Optimal Synthesis of Compliant Mechanisms using Subdivision and Commercial FEA (DETC2004-57497)

Patrick V. Hull
Engineering Directorate
NASA / Marshall Space Flight Center
Huntsville, AL 35812
Phone:256-544-6562
Email: patrick.v.hull@msfc.nasa.gov

Stephen Canfield
Department of Mechanical Engineering
Tennessee Technological University
Cookeville, TN 38505
Phone: 931-372-6359
Email: scanfield@tntech.edu

ABSTRACT

The field of distributed-compliance mechanisms has seen significant work in developing suitable topology optimization tools for their design. These optimal design tools have grown out of the techniques of structural optimization. This paper will build on the previous work in topology optimization and compliant mechanism design by proposing an alternative design space parameterization through control points and adding another step to the process, that of subdivision. The control points allow a specific design to be represented as a solid model during the optimization process. The process of subdivision creates an additional number of control points that help smooth the surface (for example a C^2 continuous surface depending on the method of subdivision chosen) creating a manufacturable design free of some traditional numerical instabilities. Note that these additional control points do not add to the number of design parameters. This alternative parameterization and description as a solid model effectively and completely separates the design variables from the analysis variables during the optimization procedure. The motivation behind this work is to create an automated design tool from task definition to functional prototype created on a CNC or rapid-prototype machine. This paper will describe the proposed compliant mechanism design process and will demonstrate the procedure on several examples common in the literature.

1. INTRODUCTION

Compliant mechanisms (CMs) are single-part devices that achieve desired force and displacement characteristics through elastic deformation of their structure. These devices hold many advantages over traditional rigid link mechanisms, principally due to the absence of rigid kinematic joints. Several of the advantages that compliant mechanisms have are single-piece construction [1], inexpensive to manufacture [1], and ease of implementation of smart actuation methods [2]. Wide ranges of uses for compliant mechanisms include, smart structures [3], semi-invasive medical surgical devices [4] and micro-electro-mechanical devices [5]. There are several methods for designing compliant mechanisms that have received much attention for the past 10 years [6], with the most common current design methods separable into two categories, topological (or structural) optimization and pseudo-rigid body techniques. This paper will add to the work of topology optimization by proposing a formulation based on a control point parameterization and subdivision that will automate the compliant mechanism design process. The following section will review the current approach and propose an alternate formulation. Some related work in design parameterization and automating the design process is also given.

In the field of distributed-compliance mechanisms, much of the research has focused on developing and modifying suitable topology optimization tools for their design. These tools have grown out of and have built upon the techniques of structural optimization. Of the tools that have been developed to date, the majority share the same general process. They define the design parameters as a discretization of the design space and then relax the binary nature of the discrete design space in order to apply gradient-based optimal algorithms. The relaxation process applied generally uses a ρ-type method (relax on a single parameter) or homogenization method (relax on multiple parameters and then find average constitutive parameters of the more complex description). A gradient-based optimization problem is then constructed based on an objective function with associated constraints which attempts to combine in an optimal fashion both flexibility and stiffness of the CM. The objective function is evaluated using finite element analysis (FEA) on the discretized design space. The sensitivity derivative calculation is expedited with the use of a constant-size stiffness matrix, and lower limits on the relaxation variables to avoid singularities. The research efforts have focused their attention within this procedural framework, modifying components of this process such as the objective function formulation, the method of relaxation, the type of FEA used or the optimization techniques employed.

This paper will build on the previous work in topology optimization by proposing an alternative design space parameterization based on control points and adding another step to the process, that of subdivision. Following the general approach of structural optimization techniques, the processes of objective function definition, finite element analysis (numerical elastic analysis) and appropriate optimal search algorithms are retained, with the design parameterization based on a series of control points over the workspace and the additional step of subdivision added to the procedure. The control points with appropriate boundary information allow a specific design to be represented as a solid model. The subdivision process acts on the design parameter control points to create additional control points prior to solid model definition and objective function evaluation to increase surface feature fidelity. The motivation behind this addition is to suggest an approach that automates the design process from task definition to functional prototype created on a CNC or rapid-prototype. In addition, the process will separate the design variables from analysis variables and allow a high fidelity of surface features defined with a minimal number of design parameters.

There are several parameterization schemes currently used to discretize the compliant mechanism design space, with the most common the direct use of finite element discretization to represent a region of area or volume. Other discretization schemes include a topology description as connecting linkages described in convenient format. For example, Xu and Ananthasuresh [6] use polygons to define a cubic curve that spans the design space of shapes while Tai et all [7] uses a collection of Bezier curves to define the location of finite elements in the design space. Similarly, a number of researchers have addressed the issues of automating the structural optimization design process [8], [9] and [10]. Papalambros and Chirehdast [9], and Marson and Dutta [10] develop specific techniques to create solid-model definitions of an optimal topology that result from optimization based on homogenization. The solid models provide a common description that allow subsequent design and analysis in appropriate CAD tools and automated manufacture of the designs. These techniques generally provide a design translation procedure at the end of the optimization process to transfer the optimal device from a design space representation to a solid-model space form. It is noted that some translation processes are designer dependent. This paper incorporates the translation to solid-model description based on the control point parameterization within the optimization process. This allows the optimization strategy to consider both the effect of design parameters as well as the characteristics of the subdivision process and the mapping to a solid-model representation. Following this approach, the final resulting designs are expected to yield physical results with a high-degree of correlation with the optimal design. It should be noted that certain computational costs are incurred with this approach, but are well within the means of solution by standard processors for compliant mechanisms.

The remainder of this paper will describe the modified topology optimization process that implements a control point parameterization and subdivision resulting in an automated compliant mechanism design tool from task specification to final prototype. This tool will then be demonstrated on several examples common in the literature. The paper will proceed as follows. A discussion of the motivation for an alternative design parameterization and use of subdivision on the topology optimization process is presented in section 2. A description of the proposed problem structure is presented in section 3. Section 4 will discuss additional implementation details resulting from the process. Several examples of implementation follow in section 5. The paper is concluded with a summary of the behavior and nature of the control point parameterization and subdivision process as part of CM topology optimization.

2. CONTROL POINT PARAMETERIZATION AND USE OF SUBDIVISION ON THE STRUCTURAL OPTIMIZATION PROBLEM

The use of control point parameterization and subdivision on the structural optimization process for compliant mechanism design has many implications for the overall CM design process. The benefits it provides to the design process form the primary motivation for this work. A brief description of control point parameterization and subdivision followed by a summary of the impact on the CM design process is presented in the following paragraphs.

The control points that form the parameterization scheme presented in this paper are spatially located over the design workspace and contain the basic information necessary to fully describe the compliant mechanism geometry. Subdivision will operate on this initial set of control points, called the design parameter control points, to create additional control points that add fidelity to the specific design. Subdivision is a geometric tool used to define a smooth curve or surface as the limit of a sequence of successive refinements [11]. At its most early stages of development, in the late 1940's, it was proposed as a "corner-cutting" technique to describe smooth curves. More formally, subdivision in its current form was developed by Catmull-Clark [12] and Doo-Sabin [13] and since then has found wide application in computer graphics and computer assisted geometric design. Subdivision has several characteristics that make it well suited for solid-body modeling. For example, subdivision generalizes the classical spline patch approaches to arbitrary topology. It provides a systematic scheme of interpolation to construct higher-definition spaces through piecewise linear curves. It provides a means to describe complex, continuous shapes with a minimal number of design parameters (see for example Fig. 1). It also provides a procedure that can remove some singularity conditions while preserving the integrity of an original design. For these and other reasons, subdivision has also recently found use in other areas of engineering for example in computer-assisted engineering [14].

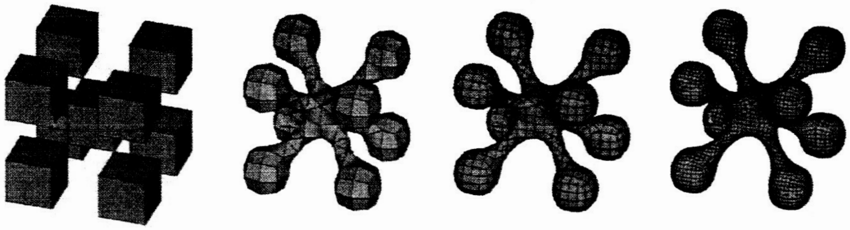

Figure 1: Subdivision process in defining a smooth surface from minimal number of parameters

The control point parameterization chosen facilitates conversion of a set of design parameters (information about the control points) to a solid model in a standard format for subsequent analysis or manufacture. This allows a different characterization for the designs to be specified in both the optimization process and in the analysis or manufacture process, allowing the most suitable characterization to be chosen for each case. In the design variable space, the control points provide a means for describing the topology, shape and even size of a potential CM design. On the other hand, the solid model description in standard format is readily accepted by commercially available FEA, CAD and CAM software. The potential CM design in its solid model form can be described with a high level of resolution or definition by use of subdivision. However, the design still retains its description in terms of the original control points. Based on the state of control points and the subdivision process, a direct and unique mapping exists from design variables to the solid model design representation (a unique mapping from design solid-model representation to control points does not necessarily exist). This mapping allows a high degree of definition in the solid model, sufficient to fully describe a design through manufacture with relatively small number of design parameters. Note that the choice of control points as design parameters is not necessary for conversion to solid models, but is commonly used in solid modeling and facilitates the use of subdivision.

With the mapping defined which results in a solid model form of the design, direct use of commercial code for analysis is readily implemented. This allows the use of automatic mesh generators and potentially a more efficient modeling of the material elastic behavior. The mapping can be defined to perform constraint management, i.e., to correct the numerical difficulties that could occur in the design parameterization. Such difficulties include checkerboard patterns and artificially high or low-stress regions in the design. Further numerical difficulties can now be avoided with the use of adequate fea meshing and analysis throughout the design process [15].

