates of engineering systems are often represented in several levels of abstractions [10] in the conceptual design level. The automated design loop of evolutionary computation may also take place in different cycles, leading to design results in corresponding representations. After the step of conceptual design, we move to the following step of detailed design. In this step, physical realization transforms the conceptual design to its final physical structure according to the physical forms of the component library used in the design, i.e. decided in the part domain of the chromosome model. It is noted that physical realization may be a comprehensive procedure itself.

4.2 Integrating human interaction with evolutionary design

Due to the characteristics of uncertainty, multi-objectives, severe constraints and high-modality related to real world designs, it is almost impossible for EC to evolve strictly realizable designs in an efficient manner if we use EC merely as a set-and-run tool. EC can perform much better to play a supporting role to enhance design insight and assist decision-making, rather than to act merely as a terminal optimizer that gives customers a final result [5] [8].

It is hoped that a design knowledge base can be created in this interactive evolutionary design process. For example, the multiple diversity solutions of engineering designs obtained through evolutionary computation can provide valuable information to the user to foster a better insight of the problem domain and help to identify best direction for future investigation. In addition, the knowledge acquired in the process may assist the designer to refine design objectives and modify design representations. In the process, Knowledge incorporation and knowledge extraction are two major forms of knowledge interaction.

In summary, human-computer interactions may happen in various forms that include but are not limited to the following aspects.

Specification of design objective and constraints

Specifications of design objectives and constraints are the input to the interactive evolutionary design framework. They are provided by the user at the beginning stage of the design, and specified at the process domain of the chromosome model. It is obvious that specifying design objectives is a process that incorporates domain knowledge and human preferences. What is more, because many assumptions about objectives and constraints may not be correct at the beginning, and are subject to changes such as the market condition, human's interactions with the computer are desired to reflect the corrections and modifications.

Design of EC operators

Take the definition of GP functions for example. Executing the GP tree can accomplish the collective tasks that the user embedded in the functions. All functions in GP tree belong to a function set. Designing the function set is therefore one of the most significant steps in setting up GP run. Because most functions in the function set deal with the configuration of building blocks/organs of the design, in practice, it is important to first decide the selection of building blocks, or the component library, of the proposed design. Although the component library should be decided in the chromosome model, at organ domain and part domain respectively, it happens often that the designer finds it necessary to change the contents of the organ and component library in the design cycle. In this case, the designer's expertise knowledge should be incorporated into the chromosome model and accordingly modify the evolutionary design process in an interactive manner.

Definition and modification of design evaluation

Design evaluation involves defining an objective or fitness function against which each design candidate is tested for suitability for matching the design specifications under various design constraints. Because the purpose of engineering design is to make products for a changing world, engineering design is an interactive process of integrating new information, new technologies and new biases from the marketplace. As a result, the fitness function should be able to take feedback from designers and customers constantly, enabling it to reflect changing market environments or user preferences.

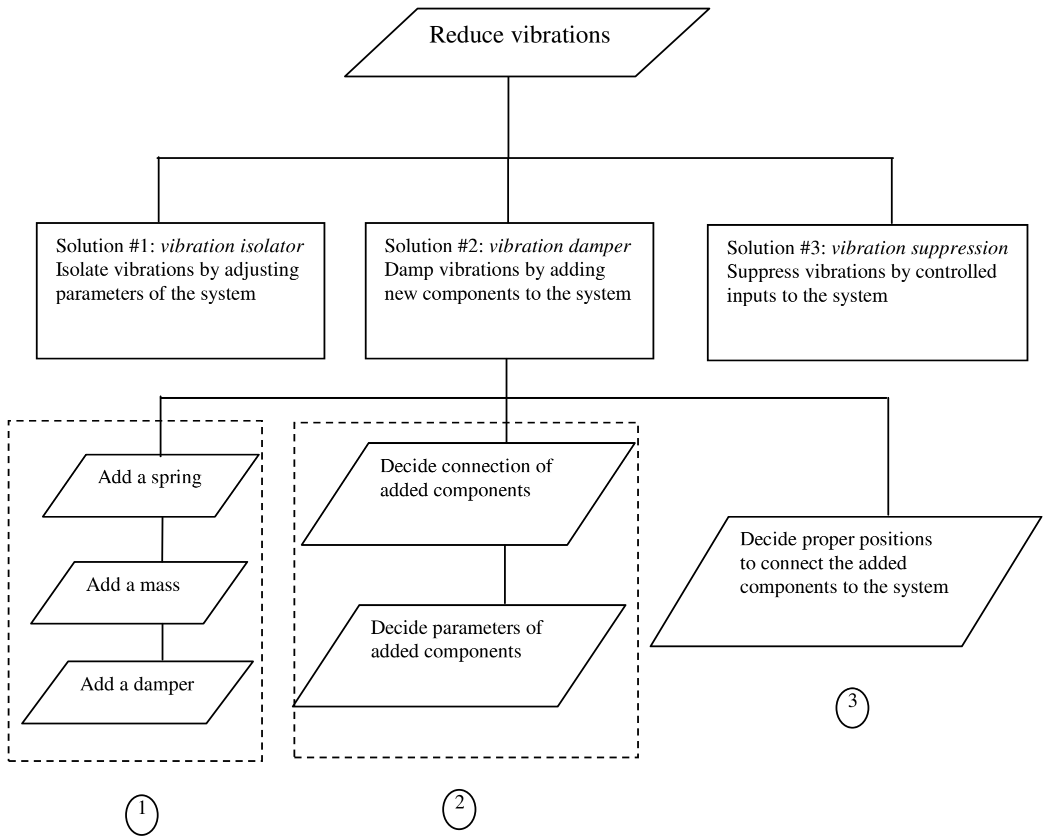
It is speculated that with the above procedures to implement an interactive evolutionary design system, a unique, rich knowledge system that can not only gather and relate data, but also process and evolve data will be created. In addition, the chromosome model in this knowledge system is now not a static one, but more a dynamic one that can evolve during the design process, continuously adapting to changing environments including market demands, customer preferences, or technology advances, etc. In this way the designers and the computer can work as a symbiotic, interacting team to tackle design problems.

