# 4 CATALOGUING AND SELECTION OF DISPLACEMENT-AMPLIFYING COMPLIANT MECHANISMS

**Neeraj Jhawar*** **and G. K. Ananthasuresh**†

**Punjab Engineering College, Chandigarh. E-mail: neerajjhawar.research@gmail.com*
*†Mechanical Engineering, Indian Institute of Science, Bangalore. Tel: +91 80 2293 2334.*
*E-mail: suresh@mecheng.iisc.ernet.in*

Selection from among competing alternatives is pursued in this work for a rapidly growing class of compliant mechanisms. By using the framework of storage and retrieval of a database, we present a method to select a displacement-amplifying compliant mechanism (DaCM) for a given application. The motivation for this work is to catalogue known compliant mechanisms and thereby select the most suitable mechanism for given quantitative specifications of the user. The quantitative data for a DaCM includes force, displacement, and stiffness specifications at the input and the output. The DaCMs in our catalogue consist of slender beams configured in particular topologies. These are analyzed using finite element analysis (FEA) and the mechanism's springs-mass lever model (SML) parameters are extracted and stored parametrically as functions of the size of the device. Thus, by using the specifications of the user, pre-computed parametric SML model parameters are utilized to select the most suitable DaCMs of appropriate size.

*Keywords:* Displacement-Amplifying Compliant Mechanisms, Catalogue, Selection.

## 1. INTRODUCTION

Compliant mechanisms are joint-free, elastically deforming structures that serve the purpose of force and motion transmission as well as transformation. In this work, we focus on a particular type of compliant mechanisms that can provide an amplified output displacement for a given input displacement. We call them Displacement-amplifying Compliant Mechanisms (DaCMs).[1] Systematic design of compliant mechanisms has been an active area of research in the last two decades. Just as the motion of a complicated rigid-body linkage is not intuitive, the motion of the compliant mechanism too is not obvious. Predicting the direction in which an elastic structure would move at a point due to a force applied at some other point requires a good understanding of elastic mechanics. It is not easy without an intuition for elastic deformation. This makes the design of compliant mechanisms hard. Hence, the development of design methods has become the focus of much of compliant mechanism research.

Two classes of design methodologies have emerged for compliant mechanism research. The first uses kinematics design principles and a pseudo rigid-body model to account for the elastic deformation of flexural joints and/or beams[2] as shown in Figure 1a. The other class of methods use elastic displacement analysis coupled with optimization techniques[3] as shown in Figure 1b. Both methods have been demonstrated to be useful in solving practical design problems. In the latter class of methods, two open-access software programs are also available (TopOpt[4] and YinSyn[5]). These programs generate optimal topologies of compliant mechanisms from nominal high-level specifications from the users. The phrase *optimal topology* here implies the geometric form of the elastically deformable structure with optimum number of holes placed at the most appropriate locations along with the optimal shapes and sizes for all its features. Since compliant mechanisms are single-piece deformable structures, their conceptual design is equivalent to topology determination. Their conceptual design can thus be automated by way of topology optimization. However, this has some limitations at present: post-processing is needed to avoid highly stressed flexural joints;[3] stress and buckling constraints are not yet implemented to

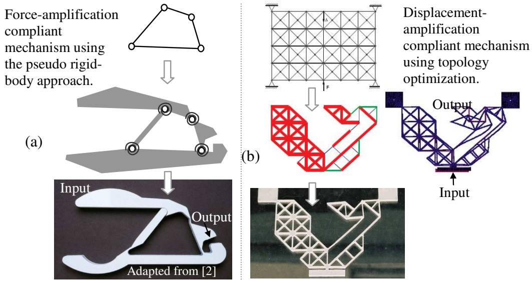


**Figure 1.** Two existing approaches to compliant mechanism design (a) Pseudo rigid-body approach where one begins with a rigid-body linkage topology, determines torsional spring constants, and then obtains the continuum design (b) Topology optimization-based approach wherein the optimization algorithm automatically generates the topology by determining the widths or thicknesses of the segments in a super-structure.

pragmatically useful extent;[7] manufacturability as in minimum feature size,[8] the level of complexity of shapes, batch-producibility, etc.; ergonomics; and aesthetics.