The subdivision process adds an additional layer of design variables to the problem that can either be specified or included as added design parameters. If using a standard subdivision technique, the simplest layer of design variables will consist of the type of subdivision process. For the following work, the type of subdivision and the number of levels of subdivision are fixed and therefore are not included in the design parameters. Future work may consider the effects of variation in the subdivision process.

Additional consequences of the separation of design variables and analysis variables via subdivision exist. This formulation results in a stiffness matrix form that is now design dependent, increasing the cost in performing sensitivity derivatives calculation. In addition, the control point design parameters as originally envisioned exist in a finite number of discrete states. Due to these observations, the use of non-gradient based search methods may be better suited to this problem formulation. These non-gradient-based searches readily permit linear or nonlinear analyses, specific mesh density (no need for variable relaxation), and readily address the issue of non-convex objective functions, for example through the use of a genetic algorithm (GA). The computational cost of using non-derivative-based searches is relatively higher, and in many instances considered prohibitive. For example, some have focused on taking specific actions to avoid the cost of solving the discrete mathematical programming problem while searching for a more discretized result [16]. On the other hand, the use of evolutionary algorithms for the CM problem and mechanical shape design has been investigated with suitable results by a number of researchers [17],[18],[19],[20],[21],[23] and [24] and will be considered an appropriate approach for this work.

3. PROBLEM FORMULATION

The application of control point parameterization and subdivision to the topology optimization problem will be presented in this section. A diagram of the general form of the current process and the modified procedure proposed in this paper are shown in figures 2 and 3 respectively. The proposed changes to the process are indicated in figure 3 with shaded boxes. The proposed process will consist of the following steps; design parameterization based on control points, objective function definition, subdivision, objective function evaluation via finite element analysis and optimization based on evolutionary algorithms. The entire procedure will be reviewed, with the usual steps briefly described, and the additional steps of parameterization and subdivision more thoroughly discussed.

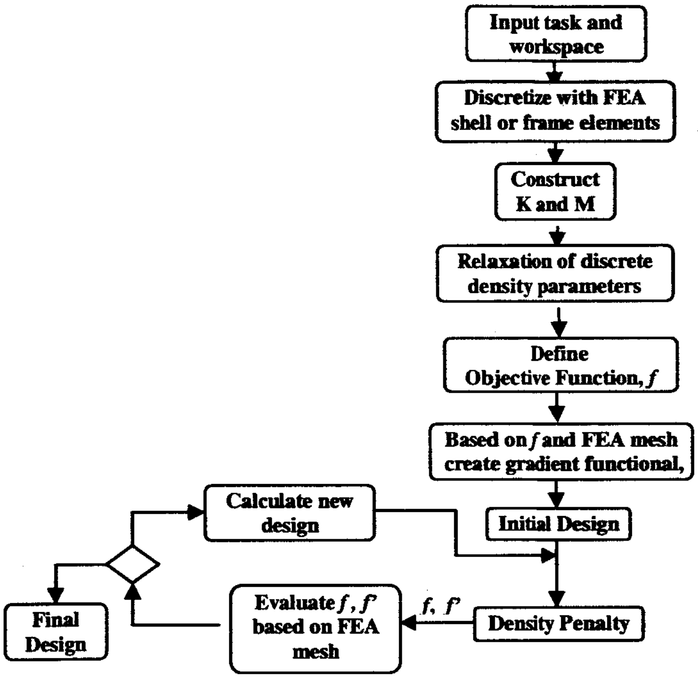
Input task and workspace
Discretize with FEA shell or frame elements
Construct K and M
Relaxation of discrete density parameters
Define Objective Function, f
Based on f and FEA mesh create gradient functional,
Calculate new design
Initial Design
Final Design
Evaluate f, f' based on FEA mesh
f, f'
Density Penalty

Figure 2: Current state of the art approach for designing distributed compliant mechanisms

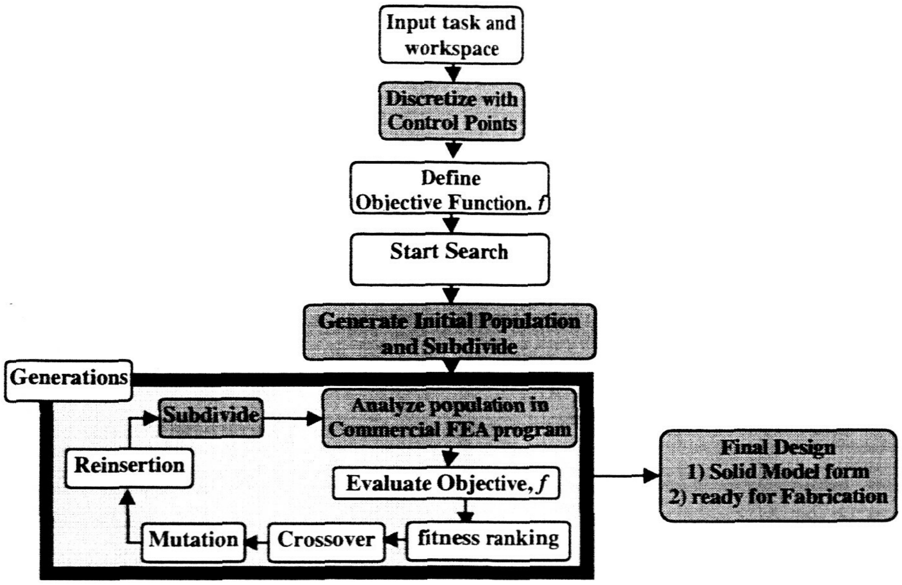

Figure 3: Outline of proposed compliant mechanism optimal synthesis tool

Problem Structure

Following the usual approach, a general continuum representing the maximum allowable material space for the desired compliant mechanism is acted on by an input force in order to provide some output reaction on a workpiece. Appropriate modeling of the workpiece is added to capture its effect on the mechanism, for example as a linear spring (fig. 4). An additional number of suitable constraints are added to hold the mechanism in place.

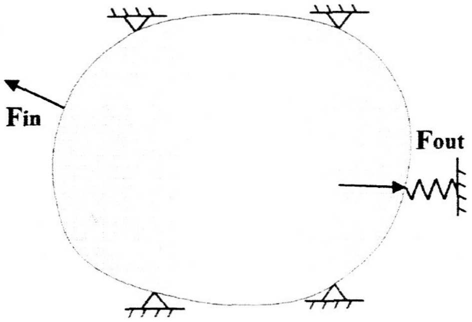

Figure 4: Design domain and problem parameters

Design Parameterization

The control points that form the parameterization scheme presented in this paper are spatially located over the design workspace and define the design space as a series of control meshes. Since a quadrilateral-based subdivision process will be implemented, quadrilateral control meshes are defined consisting of groups of four adjacent control points. This is shown in fig. 5 for a regular grid of control points and resulting control meshes. If a control point is on the interior of the design space, then it is used to define four control meshes. If the control point is on the exterior of the design space, it will be used to define two control meshes (one if it is a corner exterior node). If the *n* control points are distributed uniformly with equal number of rows and columns, the number of control meshes, *m* is given as $m = \left(\sqrt{n} - 1\right)^2$. The control points contain information including position coordinates, material properties and existence of the dependent control meshes. Note that in the optimization of this paper, only the existence of the control meshes will be varied. Based on this control point and resulting control mesh parameterization, the optimization problem can be described as,

$$\min \quad f(r(\mathbf{q})) \qquad \text{Eq. 1}$$

subject to:

$$r_i = \begin{cases} 0 \\ 1 \end{cases}, \quad \forall i, \quad i = 1 \cdots n$$

where **r** is the vector of control meshes which directly depend on the control points **q**, and can assume a state of zero or one corresponding to existence/nonexistence in the design space.

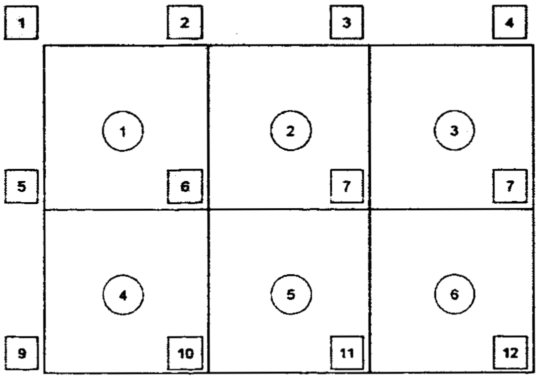

Figure 5: Control points (numbered in boxes) and control mesh (numbered in circles)

Subdivision

This section will briefly describe the subdivision process and its implementation in the compliant mechanism optimization process. A thorough treatment of subdivision is beyond the scope of this paper but can be found in sources such as Sorin and Schroder [11]. Subdivision, acting on an initial control mesh (defined by the initial control points) will be used to define the compliant mechanism surfaces. These subdivision surfaces are defined as the limit of an infinite refinement process. The process starts with the initial control points which form a minimal set of parameters that uniquely define the final surface according to the subdivision process. Additional control points are defined locally at successive iterations using the relation,

$$\mathbf{p}^{j+1} = \mathbf{S}\mathbf{p}^{j} \qquad \text{Eq. 2}$$

where **p** is a local set of control points (a subset of the control points **q**) at different levels of subdivision, j, $j+1$ and **S** is the subdivision matrix. Thus, the subdivision matrix **S** in a simplistic sense defines the subdivision process. The size of matrix **S** determines the degree of local effect of the curve (the number of local control points involved in defining an updated set of control points). The region of surface mapped by the local control points $\mathbf{p}^{j+1}$ is smaller than that of $\mathbf{p}^{j}$ for the subdivision scheme used here but results in a greater number of points overall to define the refined surface. The speed of convergence of successive subdivisions is geometric and produces reasonably refined surfaces after just a few steps.