5 An example

We are going to use an example of typewriter redesign to illustrate the Integrated Evolutionary Design method. Because the length of a conference paper is limited, its detailed design process will be introduced in another paper.

The problem was presented by C. Denny and W. Oates of IBM, Lexington, KY, in 1972. The original typewrite system includes electric voltage source, motor and mechanical parts. The problem with the design is the position output of the load has intense vibrations. The design specification is to reduce the vibration of the load to an acceptable level, given certain command conditions for rotational position. In particular, we want the settling time to be less than 70ms when the input voltage is stepped from zero to one.

Given the design specification, we need to create a chromosome model based on our analysis of the design system. The Function-Means-tree used to carry out this analysis is shown in Figure 4.



Function 1,2, and 3 can all be accomplished by an evolutionary engineering design approach based on genetic programming and bond graph in the conceputal level

Figure 4. A Function-Means tree for vibration damper of the typewriter system

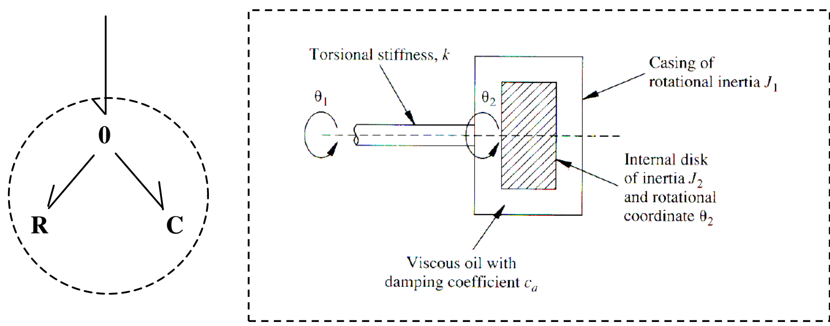
To reduce vibrations, there are basically three ways: vibration isolation, vibration absorber / damper, or active vibration suppression.

Vibration isolation reduces vibrations through adjusting parameters (e.g. the stiffness and damping) of the existing system to cause its vibration response to behave in a desired fashion. After analysis, we decided not to take this method because the original design has fixed the materials so that it is difficult to change the mass and stiffness of the system more than a few percent.

Active vibration suppression uses an external adjustable (or active) device, called an actuator (e.g. a hydraulic piston, a piezoelectric device, an electric motor) to provide a force to the device, structure, or machine whose vibration properties are to be changed. Again, it is not attractive to us because the added components of actuators and sensors may be too expensive for a typewrite design.

Therefore, we are left with the choice of vibration absorber / damper, which basically inserts new components to take vibrations from the primary system that is to be protected from the vibrations. Then several questions need to be answered, e.g. what types of components need to be inserted, how should we connect the components before inserting them to the primary system, at which positions should the components be inserted, and how to decide parameters for the components, etc.

Given several modifiable sites that indicate potential locations that the added components may be inserted, the evolutionary design approach described in this research can answer the above questions in a highly automated manner and suggest several designs (with different topologies) that damp the vibrations successfully. The designs have settling time to be about 50ms, thus fulfilling the design specification, i.e. to make the settling time to be less than 70ms. The method utilizes genetic programming as an automated search mechanism and bond graph as a behavioral modeling tool for the typewrite system. Figure 5 shows the inserted components and their physical realization in one of the design candidates. Figure 6 shows the simulation result of the evolved design. More details can be found in [12].