In this work, we take a different approach based on selection among known designs most of which were produced by topology optimization but some were conceived intuitively. We do this for two reasons: (i) all practical requirements cannot be incorporated into topology optimization at this time of its development, and (ii) the number of compliant mechanisms designs is steadily growing making selection an attractive, computationally efficient alternative design paradigm. Selection approach is computationally efficient because it only requires evaluation of designs in the catalog to see if any of them suit the new application. Furthermore, the designs in the catalog already satisfy many of the practical requirements. While our approach can be extended to other types of compliant mechanisms (e.g., force-amplifying compliant mechanisms, bistable compliant mechanisms, compliant grippers, compliant suspensions for single and two axis platforms, flexures, etc.), we consider only DaCMs in this paper.

### 1.1. Problem Statement and Related Issues

The tasks addressed in this work can be stated as follows: "cataloguing of DaCMs and parameterizing them with sufficient generality so as to select one or more suitable DaCM designs for a new application."

It is not enough to store DaCM designs as pictures or sketches in a computerized catalog because their geometric form and material properties are crucial for their functionality. Hence, we store meshed finite element analysis (FEA) models in the catalog. But this does not restrict the size of the mechanism or the aspect ratio of the rectangle that bounds the space occupied by the mechanism. Since the DaCMs we have considered are made of slender beam segments, there is adequate generality in morphing the DaCMs of the catalog to suit the user specifications. Furthermore, the DaCMs are parameterized as a function of the size and the aspect ratio. The parameters come from a lumped model of a DaCM called the spring-mass-lever model (SML). This model is described in Section 2. The selection algorithm uses the parameterized SML models. The algorithm, the modules of the prototype software implemented in Matlab, and the graphical user interface (GUI) are presented in Sections 3 and 4.

## 2. SPRING-MASS-LEVER MODEL AND CATALOGUING DaCMs

The kinematic and elastic behaviors of a DaCM can be expressed using an SML model so that its amplifying feature as well as the stiffness and inertia properties can be accounted for Ref. 1. The concept of an SML model is similar to that of representing an elastic structure as a single lumped spring and a mass. A single spring is enough if there is only one port where a force is applied. But a DaCM has two ports: input and output. Since it amplifies displacement, it is like a lever but a lever with finite stiffness at the input and output sides. Unlike a normal rigid-body lever, the de-amplification

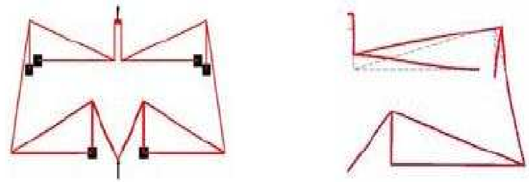

**Figure 2.** (a) A Displacement-amplifying Compliant Mechanism (b) the right symmetric half along with the deformed profile shown in dashed lines.

from the output to the input side is not simply the reciprocal of the amplification factor from input to the output. In order to appreciate this and the other features of compliant mechanisms, consider the DaCM shown in Figure 2a.

The DaCM shown in Figure 2a is a 2D mechanism consisting of slender beam segments shown as lines. The black squares indicate the anchors; i.e., those points are fixed to a frame. The input is indicated with $i$ and output with $o$. Arrows at the input and output indicate the respective directions of motions. The symmetric right half is shown in Figure 1b along with the deformed profile in dashed lines. It can be noticed that the input has hardly moved while the output has moved substantially. If we effect an input displacement $u_i$ by applying a force there, we get an output displacement $u_o$. The ratio of the two, $n = u_o/u_i$ is called the inherent amplification of the DaCM. The DaCM would deform in a certain way under this force and it is shown in Figure 2b. Now, imagine applying a force at the output and effecting displacements $v_o$ and $v_i$ at the output and input points, respectively. Now, this mechanism would deform in a different way. Consequently, $v_i/v_0 \neq (1/n)$. In other words, the deformation pattern of the DaCM, or any elastic structure for that matter, depends on the point of application of the force and the direction of the force.