In the subdivision scheme, the process starts with an initial control mesh, defined by the set of initial control points, $\mathbf{q}^{1}$ (from which sets of local control points, $\mathbf{p}^{1}$ are extracted). The scheme applies its rules, for example as in Eq. 2, in a recursive manner to end up with a final set of control points $\mathbf{q}^{k}$ defining the desired surface. Note that each recursive step produces a greater number of control points (and doubles the number of possible control meshes) such that the size of $\mathbf{q}^{i+1}$ is greater than $\mathbf{q}^{i}$ and $\mathbf{q}^{k}$ is much greater than that of $\mathbf{q}^{1}$. The initial control mesh, $\mathbf{q}^{1}$ provides the design parameter space, while the final control mesh, $\mathbf{q}^{k}$ describes the solid-model space.

The rules defined for subdivision represented by Eq. 2 can be applied only to regions of the mesh that are locally regular, i.e., that have the expected valence (number of lines connecting a given control point to other control points). Additional rules have to be defined for regions that are not regular, such as boundaries and the like, these rules are discussed further in appendix A. However, these rules follow the form of Eq. 2. Note that the addition of these rules allows for "constraint management" and can permit subdivision rules that eliminate singular connections that may exist in the initial control mesh.

A wide variety of subdivision schemes are available, with a number of the more common rules categorized in [11]. For this work, the Doo-Sabin/Midedge scheme is selected with a quadrilateral element meshing (initial control mesh defined by groups of four control points). This scheme is a stationary subdivision scheme (one in which the subdivision rules are held constant throughout the process) generating C^1 continuous surfaces on arbitrary meshes. The Doo-Sabin [13] scheme is a vertex split scheme. In the vertex split scheme, several new vertices are created for each old vertex, with new vertices retained and old vertices discarded. Therefore, vertex split schemes are approximation schemes (vs. interpolating schemes) and will not necessarily retain points in the initial control mesh. Approximation schemes however tend to converge to a limit surface more quickly than interpolation schemes. The vertex split scheme when applied to quadrilateral elements results in new control points with valence of four. The subdivision scheme used in this paper is adapted from [11] and is presented in Appendix A.

The Doo-Sabin/Midedge subdivision scheme adapted for compliant mechanism topology mapping is shown for several examples in figs. 6-8. In each figure, both the initial control-point mesh (corresponding to the topology design space, labeled A) and the second level control point mesh (corresponding to solid model in this application, labeled B) are shown. Note in each figure the singular conditions that exist in the design space are eliminated in the solid-model space. One can also observe the addition of control points and control meshes from initial to second level subdivision, (and that further levels of subdivision would result in a more refined final surface). It is noted that this scheme applies equally well to planar as well as spatial workspaces. Finally, it should be noted that the solid model that results from the design space (initial control mesh) is highly dependent on the subdivision scheme chosen.

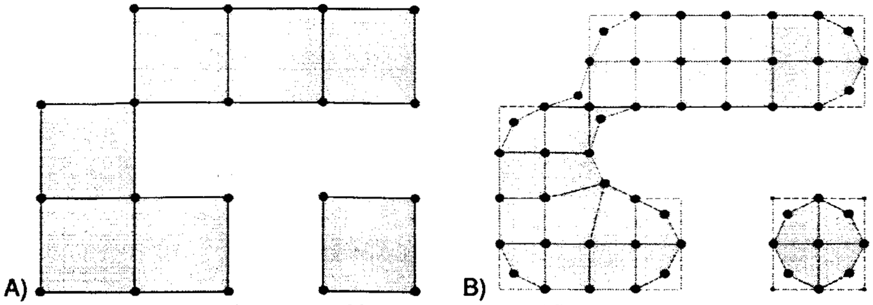

Figure 6: A)Singularity condition created by two blocks and B)subdivision to recapture the shape

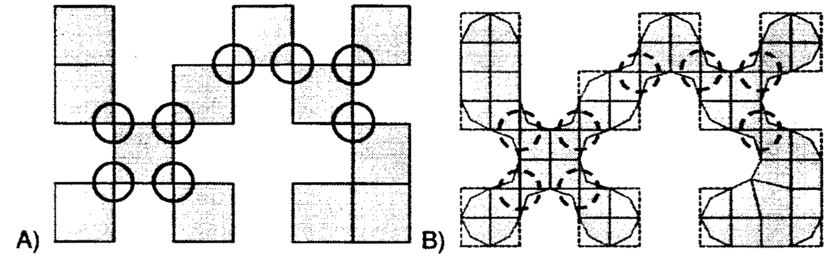

Figure 7: A) checkerboard mesh topology, B) subdivided checkerboard mesh topology

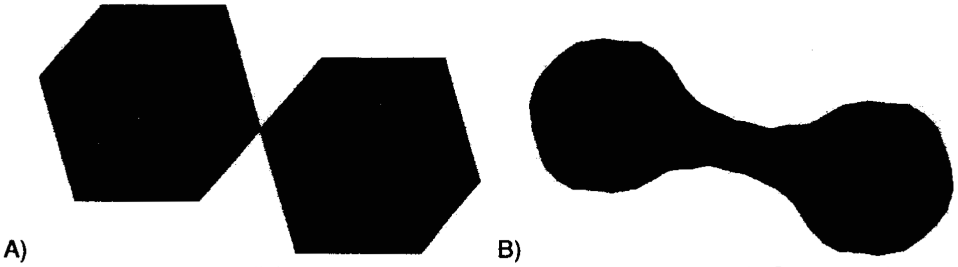

Figure 8: example of a 3D subdivided shape A) 3D control mesh formulation, B) Subdivided control mesh

Objective Function Definition

The fitness or objective function for this problem will follow the precedent of designing for flexibility as well as stiffness [22]. While many objective functions have been defined and could be implemented, a measure defined in Parsons and Canfield [20] is applied here. For this function, a measure of flexibility is defined as the total work in the output spring, W_o

$$W_o = sign(u_o) * \frac{1}{2} * k_s * u_o{}^2 \qquad \text{Eq. 3}$$

where u_o is the scalar output displacement in the direction of the spring, k_s is the spring stiffness. The measure of stiffness will be defined as the strain energy in the system, W_i

$$W_i = \frac{1}{2} * \mathbf{u}^T * \mathbf{K}_{tot} * \mathbf{u} \qquad \text{Eq. 4}$$

with **u** the vector of nodal displacements and the $\mathbf{K}_{tot}$ the stiffness matrix representing both the mechanism and the external springs. The objective function that maximizes the output energy while minimizing total strain energy in the mechanism is then given as,

$$f = \frac{sign(u_o) * \frac{1}{2} * k_s * u_o^{\ 2}}{\frac{1}{2} * \mathbf{u}^T * \mathbf{K}_{tot} * \mathbf{u}} \qquad \text{Eq. 5}$$

with *f* the fitness or objective function.

Objective Function Evaluation

The objective function is evaluated as the solution to the elastic boundary-value problem using commercial finite element analysis software. This work employed ANSYS 8.0 for all finite element analysis. The fea software is called as part of the optimization routine and is sent information on a specific design to be modeled in solid model form (a surface defined by the collection of the final control meshes). It is also sent pertinent loading and boundary conditions. Automatic meshing using a 2-D 8-node structural solid element is specified, with the smart-sizing capabilities in ANSYS employed. An elastostatic finite element analysis was performed on each design, implemented in ANSYS by a batch file written in MATLAB. Upon completion of the pre and post-processing stages within the FEA software, ANSYS returns solution information that includes nodal coordinates, primary stresses and nodal displacements. Using this information, the fitness function value is determined for that particular design. While the analysis of a single member of the population is performed very quickly, the fitness evaluation of a population (rather than a single point in the design space) is required by the genetic algorithm and forms a primary source of the computational costs of this approach. However, the problem lends itself readily to being processed in a parallel fashion, in the case where multiple seats of the fea software are available. The nature of this problem therefore could reduce some of the negative factors in this problem, the long computation time, by providing a simple means to easily harness a greater amount of computational resources.

Reachability

The design parameterization based on discrete states and/or existence of specific control meshes allows significant variation in the geometric description of the compliant mechanism. This parameterization could also result in multiple, disconnected regions in the design space, or even a disconnect of the mechanism between the input, output and necessary boundaries. In general, the existence of disconnected regions does not pose any difficulties as long as the mechanism demonstrates "connectedness" (a structural material path between the input, output and necessary boundaries). These disconnected regions do not affect the objective function evaluation in general and are ignored by the fea process. It should be noted that these disconnected regions are retained in the design member information however, as they may provide useful and necessary information at a later stage of the optimization process. However, connectedness of the mechanism is essential and a check for such on each member of the design population is performed using tools from graph theory [23]. This process will consist of defining a reachability matrix and then simply verifying the connectedness of specified positions in the design from this matrix. The reachability matrix is an *mxm* matrix where *m* is the number of initial control meshes (pre-subdivision) and is constructed through a series of matrix multiplications as shown in [23]. Reachability analysis is performed first in generating an initial population of connected members and then is applied on each member resulting from a crossover. If a design is not connected, it is assigned a poor fitness value and skips the fea analysis.