Organ view

Parts view

Figure 5. The inserted components and their physical realization

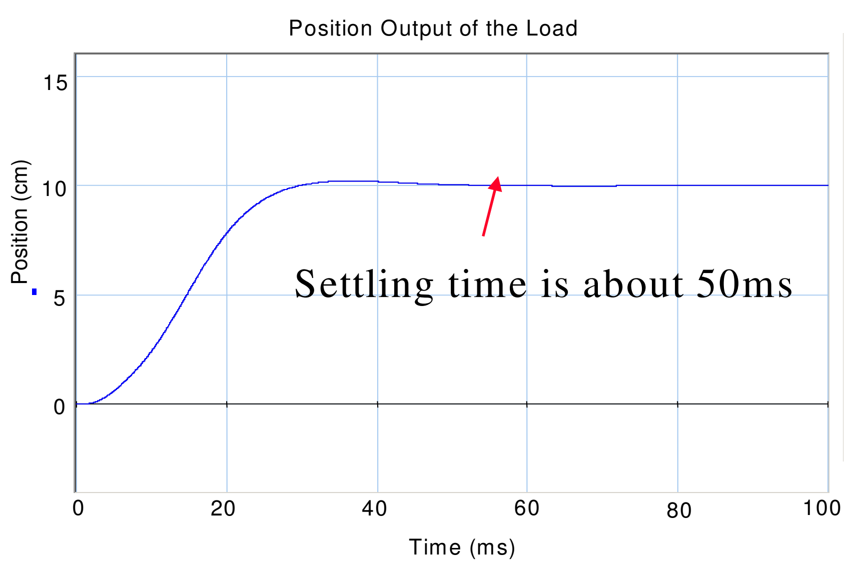


Figure 6. Simulation result of the evolved design

6 Conclusions

The paper proposes an integrated evolutionary engineering design framework that integrates the chromosome model in the domain theory, the evolutionary design, and human interaction. In this framework, the chromosome model is not a static one, but dynamic and evolvable with the help of evolutionary design process. This evolvable chromosome model helps the designer to improve creativity in the design process, suggesting them with unconventional design concepts, and preventing them from looking for solutions only in a reduced solution space. The systematic analytical process to obtain a chromosome model before running evolutionary design algorithms also helps the designer to have a complete view of design requirements and intentions. Human interaction is integrated to the framework due to the complex and dynamic nature of engineering design. It also helps the designer to accumulate design knowledge and form a design knowledge base. Although a simple example demonstrates its feasibility, many issues still need to be addressed before the framework can be utilized widely in industry.

References

[1] Andreasen M. M., “The Role of Artefact Theories in Design”, in Universal Design Theory, H. Grabowski, S. Rude, G. Grein (ed.), 1998, pp. 47 – 56

[2] Antosson E. K. and Cagan J. (ed.), “Formal Engineering Design Synthesis”, Cambridge University Press, 2001

[3] Gero J. S., “Computers and Creative Design,”, in M. Tan and R. Teh (eds), The Global Design Studio, National University of Singarpo, 1996, pp. 11-19

[4] Clement S., Jordan A., Vajna S., “The Autogenetic Design Theory – an Evolutionary View of the Design Process”, International Conference on Engineering Design, ICED 03, Stockholm, August 19 – 21, 2003

[5] Parmee I. C. (ed.), “Engineering Optimization, Special Issue on Evolutionary Search, Exploration and Optimization for Engineering Design”, Taylor & Francis Group, Vol 36, No.2, 2004

[6] Deb K. , “Optimization For Engineering Design: Algorithms and Examples”, Prentice Hall of India, 2004

[7] Koza J, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, The MIT Press, 1992

[8] Kamalian, R.H., Agogino, A.M., and Takagi, H., “The Role Of Constraints and Human Interaction in Evolving MEMS Designs: Microresonator Case Study”, Proceedings of DETC'04, ASME 2004 Design Engineering Technical Conference, Design Automation track, Paper # DETC2004-57462, CD ROM, ISBN # I710CD.

[9] Mortensen, N. H., “Design Modelling in a Designer’s Workbench – Contribution to a Design Language”, dissertation, Department of Control and Engineering Design, Technical University of Denmark, 1999

[10] Summers J., Hernandez N., Zhao Z., Shah J., Lacroix Z., “Comparative Study of Representation Structures for Modeling Function and Behavior of Mechanical Devices”, ASME Computers in Engineering Conf., Pittsburgh, DETC01/CIE-21243, Sept. 2001.

[11] Wang J, Fan Z, Terpenny JP, Goodman ED, "Knowledge interaction with genetic programming in mechatronic systems design using bond graphs", IEEE Transactions on Systems, Man, and Cybernetics, Part C. (to appear)

[12] Fan Z, Wang J, Goodman ED, "Exploring Open-Ended Design Space of Mechatronic Systems", International Journal of Advanced Robotic Systems, 1(4), 2004, pp. 295-302

Zhun Fan
Department of Mechanical Engineering
Technical University of Denmark
Nils Koppels Alle
DTU – Building 404
DK – 2800 Kgs. Lyngby
Denmark
Phone: 45 45 25 62 71
Fax: 45 45 93 15 77
E-mail: zf@mek.dtu.dk