The non-interchangeability of input and output behavior of a DaCM can also be understood with a simpler example of a cantilever beam with a tip load. The beam bends under a sufficiently large transverse load causing both axial and transverse displacements of the tip. But when only an axial–say tensile–load is applied, the mode of deformation is simple axial stretching without any transverse displacement of the tip. The SML model captures this subtle feature of DaCMs in the way they deform under loads applied at the input and output.

Figure 3 shows the SML model of a DaCM within a dashed rectangle. It has five parameters: $n =$ inherent amplification from the input side to the output side; $k_{eci} =$ input side stiffness; $k_{co} =$ output side stiffness; $m_{ci} =$ input side inertia; and $m_{co} =$ output side inertia. Two more springs are also shown in Figure 3 to illustrate that this model allows the modeling of any other attachments to a DaCM. Thus, DaCM can be treated like a black-box using its SML. Here, we show a spring at the input side to indicate actuation stiffness or the stiffness of a signal such as the one encountered in a compliant accelerometer or another sensor. We also show an external spring at the output to indicate a load. Forces can now be applied directly on the masses, i.e., the input and output ports. Because of the way the masses are arranged at the input and output sides, the displacement patterns of the SML are different for forces applied at the input and output.

For static behaviors, we only need $n, k_{ci}$, and $k_{co}$. If for two DaCMs, these three parameters are identical, we can conclude that both are the same for the purposes of basic functionality. We use this feature to compare different DaCMs in the catalog. Quantitative differences in the performance can also be determined using this model. This forms the basis for our selection algorithm. We explain the procedure for obtaining the three parameters for a given DaCM's meshed FEA model.

## 2.1. Determining the SML parameters

The SML parameters, as explained above, are lumped parameters that describe the abstracted input-output terminal behavior. In order to compute them, however, we need to do the finite element analysis of the compliant mechanism. As shown in Figure 4, once the input and output as well as the anchor

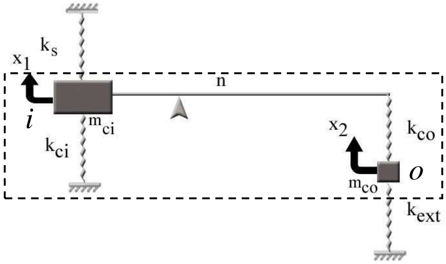


**Figure 3.** A spring-mass-lever model of a DaCM is shown within the dashed rectangle. Two additional springs ($k_s$ = input actuation/signal spring, and $k_{ext}$ = external output spring) are also shown to illustrate that SML enables a block-box approach to a DaCM.

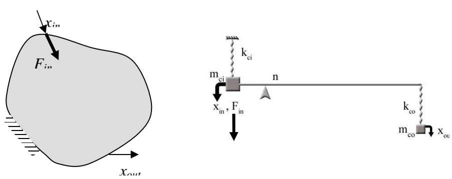

**Figure 4.** A compliant mechanism and its SML model when only the input force is applied. This is used to calculate (with the help of the finite element analysis of the meshed model of the compliant mechanism) the input-side stiffness, $k_{ci}$, and inherent amplification, $n$.

portions are identified, we apply an input force $F_{in}$ only and measure the displacement $x_{in}$ at the input. With these, as in an ordinary lumped spring modeling of an elastic structure, we compute the input-side stiffness, $k_{ci}$.

$$k_{ci} = \frac{F_{in}}{x_{in}} \tag{1}$$

By also measuring the output displacement, $x_{out}$, we compute the inherent amplification, $n$.

$$n = \frac{x_{out}}{x_{in}} \tag{2}$$

Next, we consider a different loading situation shown in Figure 5, to get the lumped behavior of the compliant mechanism when the force is applied only at the output in the direction opposite to the intended output displacement, i.e., the direction of the output load. By considering the SML model of this situation shown in Figure 5, we write the potential energy, $PE$, as follows. Note that the potential energy is the sum of the strain energy and the negative of the work done by the external forces.