Optimization via Genetic Algorithms

Genetic programming is well suited in many ways to the optimal design of CMs and will be used in this algorithm. As a guided random search, genetic algorithms avoid the need for gradient derivations, allowing much greater freedom in the selection of objective functions and the use of linear or nonlinear finite element analysis tools. It also allows ready implementation of the automatic meshing tools available in commercial finite element analysis software. Genetic algorithms are also capable of handling both convex and non-convex objective functions [17]. The genetic algorithm (GA) returns a family of solutions that gives the designer greater freedom in selecting the final design. Genetic algorithms can be combined with other search tools, often providing solutions for final refinement by a specific optimization tool. Finally, genetic algorithms have the ability to perform multi-objective optimization as required by a general design tool, [20]. The use of genetic algorithms can also pose some limitations. In general they have a relatively high computational cost. They are non-deterministic in nature and have a convergence behavior that is difficult to predict.

In initiating the genetic programming application, each design is assigned a blueprint of its characteristics, encoded in a binary string called a chromosome. The chromosome consists of a combination of genes. Each gene corresponds to one of the quadrilateral control meshes defined by the control points and has two distinct states (represented as a 0 or 1). Therefore, each gene requires one bit and the chromosome contains m bits with $m=\left(\sqrt{n}-1\right)^2$ the number of control meshes and n the number of control points in a square grid. Following the evolutionary precedent, only the fittest designs contribute largely to future generations, while poorly suited designs are removed from the population. Occasional mutation of individual traits occurs throughout the process, ensuring diversity in the gene pool allowing designs to continue to evolve. The cycle of breeding, mutation, and attrition continues for as many generations as specified by the user. In the end, the designer is provided a group or family of good solutions rather than a single solution.

The general approach to GA implementation proceeds as follows. The genetic algorithm toolbox for Matlab created at the University of Sheffield is employed [25]. This toolbox has built-in functions for processes in genetic algorithms including fitness evaluation, cross-over, mutation, and generation recombination. The fitness value resulting from the objective function evaluation is acted on by a linear transformation to ensure a positive, representative value, with improved designs having smaller fitness values. A ranking or selection of the individuals for reproduction is based on a stochastic sampling process with ability to apply linear or non-linear ranking. The recombination routine allows multiple forms of cross-over including single or multi-point. The mutation function provides both binary and integer mutation. A reinsertion function allows the use of an elitist strategy, one that allows the most-fit individuals to propagate through successive generations, while maintaining a constant size in the population. The genetic algorithm proceeds through this cycle of fitness evaluation, cross-over, mutation, and generation recombination to create successive generations with potentially improved characteristics.

The stopping criterion is generally specified as a maximum number of generations. Genetic algorithms are widely regarded as a global optimization method, given sufficient population size and mutation rate. However, it is difficult to guarantee that a global solution has been located and in part rests with the designer. In this work, a combination of large population, sufficient mutation rate and stochastically selected initial states was used to help guarantee a global optimum. Additionally, a hybrid GA can be used, as discussed by Dozier et al [26], in which the GA is first employed to get a solution a the region of a global optimum, and then switches to a gradient-based technique to find that optimum.

Problem Formulation Summary

Based on these components, a general procedure for topology optimization is created. An initial population is selected at random over the design workspace, with each design guaranteed to be feasible (reachability satisfied). The evolutionary processes of fitness evaluation, cross-over, mutation, and generation recombination then proceed to guide the selection of new generations of solutions. At each stage, subdivision and objective function evaluation are performed for each member of the population. Note that population members that do not produce feasible results are assigned a base-line fitness value, but are not necessarily removed from the population. This process continues for a specified number of generations and results in a population of optimal topologies for the compliant mechanism. The designer is free to choose from this final population or to use one member as a starting point for a gradient-based optimization.

Design tool implementation

The current implementation of the CM design tool is briefly described here. The designer interface and supervisory algorithm is implemented in Matlab. The Matlab routine collects all problem-specific information from the user, and then schedules the execution of the problem processing phases. New functions for the subdivision procedures are created based on the modified Doo-Sabin/Midedge algorithms. The genetic algorithm optimization routine is implemented directly in Matlab by making use functions available in the genetic algorithm toolbox [25]. Elastic analysis of potential CM designs is performed using ANSYS, with Matlab scheduling these tasks and passing and receiving the information necessary to perform these analyses. Finally, post-processing capabilities are provided through a combination of Matlab routines as well as post-processing capabilities in ANSYS and include plotting the results or sending the results to a rapid-prototyping machine.

4. EXAMPLE DESIGNS

This section demonstrates the use of subdivision as part of the compliant mechanisms design optimization tool. Provided next is a demonstration of several examples that are common to the CM design literature; the compliant inverter, gripper and inverted compliant gripper. These examples will demonstrate the implementation of the optimal design tool described in this paper. As part of the GA-based optimization routine, a population of solutions result at the end of the optimization process. The solution with the best objective function value from this final family is shown for each example.

Compliant Inverter I

The first example will consider the compliant inverter mechanism demonstrated by Yin [27]. The design region with input and output is depicted in Fig. 9. The design parameters given for this problem are the design space size = 15cm *x 15cm*; control points equally spaced 0.5 cm in both x and y dimensions, the material thickness, $t = 0.3\ cm$, modulus $E = 2MPai$; the input force, F_{IN} = 200N and the FEA element type selected is an 8 node structural solid. A symmetric half of the design space is considered as in fig. 9b. The GA parameters are defined as single point crossover, 60% crossover rate and a 0.5% mutation rate. The problem is started with an initial population of 50 members and is run for 200 generations. This search was performed on a 700MHz Linux box with a total optimization time of 10 hrs., with approximately 6,000 design evaluations. The optimal compliant gripper topology is demonstrated in Fig. 10 and the history of convergence for this problem is shown in Fig. 11.

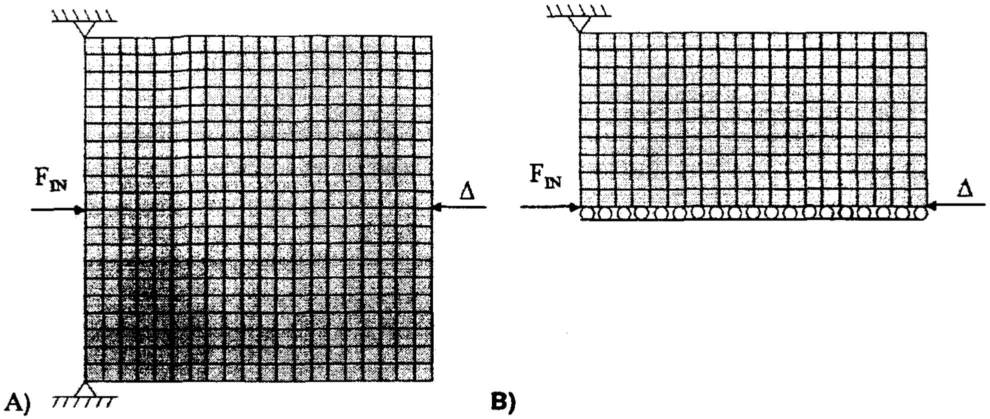

Figure 9: Initial setup for gripper problem, BC's, force locations and finite areas

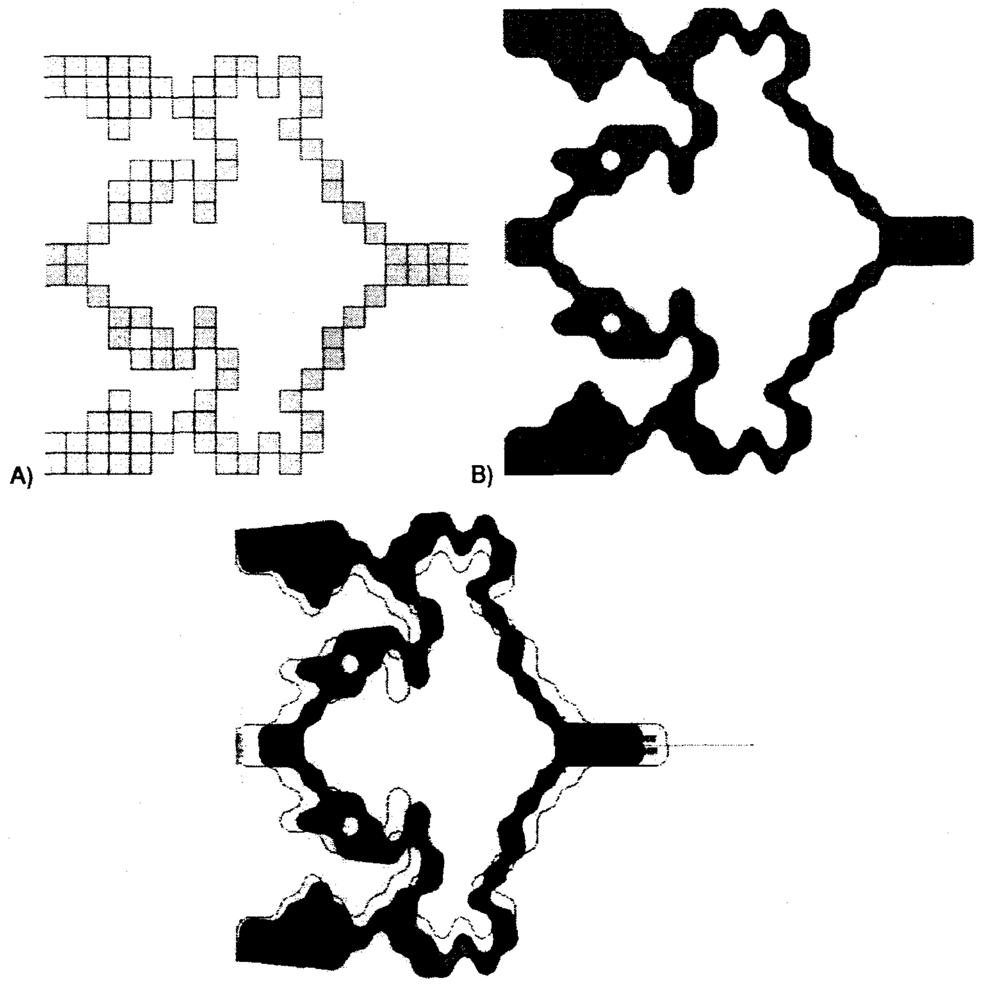

Figure 10: A) Optimum initial control mesh solution, B) Optimum subdivided solution and C) Optimum subdivided, FEA meshed and deformed solution