$$PE = -F_{out}y_{out} + \frac{1}{2}k_{co}(y_{out} - ny_{in})^2 + \frac{1}{2}k_{ci}y_{in}^2 \tag{3}$$

By differentiating $PE$ with respect to $y_{in}$ and $y_{out}$, and equating them to zero for static equilibrium, we get two equations that can be solved to get $y_{in}$ and $y_{out}$. The resulting expression for $y_{out}$ is used to solve for $k_{co}$ as follows.

$$k_{co} = \frac{F_{out}k_{ci}}{k_{ci}y_{out} - F_{out}n^2} \tag{4}$$

When dynamic behavior is considered, which is not done in this paper, the lumped inertia parameters, $m_{ci}$ and $m_{co}$, can be computed in a similar way by computing the first two natural frequencies with the help of the modal analysis of the SML model. The SML parameters completely determine the essential kineto-elastic behavior of the compliant mechanism.

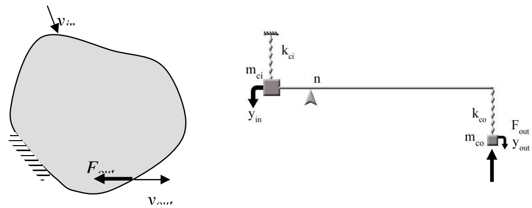


**Figure 5.** A compliant mechanism and its SML model when only the output load is applied. This is used to calculate (with the help of the finite element analysis of the meshed model of the compliant mechanism) the output-side stiffness, $k_{co}$.

### 2.2. Cataloguing DaCMs

The proposed DaCM catalogue has three types of data: (i) pictures and animations, (ii) meshed finite element model, and (iii) SML parameters.

The purpose of the first one is to show the mechanism topology with the complete details of the geometry and enable pictorial depiction of how it works. This includes pictures and animation of the deformation pattern. It should be noted that the animated deformation of a compliant mechanism says a lot to the discerning user.

The second one–the meshed model–is really what it says it is. It consists of the essential data pertaining to the finite element model of the DaCM. It is similar to the input file of typical commercial finite element analysis software. It consists of the nodal coordinates, element connectivity (i.e., which nodes make up an element), cross-section properties of the elements, material property data, and boundary conditions. Boundary conditions define which nodes are fixed and where the input force is applied, and where the output load and/or spring are connected.

The third is the important and non-trivial part of the database. Here, we store the SML parameters, $k_{ci}$, $k_{co}$, and $n$. These parameters are given as functions of the size of the DaCM. This is explained next.

The default size of the DaCM is not necessarily the size that a designer desires for a new application at hand. In view of this, we compute the three SML parameters a priori for varying sizes. By sizes, here, we mean the extents in $x$ and $y$ directions of the rectangle that bounds the DaCM. For this, we stretch the finite element mesh in both directions by different magnitudes as necessitated by the user-specified size. This is applicable only to the nodal coordinates while we keep the element cross-section properties the same. For each pair of stretches in $x$ and $y$ directions, we perform a finite element analysis and extract the SML parameters as per the procedure explained in Section 2.1. This is done with sufficiently fine resolution of the sizes so that the SML parameters can be reasonably interpolated for any specified size. Thus, each DaCM is catalogued with all its size variations. The cross-section dimensions of the DaCM are not varied here because that is better done by size-optimization[9] rather than uniformly changing all the dimensions by the same factor.

## 3. SELECTION ALGORITHM

### 3.1. Example Problem Statement

DaCMs are useful for sensor and actuator applications. Here, we consider an actuator example to explain the selection algorithm. The parameters specified for each problem are generic and so the algorithm can be used for other applications with only small changes in the terminology and slight modifications of the procedure. The problem of interest, a valve example, is shown in Figure 6. As shown in the figure, the displacement of an actuator needs to be amplified at the output against an output load. A DaCM is necessary if the maximum stroke of the actuator is far too small as compared with the required displacement at the output. In order to achieve this, the actuator should have a fairly large force as compared with the output load. Such a situation, for instance, arises with