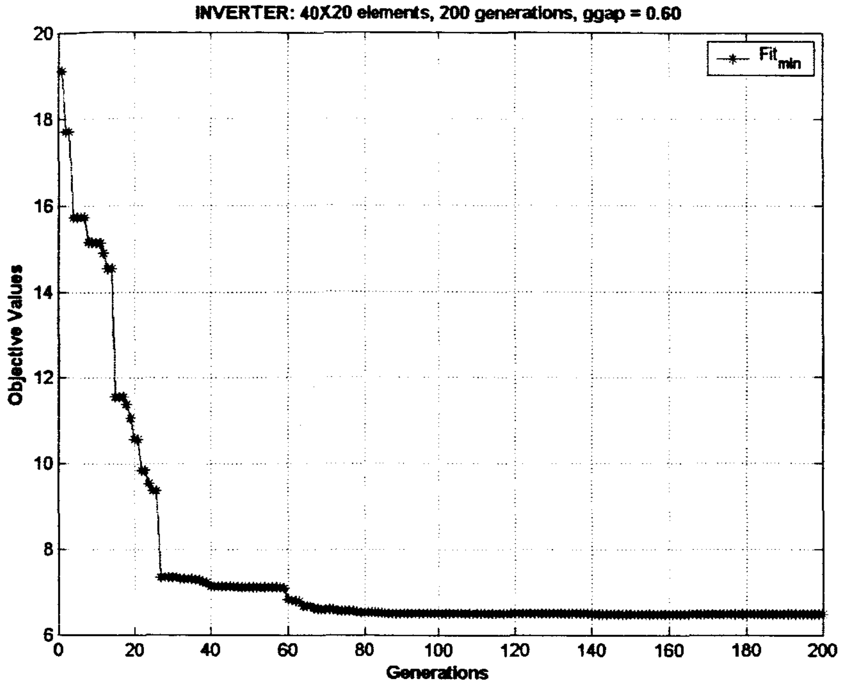

Figure 11: Optimization of a inverter problem with minimum objective values

The convergence of the GA optimization for the compliant gripper occurs at about 80 generations. The design was permitted to evolve further to allow mutation effects to possibly find a better solution. The resulting topology of this example is comparable with that given in the literature.

Compliant Inverter II

The second example is based on the inverter given by Saxena 0. The design domain for the compliant inverter is shown in Fig. 12 with pin boundary conditions in the upper and lower left hand corners, the input a displacement to the left and output a displacement of the spring to the right. This problem is discretized with control points and control meshes as shown in fig. 12. The input parameters given for this problem are; the *design space size = 12cm X 12cm* control points equally spaced at 0.6 cm in x and y directions, material thickness, *t=0.1cm*, modulus *E=2MPa,* input displacement *$Disp_{in}$=0.5cm*, output spring stiffness, *K=10N/cm* and fea element type selected = 8 node structural solid. The GA is initialized with a single point crossover, 60% crossover rate, 0.5% mutation rate, 50 member initial population and is run for 375 generations. This search was performed on a 700MHz Linux box with a total optimization time of 18 hrs., approximately 11,200 designs were evaluated. The results from this problem are shown in Fig. 13

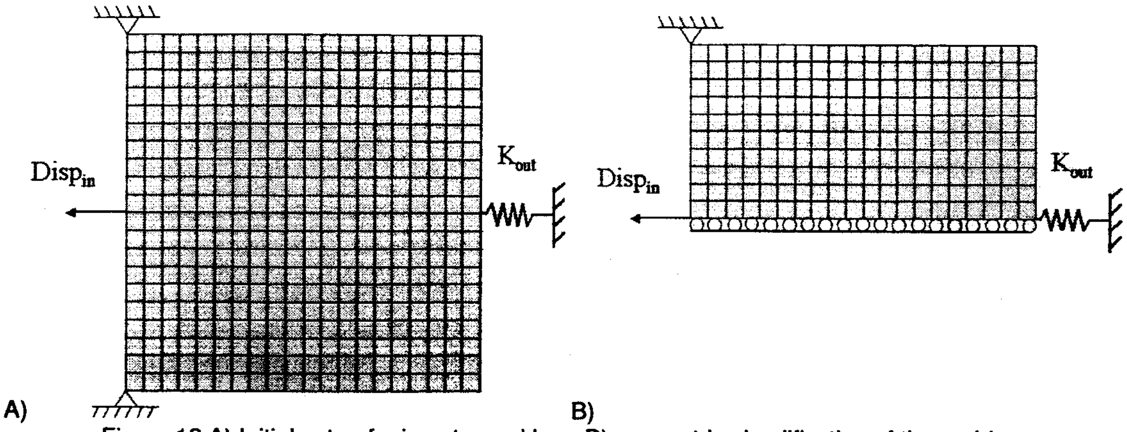

Figure 12 A) Initial setup for inverter problem, B) symmetric simplification of the problem

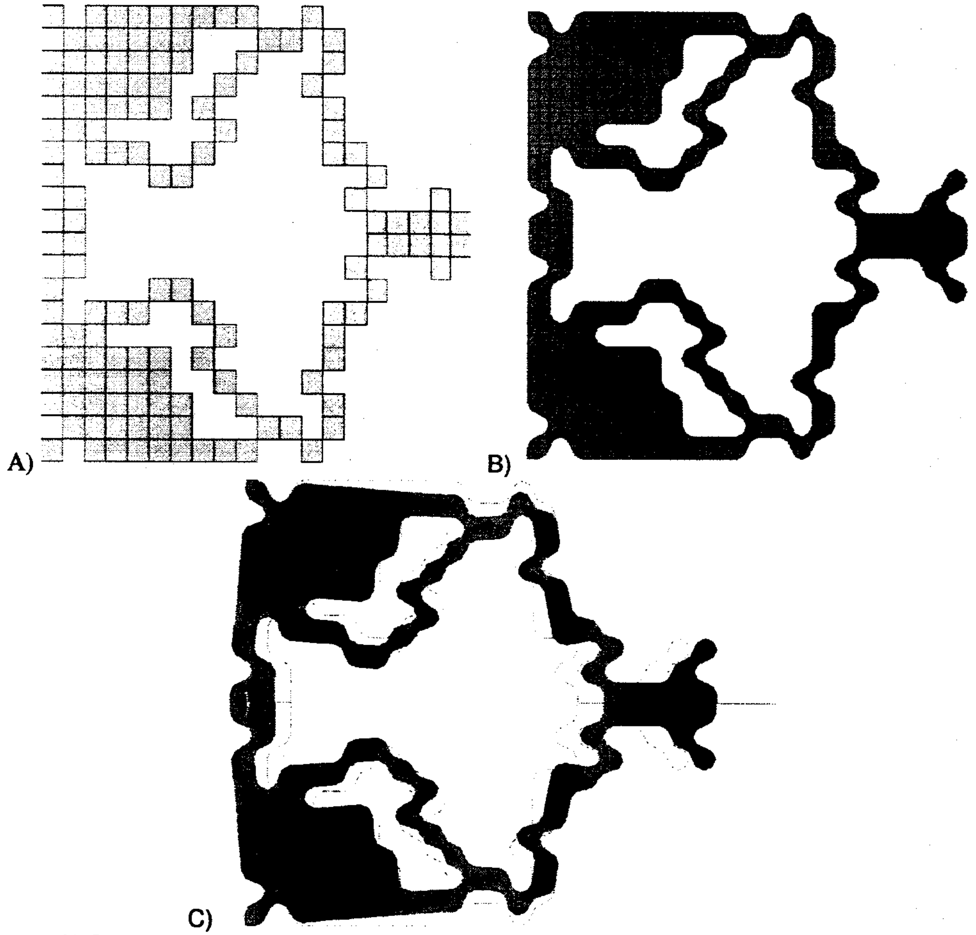

Figure 13: A) Optimum solution in initial control mesh, B) Optimum subdivided solution and C) Optimum subdivided, FEA meshed and deformed solution

The minimum objective function history plot during the generational step is shown in Fig. 14.

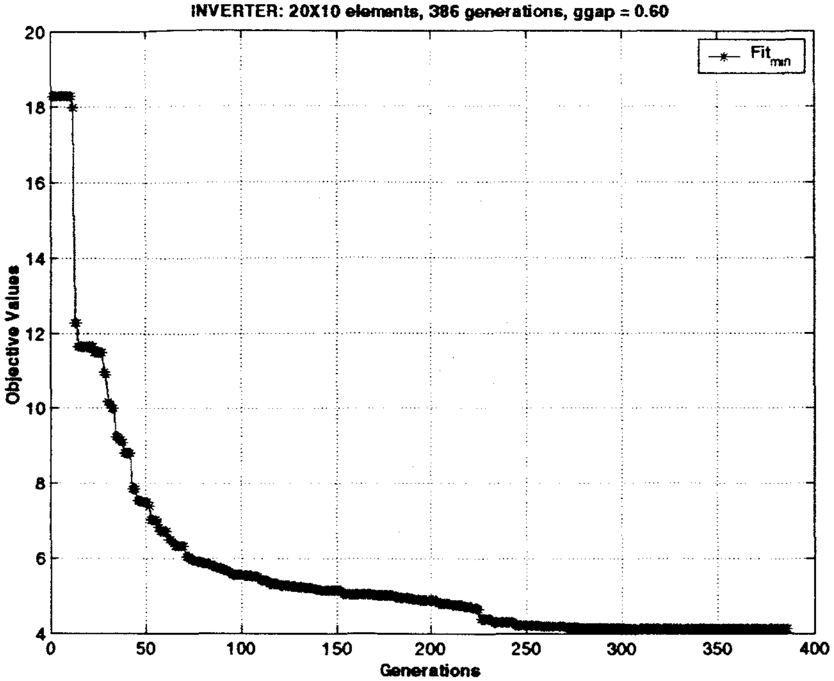

Figure 14: Optimization of an inverter problem with minimum objective values

Convergence of the minimum objective value occurs at approximately 275 generations. This example has been quantitatively compared with the results from Saxena 0 with deflection results comparing to within ± 10%.

Compliant Gripper

The third example is a gripper problem similar to the example given by Hetrick and Kota [28]. The initial design region is defined in Fig. 15. The design parameters given for this problem are; the design space size = 12cm x 12cm, control points equally spaced 0.6 cm in x and y directions, material thickness, t=0.1cm, modulus E=2MPa, input displacement, $Disp_{in}$ = 0.5cm, output spring stiffness, K=5N/cm and fea element type = 8 node structural solid. This problem, like the compliant inverter will be solved symmetrically, as shown in Fig. 15b, for topology optimization. The GA is defined with a single-point crossover, 60% crossover rate, 0.5% mutation rate, 50 member initial population size, and is run for 400 generations. This search was performed on a 700MHz Linux box with a total optimization time of 18 hrs, approximately 12,000 designs were evaluated. The resulting compliant gripper design is shown in Fig. 16 and its convergence plot is shown in Fig. 17.

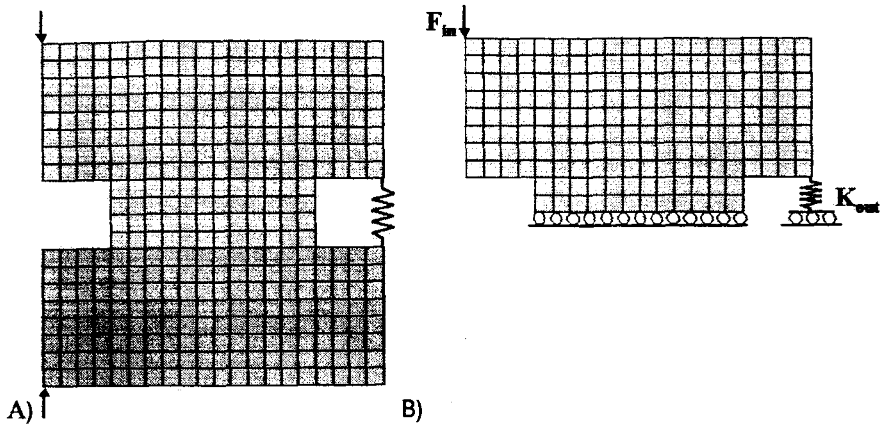

Figure 15: A) Initial setup for inverter problem, B) symmetric simplification of the problem

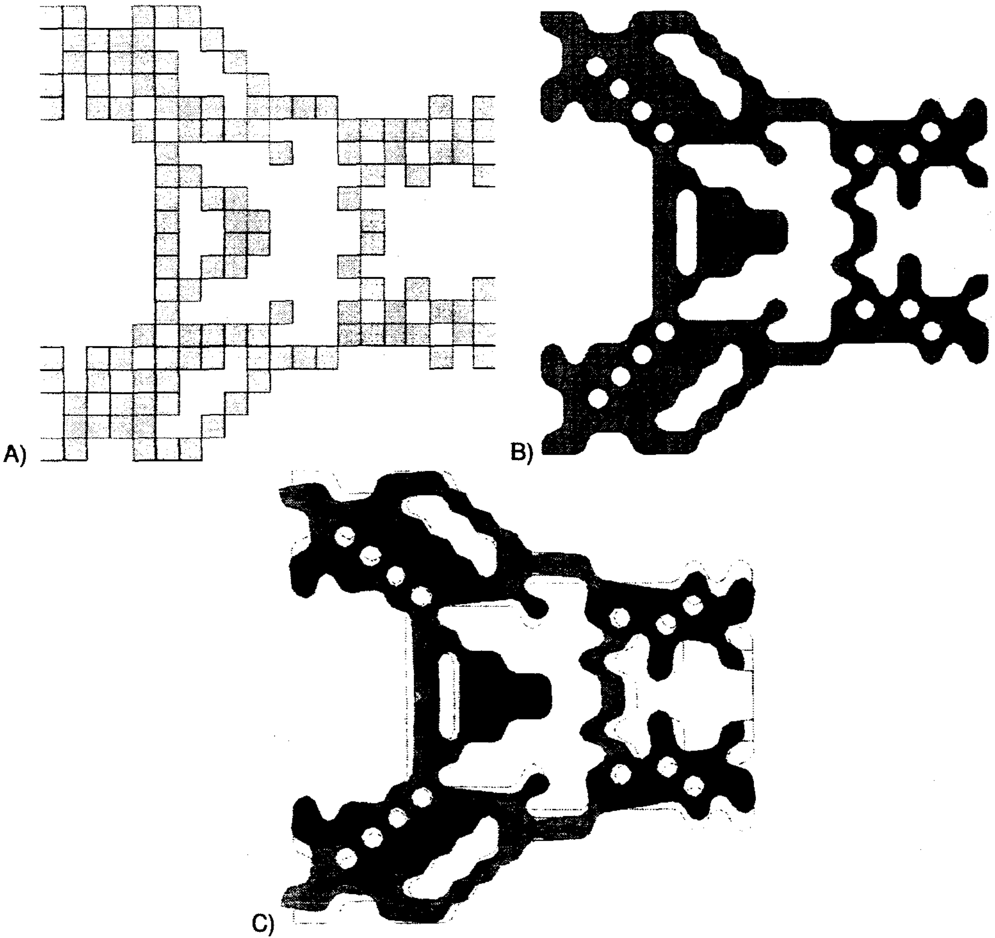

Figure 16: A) Optimum solution in initial control mesh, B) Optimum subdivided solution and C) Optimum subdivided, FEA meshed and deformed solution

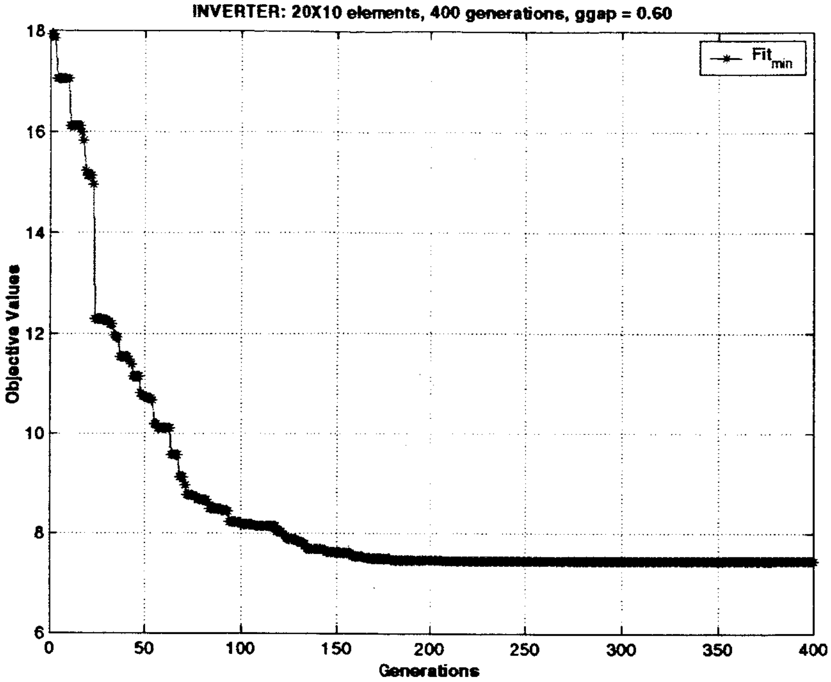

Figure 17: Optimization of a gripper problem with minimum objective values

The solid model of this example was then used directly (absent of designer intervention) to manufacture a prototype, shown in fig. 18, of the compliant gripper on a CNC mill. The material used in this prototype is polyethylene. Notice this gripper does have a variation of bulky and thinned sections as well as extra material or wavy sections. This can be explained in part as follows. Excess material is not culled by the objective function defined in this process, while wavy sections may be introduced to reduce axial stiffness of a given section. The results of this example are compared quantitatively with experimental results on the prototype as shown in figure 19 to yield reasonable agreement.