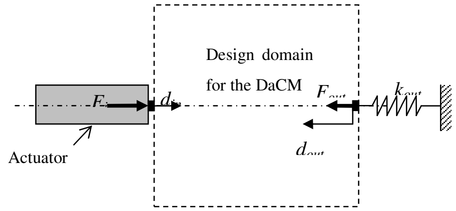


**Figure 6.** Specifications of an example problem of an actuator for which a Displacement-amplifying Compliant Mechanism (DaCM) is necessary.

piezo-actuators. Per Figure 6, the following parameters are defined for this application:

Input force $= F_{in}$
Maximum input displacement $= d_{in}$
Stiffness of the actuator $= k_a$
Output load $= F_{out}$
Required output displacement $= d_{out}$
Stiffness at the output $= k_{ext}$

Maximum input force is transferred to the DaCM when the input displacement is zero. Due to the stiffness of the actuator, the effective force transferred to the mechanism gradually decreases and becomes zero at the maximum input displacement. Hence, only two out of the three parameters, $F_{in}$, $d_{in}$, and $K_a$, are specified. That is, $k_a$ can be inferred when $F_{in}$ and $d_{in}$ are specified. Both output force and the output spring may not usually exist; only one of them is usually there. For the sake of generality, both are included here. Except the output displacement, all other values are specified by the chosen actuator and the output load.

## 3.2. Re-Sizing and Selection Procedure

When the user enters the values of the parameters shown in Figure 6, an algorithm is run to process the data in the catalogue of DaCMs. For each DaCM, using the SML parameters, the input and output displacements are calculated using the following formulae derived by applying the static equilibrium equations to the SML model attached to the actuator and external springs shown in Figure 3.

$$u_{in} = \frac{F_{out} n k_{co} + F_{in}(k_{co} + k_{ext})}{k_{ext}(k_{ci} + k_s + n^2 k_{co}) + k_{co}(k_{ci} + k_a)} \tag{5}$$

$$u_{out} = \frac{F_{out}(k_{ci} + k_s + n^2 k_{co}) + F_{in} n k_{co}}{k_{ext}(k_{ci} + k_s + n^2 k_{co}) + k_{co}(k_{ci} + k_a)} \tag{6}$$

The computed displacements are checked against the maximum possible input displacement and the required output displacement. That is, the DaCMs that satisfy the following conditions are marked as suitable ones.

$$u_{in} - d_{in} \leq 0 \tag{7}$$

$$d_{out} - u_{out} \leq 0 \tag{8}$$

This simple check is not likely to yield a solution for any specifications given by the user. The reason for the likely failure is not the number of designs in the catalog (which is eight at present and can be extended to 16 immediately and to a much larger number later) but the incompatibility in the size of the mechanism as a whole and the dimensions of the beam segments. For this reason, as noted in Section 2.2, each DaCM is stored with a long list of SML parameters that vary with the size of the mechanism. In order to use this feature, the user is prompted to enter the maximum and minimum size of the mechanism in the $x$ and $y$ directions.

Now, the database is searched to see for which size-variations of the DaCM, Equations (7) and (8) are satisfied, and those entries are retrieved and ranked by $d_{out}$, the one with the largest value of $d_{out}$ being the most preferred. The satisfactory ones ranked in this manner are presented to the user for final selection.

## 4. CHOOSING THE EXTERNAL SPRING CONSTANT

As an alternative to the above algorithm, we also consider choosing a suitable external spring constant value. The purpose of this alternative algorithm is to increase the scope of finding more designs that are suitable. This alternative is applicable irrespective of whether the user specifies $k_{ext}$ or not. As can be understood from Figure 3 and Equation (5), the output spring (i.e., $k_{ext}$) helps control $u_{in}$. Since there is an upper limit on $u_{in}$, we can find a suitable $k_{ext}$ using the following.

$$k_{ext} = \frac{k_{co}(F_{in} + F_{out}n - d_{in}k_s - d_{in}k_{ci})}{d_{in}\left(k_{ci} + k_s + n^2 k_{co}\right) - F_{in}} \tag{9}$$

The corresponding output displacement is then computed as follows.

$$u_{out} = \frac{u_{in}\left(k_{ci} + k_s + n^2 k_{co}\right) - F_{in}}{nk_{co}} \tag{10}$$

These values are computed in real time and additional designs along with the values of $k_{ext}$ are displayed to the user. The satisfactory ones are ranked as above to present to the user. Low $k_{ext}$ takes precedence in ranking.