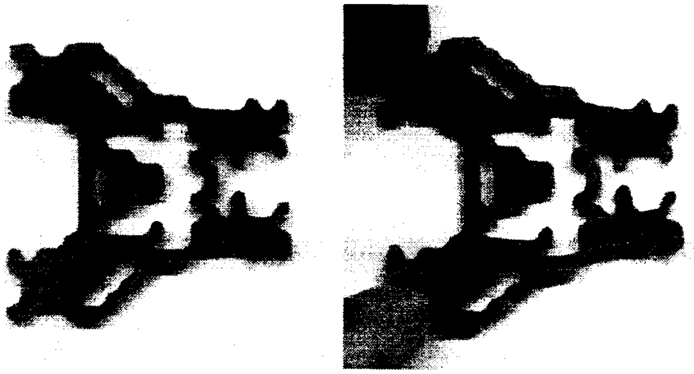

Figure 18: Prototype of compliant gripper

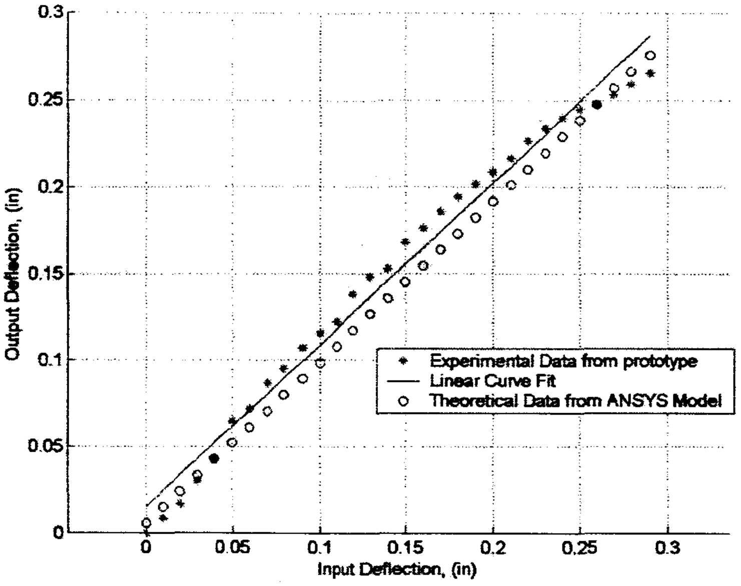

Figure 19: Comparison of fea model and prototype response

5. DISCUSSION AND CONCLUSIONS

This paper has presented a modification to the standard topology optimization problem for compliant mechanisms through a design parameterization based on control points and the addition of the process of subdivision. The problem formulation that includes these steps is described in detail in this paper. The algorithm is incorporated using a combination of commercial FEA software, genetic algorithm toolbox for Matlab, and a modified subdivision algorithm implemented as a function in Matlab. The procedure is then demonstrated on several problems common for compliant mechanisms, demonstrating a reasonable correlation with historical and experimental results. A general discussion of the utility and applicability of this modified approach is considered here followed by concluding remarks on the process implementation.

As noted earlier, the design parameterization on control points, the use of subdivision and description of designs as solid models separates design variables from analysis variables during the optimal synthesis procedure. This results in two descriptions for each design candidate; a description in design space and a description in solid-model space. The representation of the design space as control points is similar in many ways to the approaches that discretize using quadrilateral elements. The design parameters contain information on position, material properties and boundary classifications. Only variation in existence of control meshes defined by the control points is considered in the optimization process in this paper. Alternatively, the solid-model space for each design provides a standardized description of the design that is readily acceptable by commercial engineering software. The use of commercial FEA is demonstrated in the above examples. Other analysis tools or automated manufacture could also be easily employed with the solid-model space description of this design. The result is that this technique provides an approach for automating the entire development process from task statement to manufacture of a compliant mechanism prototype. Other approaches for automating the entire development cycle have been presented [8], [9] and [10]. These techniques generally provide a design translation procedure at the end of the optimization process to transfer the optimal device from a design space representation to a solid-model space form. The mapping in this paper is incorporated within the optimization stage, therefore allowing the optimization strategy to consider both the effect of design parameters as well as the characteristics of the subdivision process and the mapping to a solid-model representation. Following this approach, the final resulting designs are expected and demonstrated to yield physical results with a high-degree of correlation with the optimal design

The mapping defined from design-space to solid-model space during the optimization process could be performed several ways. In this paper, the process of subdivision is suggested with a modified Doo-Sabin subdivision scheme implemented in the topology synthesis algorithm. The choice of mapping will influence the entire optimization procedure as well as the resulting designs. Some examples of the influence from the mapping presented here are considered. A properly defined subdivision process can reclaim some numerical instabilities that may exist in the design space, for example checkerboard instabilities in the design. Therefore, the absence of singular point flexures is guaranteed, avoiding an optimization loophole as discussed by Yin and Ananthasuresh [27].

The mapping applied in this paper, a modification of the Doo-Sabin scheme, is a vertex split or approximation scheme; the original control points are not retained in defining the final control mesh or the design in solid model space. Therefore, additional rules must be added to the scheme to retain boundary conditions as specified in the original problem set-up. The discrete nature of the design variables and the mapping provided by subdivision can increase the complexity of applying relaxation to the design space variables; an additional dimension to the mapping in this instance would be required. It should be noted that this difficulty in spatial mapping is mirrored by the difficulties in manufacturing a design with relaxed design parameters however. This work has addressed this difficulty by using a non-gradient-based optimization procedure resulting in a computationally expensive process. However, the examples demonstrate successful implementation of the process with reasonable run time. The non-relaxed control mesh design variables are represented as discrete values in a coding well suited to the genetic algorithm. This binary representation of the control point design variables creates the need for an additional check to be performed on each potential design. In the binary state, a large number of possible solutions are not valid due to a lack of connectivity between the input, output and necessary boundary conditions. Potential solutions are checked for connectivity (thus providing feasible solutions) using the reachability matrix. It should be noted that two feasible solutions could yield non-feasible offspring in the cross-over process. Therefore, a combination of past and current generations is retained in each generation to guarantee a constant population size. Such an approach also guarantees that the best solution in each generation will be equal to or better than the optimal solution from the previous generation. Finally, it is noted that mesh dependencies are still evident in this approach. This is expected, as mesh-dependencies are a function of the design parameterization, which is retained in the design formulation.

ACKNOWLEDGMENTS

This project was instigated with the encouragement and support of several individuals at NASA/Marshall Space Flight Center. We would like to extend our gratitude to the following individuals at MSFC: Dr. Connie Carrington, MSFC: For support and guidance on this project and Joe Howell, MSFC: For concept discussions and encouragement

REFERENCES

[1] Ananthasuresh, G. K., and Kota, S., 1995, "Designing Compliant Mechanisms," Mechanical Engineering, **117**(11), pp. 93-96.

[2] Fu, Y. and Du, H., 2003, "Fabrication of Micromachined TiNi based Microgripper with Compliant Structure," *Proceedings of SPIE*, Vol. **5116**, pp. 38-47.

[3] Frecker, M.I., Kota, S. and Kikuchi, N., 1998, "Optimal Design of Compliant Mechanisms for Smart Structures Applications," *SPIE Conference on Mathematics and Control in Smart Structures*, San Diego, California, Vol. **3323**, pp. 234-242.

[4] Dziedzic, R. P. and Frecker M. I. 2002 "Design of Multifunctional Compliant Mechanisms for Minimally Invasive Surgery," *Proceedings Of the 2002 Design Engineering Technical conferences DETC2001/Mech-34212.*

[5] Kota, S., 1999, "Design of Compliant Mechanisms. Applications to MEMS," *SPIE Conference on Smart Electronics and MEMS, Newport Beach, California*, Vol. **3673**, pp. 45-54.

[6] Dong X. and Ananthsuresh, G.K., 2003, "Freeform Skeletal Shape Optimization of Compliant Mechanisms," Journal of Mechanical Design, **125**, pp. 253-261.

[7] Tai, K., Cui, g. Y. and Ray, T., "Design Synthesis of Path Generating Compliant Mechanisms by Evolutionary Optimization of Topology and Shape," *Proceedings of the ASME 2000 Design Engineering Technical Conferences, DETC2000/DAC-14518.*

[8] Canfield, S. and Frecker, M., 1999 "Design of Piezoelectric Inchworm Actuator and Compliant End-Effector for Minimally Invasive Surgery," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. **3668**(2), pp. 835-843.

[9] Papalambros, P. Y. and Chirehdast, M., 2000, "An Integrated Environment for Structural Configuration Design," Journal of Engineering Design, Vol. 1, No. 1, 1990, pp. 73-96.

[10] Marson, A. L. and Dutta, D., 1996 "Construction of a Surface Model and Layered Manufacturing Data from 3D Homogenization Output," Journal of Mechanical Design, Vol. **118**, pp. 412-418.

[11] Sorin, D. and Schroder, P. 2000 Organizers of the "Subdivision for Modeling and Animation," SIGRAPH 2000 Course Notes.

[12] Catmull, E. and Clark, J., 1978 "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes," Computer Aided Design, **10**, 1978, pp.350-355.

[13] Doo, D. and Sabin, M., 1978 "Behavior of Recursive Division Surfaces near Extraordinary Points," Computer Aided Design. 1978, **10**(6), pp.356-360.

[14] Green, S., Turkiyyah, G. and Storti, D., 2002 "Proceedings of the seventh ACM symposium on Solid modeling and applications," *ACM Symposium on Solid Modeling and Application*, pp.265-272.

[15] Haftka, R.T., and Grandhi, R. V., 1986, "Structural Shape Optimization- A Survey," Computer Methods in Applied Mechanics and Engineering, **57**, pp. 91-106.

[16] Lau, G.K., Du, H. and Lim, M. K., 2001, "Use of Functional Specifications as Objective Functions in Topological Optimization of Compliant Mechanism," Computational Methods Applied Mechanical Eng., **190**, pp. 4421-4433.

[17] Chapman, C. D., Saitou, K. and Jakiela, M. J., 1994, "Genetic Algorithms as an Approach to Configuration and Topology Design," Journal of Mechanical Design, **116**, pp. 1005-1012.