## 5. SOFTWARE WITH A GRAPHICAL USER INTERFACE

The selection algorithms are implemented in Matlab environment with a GUI. Since the computation involved in the algorithms is not significant, the speed is not limited at present. Implementing in Matlab has the advantage that developing the code, plotting figures and interacting with the user are easy.

The GUI contains a line-sketch of the DaCM along with its deformed configuration. Also included in the GUI are two surface plots that show the values of $u_{in}$ and $u_{out}$ in the same units as those entered by the user against the sizes in $x$ and $y$ along their respective axes. The numerical values of $u_{in}$ and $u_{out}$ are indicated in the text boxes. The user can enter the desired overall sizes of the mechanism in the text boxes provided for them. Even though the algorithm already selects the best possible size-variant of the mechanism, the software gives the option to the user to enter different values for the sizes in both the directions. As the user enters different values, two guiding rectangles in the surface plot shift to indicate the changed choice. This enables the user explore the effect of changing the size of the mechanism. Two buttons, NEXT and PREVIOUS, are provided to enable the user to consider each mechanism, if he/she so desires. There is also a button "k(ext) Test". When the user clicks on this button, the algorithm explained in Section 3.3 is run and two plots that indicate the values of $k_{ext}$ and $U_{out}$ are generated. As before, the user can enter different sizes in the $x$ and $y$ directions and explore different DaCM mechanism topologies for each of which the most suitable size-variant is displayed at first. The user can still see what happens if the size is relaxed or tightened. The entire sets of attributes are displayed for the final mechanism chosen as shown in Figure 7.

## 6. DISCUSSION

A comment on the size of the database is appropriate here. As stated, only eight DaCM designs are coded in the software now. All these are very different in terms of their SML parameters. Some more are already known in the literature and many more can be created using topology optimization. This will be done as this ongoing work progresses. But it should be noted that, because we allow size variations, there is enough scope to explore a number of design alternatives and identify all those that are applicable.

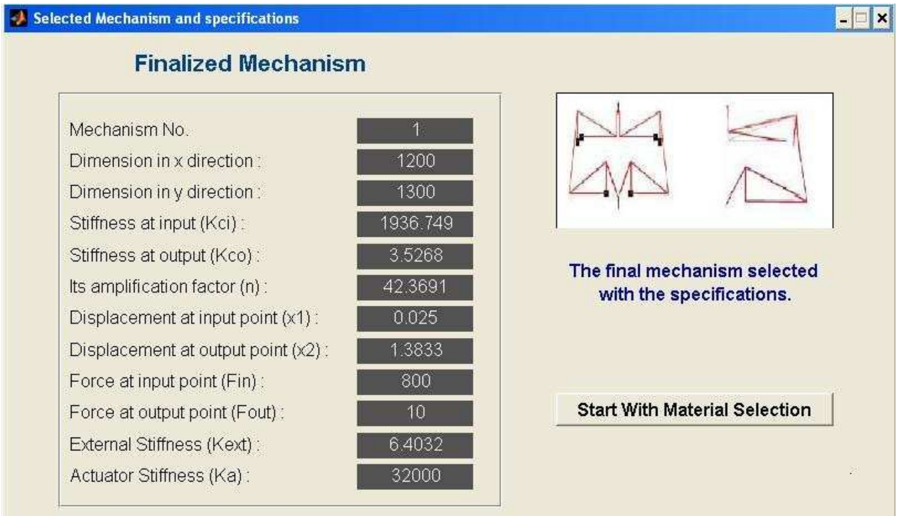


**Figure 7.** Display of the attributes of the final mechanism preferred by the user from among the alternatives suggested. The software gives the user the option of choosing or discarding the alternative designs that meet the specified requirements.

The principal benefit of the selection approach described here is to identify suitable DaCMs with the appropriate dimensions for given specifications. The topology optimization procedures that are widely researched in the literature also enable the same. But the difference here is that only manufacturable and tested designs are present in the catalog. Additionally, the selection is done with very little computation because a parameterized database exists in the software. More features will be included in this software as part of the extensions of the ongoing work. For example, the cross-section dimensions of the beam elements are currently kept constant as the size of mechanism is changed. But it does not have to be this way. The scope of the design can be expanded by parameterizing the SML model in terms of the cross-section dimensions.

In Figure 7, there is a button that shows “Start with Material Selection”. This points to another extension (see Ref. 10) where the maximum stress allowed by a material can be taken into account so that strength considerations too can be incorporated into the software. It should be noted that including stress constraints in topology optimization is still an unsolved problem. Similarly, accounting for possible buckling is also an unsolved problem. Here, since the parameterized SML parameters of the DaCMs can be checked for buckling and other potential problems before putting into the database, the selection approach proposed in this paper looks attractive from a practical viewpoint. Finally, the natural frequency requirements can be met by also including the inertia parameters of the SML model.

## 7. CLOSURE

Even though the optimization methods can generate optimal conceptual topologies, the resulting designs are not often practical in view of manufacturability, meeting strength considerations, buckling, etc. In this paper, we propose an alternate approach to design based on selection from known topologies. By using a spring-mass-lever model and parameterizing it in terms of the mechanism’s overall size, we presented a selection algorithm and prototype software that implemented the algorithm. The software searches the catalog based on the user specifications and identifies the ones that satisfy the requirements by varying the size as necessary. This pragmatic approach yields practical designs that can be readily manufactured. Future work aims to extend this framework to obtain designs that are free from the problems such as buckling and excessive stress while also enabling material selection and meeting natural frequency requirements.

## ACKNOWLEDGMENTS

The authors thank Professor Amaresh Chakrabarti (Centre for Product Design and Manufacturing) for helpful discussions. This work was financially supported by the Summer Undergraduate Mentorship in Mechanical Engineering Research (SUMMER) programme of the Department of Mechanical Engineering at the Indian Institute of Science, Bangalore.


## REFERENCES


[1] Krishnan, G. and Ananthasuresh, G. K. (2008). A Systematic Method for the Objective Evaluation and Selection of Compliant Displacement Amplifying Mechanisms for Sensor Applications, *Journal of Mechanical Design*, (In press).

[2] Howell, L. L. (2001). Compliant Mechanisms, John Wiley & Sons, New York.

[3] Sangamesh R. Deepak, M. Dinesh, Deepak Sahu and G. K. Ananthasuresh (2008). A Comparative Study of the Formulations and Benchmark Problems for the Topology Optimization of Compliant Mechanisms, *ASME Journal of Mechanisms and Robotics*, 1(1), 20–27.

[4] www.topopt.dtu.dk

[5] www.mecheng.iisc.ernet.in/∼suresh/YinSyn

[6] Saxena, A. and Ananthasuresh, G. K. (2000). On an optimal property of compliant topologies, *Structural and Multidisciplinary Optimization*, 19, 36–49.

[7] Saxena, A. and Ananthasuresh, G. K. (2001). Topology Optimization of Compliant Mechanisms with Strength Considerations, *Mechanics of Structures and Machines*, 29, 199–222.

[8] Guest, J. K. (2006). A Penalty Function for Enforcing Maximum Length Scale Criterion in Topology Optimization, *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, 6–8 September.

[9] Hetrick, J. A. and Kota, S. (1999). An energy formulation for parametric size and shape optimization of compliant mechanisms. *ASME Journal of Mechanical Design*, 121(2), 229–234.

[10] Rakshit, S. and Ananthasuresh, G. K. (2008). Simultaneous material selection and geometry design of statically determinate trusses using continuous optimization, *Structural and Multidisciplinary Optimization*, 35, 55–68.