[18] Jakiela, M. J., Chapman, C. D., Duda, J., Adewuya, A. and Saitou, K., 2000, "Continuum Structural Topology Design with Genetic Algorithms," Computational Methods Applied Mechanical Eng., **186**, pp. 339-356.

[19] Lu, K. and Kota, S., 2003, "Design of Compliant Mechanisms for Morphing Structural Shapes," Journal of Intelligent Material Systems and Structures, **14**, pp. 379-391.

[20] Parsons, R. and Canfield, S., 2002 "Developing Genetic Programming Techniques for the Design of Compliant Mechanisms," Journal of the International Society for Structural and Multidisciplinary Optimization, **24**(1), pp. 78-86.

[21] Saxena, A., 2002, "On Multiple-Material Optimal Compliant Topologies: Discrete Variable Parameterization Using Genetic Algorithm," *Proc. of The 2002 Design Engineering Technical Conferences, Montreal, Canada, DETC2002/MECH-34209.*

[22] Frecker, M.I., Ananthsuresh, G.K., Nishiwaki S., Kikuchi, N., and Kota, S., 1997, "Topological Synthesis of Compliant U Multi-Criteria Optimization," Journal of Mechanical Design, **119**, pp. 238-245.

[23] Roberts, F S., 1976 "*Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems,*" Prentice-Hall, Phoenix, Az.

[24] Deb, K. and Goel, T., 2001, "A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design," *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001)*, 7--9 March. (Zurich, Switzerland), (pp. 385--399).

[25] Chipperfield, A., Fleming, P., Pohleim, H., and Fonesca, C. 2002 "Genetic Algorithm Toolbox, for use with MATLAB," Users Guide Version 2.1.

[26] Dozier, G.; Bowen, J., and Homaifar, A., 1998, "Solving constraint satisfaction problems using hybrid evolutionary search," IEEE Transactions on Evolutionary Computation, Vol. **2**(1), pp. 23-33

[27] Yin, L. and Ananthsuresh, G. K., "Design of Distributed Compliant Mechanisms," Mechanics Based Design of Structures and Machines, Vol. **31**(2), pp151-179, 2003.

[28] Saxena, A. and Ananthasuresh, G. K., 2000, "On an Optimal Property of Compliant Topologies," *International Research Journal of Structural Optimization*. Vol. **19**, Issue 1, pp. 36-49.

[29] Hetrick, J. A. and Kota, S., 1998, "Size and Shape Optimization of Compliant Mechanisms: An Efficiency Formulation," *Proceedings of the ASME 1998 Design Engineering Technical Conferences, DETC98/MECH-5943.*

APPENDIX A

A brief summary of the subdivision scheme derived by the authors and used in this paper is given here. See [11] for a more complete description of various subdivision schemes. The subdivision matrix, **S**, is defined first for interior or boundary nodes, and then for nodes characterized as a singularity point. The subdivision scheme used here requires several definitions: *Vertex point (V)*: Point about which the local subdivision scheme is centered, every point in the original shape will be a vertex point at some time. *Edge Point (E)*: These points are directly connected to the current vertex point by a single line, there are between 2 and 4 edge points at a time and a *Face Point (F)*: A point, which is located diagonally from the vertex point, it requires a path of exactly 2 line segments to get to it and must cross diagonally the face, and *Valence*: Represents the number of edge points connecting a given node. These nodes are shown in Fig. 1A.

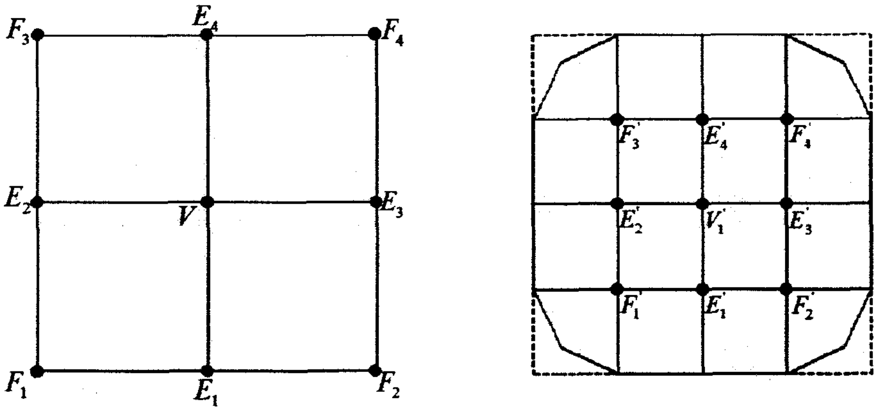

Figure 1A: a) midpoint vertices, b) new vertexes calculated from midpoints

The new control points shown in Fig. 1A are derived from the original control points according to the equation,

$$\begin{bmatrix} V^1 \\ E^1{}_1 \\ E^1{}_2 \\ E^1{}_3 \\ E^1{}_4 \\ F^1{}_1 \\ F^1{}_2 \\ F^1{}_3 \\ F^1{}_4 \end{bmatrix} = S \cdot \left\{ \begin{matrix} V \\ E_1 \\ E_2 \\ E_3 \\ E_4 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \end{matrix} \right\} \qquad \text{Eq. 1A}$$

where a trailing superscript indicates the new set of control points (first level of subdivision). The Subdivision matrix with the corresponding weights parameters and then values is given in Eqn. 2A.

$$S = \begin{bmatrix} \alpha_4 & \alpha_1 & \alpha_1 & \alpha_1 & \alpha_1 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 \\ \alpha_2 & \alpha_2 & \alpha_3 & \alpha_3 & 0 & \alpha_3 & \alpha_3 & 0 & 0 \\ \alpha_2 & \alpha_3 & \alpha_2 & 0 & \alpha_3 & \alpha_3 & 0 & \alpha_3 & 0 \\ \alpha_2 & \alpha_3 & 0 & \alpha_2 & \alpha_3 & 0 & \alpha_3 & 0 & \alpha_3 \\ \alpha_2 & 0 & \alpha_3 & \alpha_3 & \alpha_2 & 0 & 0 & \alpha_3 & \alpha_3 \\ \alpha_4 & \alpha_4 & \alpha_4 & 0 & 0 & \alpha_4 & 0 & 0 & 0 \\ \alpha_4 & \alpha_4 & 0 & \alpha_4 & 0 & 0 & \alpha_4 & 0 & 0 \\ \alpha_4 & 0 & \alpha_4 & 0 & \alpha_4 & 0 & 0 & \alpha_4 & 0 \\ \alpha_4 & 0 & 0 & \alpha_4 & \alpha_4 & 0 & 0 & 0 & \alpha_4 \end{bmatrix} \qquad \text{Eq. 2A}$$

with the weights for this subdivision scheme given in Eqn. 3A.

$$\begin{aligned} \alpha_1 &= 1/8 \\ \alpha_2 &= 3/8 \\ \alpha_3 &= 1/16 \\ \alpha_4 &= 1/4 \\ \alpha_5 &= 1/2 \\ \alpha_6 &= 5/16 \end{aligned} \qquad \text{Eq. 3A}$$

In a similar manner, a subdivision matrix S is defined to accommodate boundary nodes (using only three edge nodes and two face nodes)

The second subdivision matrix needed is for the special case of a singularity point (a vertex with two or less face points or two blocks connected by a single point) as shown in Fig. 2A. All four face and edge points are used in defining four new face and edge points (as in Eq. 1A). The S-matrix for the singularity condition is given in Eq. 4A

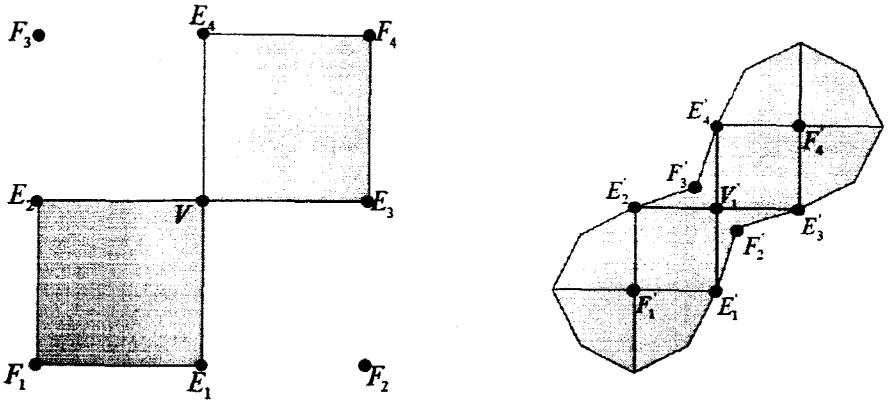

Figure 2A: a) midpoint vertices, b) new vertexes calculated from midpoints

$$S=\begin{bmatrix} \alpha_6 & \alpha_1 & \alpha_1 & \alpha_1 & \alpha_1 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 \\ \alpha_5 & \alpha_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_5 & 0 & \alpha_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_5 & 0 & 0 & \alpha_5 & 0 & 0 & 0 & 0 & 0 \\ \alpha_5 & 0 & 0 & 0 & \alpha_5 & 0 & 0 & 0 & 0 \\ \alpha_4 & \alpha_4 & \alpha_4 & 0 & 0 & \alpha_4 & 0 & 0 & 0 \\ \alpha_5 & \alpha_4 & 0 & \alpha_4 & 0 & 0 & 0 & 0 & 0 \\ \alpha_5 & 0 & \alpha_4 & 0 & \alpha_4 & 0 & 0 & 0 & 0 \\ \alpha_4 & 0 & 0 & \alpha_4 & \alpha_4 & 0 & 0 & 0 & \alpha_4 \end{bmatrix} \qquad \text{Eq. 4A}$$

With the weightings defined in Eq. 3A.