An Online Interactive Computer Program for Pragmatic Design of Compliant Mechanisms

Ranu Kundu, C. V. Aravind, Sudarshan Hegde, and G. K. Ananthasuresh

Abstract

Kinematic synthesis techniques that include springs to model elastic deformation, and solid mechanics and optimization-based techniques developed so far for compliant mechanism design are general enough for initial or conceptual design but are not, at this point of time, capable of easily handling practical requirements. Furthermore, they cannot judge a priori whether a set of user-specifications has a solution or not. To address these issues, we have recently developed a lumped input-output model of compliant mechanisms and a comprehensive method of design based on selection and re-design of entries stored in a database of compliant mechanisms. The method is capable of informing the user if the specifications given to it are feasible or not. It also enables the user, when it is possible, to select a topology of choice and re-design it interactively with due attention to maximum stress, manufacturing considerations, and material selection. To make this method accessible to designers at large, this paper presents an online implementation of the method with graphical user interface (GUI) (http://www.mecheng.iisc.enet.in/~m2d2/CMdesign). The necessary theoretical background is included and a case-study is presented.

Keywords: Compliant mechanisms, spring-lever model, design by selection, interactive design

1 Introduction

Compliant mechanisms that transmit or transform motion, force, and energy using elastic deformation of a single joint-less body are used in many applications today from large-scale morphing wings of aircraft and wind-turbine blades [1] to miniature devices in micro and nano-scale systems [2]. Research in this field is advanced by both ingenuity of the researchers and inventors who conceive new device concepts and the development of systematic design techniques. Extensive research in the last 25 years has led to two classes of design techniques for compliant mechanisms: (i) kinematic synthesis techniques enhanced to handle elastic deformations using springs [3] and (ii) structural optimization-based techniques [4, 5] adapted to compliant mechanisms. There are also many other useful methods including the emerging class of techniques based on building-blocks [6]. As in rigid-body linkage

_____

Ranu Kundu
Department of Mechanical Engineering, Indian Institute of Science Indian Institute of Science, Bangalore-560012, India, E-mail: ranu@mecheng.iisc.ernet.in.

C. V. Aravind
Infosys Inc., E-mail: cvaravinda@gmail.com.

Sudarshan Hegde
Department of Mechanical Engineering, Indian Institute of Science Indian Institute of Science, Bangalore-560012, India,, E-mail: hegde@mecheng.iisc.ernet.in.

G. K. Ananthasuresh
Department of Mechanical Engineering, Indian Institute of Science Indian Institute of Science, Bangalore-560012, India, E-mail: suresh@mecheng.iisc.ernet.in.

synthesis, most of these methods emphasize initial design that focuses only on basic functionality rather than practical aspects. That is, they aim to provide to the user a geometric form of an elastic continuum that can deform as desired by a designer. Typically, forces and/or displacements are specified at the input and output. This is similar to the standard tasks often mentioned in mechanism synthesis: function generation, path generation, and motion generation.

Figure (1) shows a typical geometric form generated by a topology optimization program YinSyn (http://www.mecheng.iisc.ernet.in/~m2d2/YinSyn) [7] for specified input force and output displacement specifications. Similar results are obtained using another program called TopOpt [8]. Such geometric forms generated by the compliant mechanism synthesis techniques often cannot directly account for strength considerations, selection of material and manufacturing techniques, and manufacturing and space limitations. More importantly, it is not possible in these methods to know if a compliant mechanism is even possible for a given set of user-specifications before attempting to solve the synthesis problem. Contrast this with rigid-body linkage synthesis [9] in which we know whether a linkage of chosen topology can pass through a certain number of user-specified points on an output path. In other words, there is no 'Burmester theory' [9] for compliant mechanisms.

While it may take a long time and a lot more research to develop an insightful Burmester-like theory for compliant mechanism design, we have recently attempted to answer the question of feasibility of user-specifications (see Table 1) for single-input-single-output compliant mechanisms [10, 11]. The ensuing method also enables the users to select a suitable mechanism from a database whenever it is possible and also re-design the most promising mechanism by accounting for practical considerations with the user in the loop. Therefore, it was natural to develop a computer program with a graphical user interface (GUI). We have chosen Java programming language as it is most effective for use on the Internet because of its web-readiness and platform-independence. The developed online interactive program is at http://www.mecheng.iisc.ernet.in/~m2d2/CMdesign for easy access for any interested users and also for increasing the database of compliant mechanisms and enhancing the program with feedback from users.
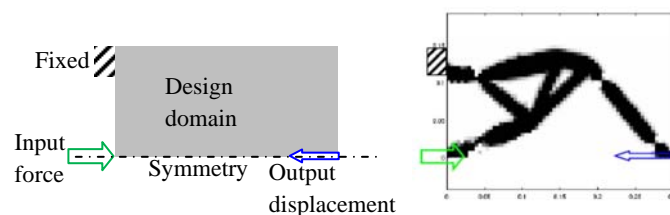

Figure 1: Input force and output displacement specifications and topology-optimized image of a compliant mechanism generated by YinSyn [7]

The method uses newly developed lumped models called *spring-leverage* (SL) model for static [10] and *spring-mass-leverage* (SML) model [11] for static and dynamic behaviors, respectively. In the current Java implementation only static behaviour is considered. Hence, we briefly review the theory for the SL model to explain how the method works. This is done in Section 2. Section 3 contains the implementation details of the Java program. Section 4 contains a detailed case-study that explains how to use the program interactively. Additional case-studies are available at the aforementioned website. Closing remarks are in Section 5.

Table 1: User-specifications for single-input-single-output compliant mechanisms

S. No.	User-specification	Unit	Min. and max. values
1	Input force	N	$F_{in}^{min}, F_{in}^{max}$
2	Input displacement	m	$x_{in}^{min}, x_{in}^{max}$
3	Actuator stiffness	N/m	k_a^{min}, k_a^{max}
4	Output force (load)	N	$F_{out}^{min}, F_{out}^{max}$
5	Output displacement	m	$x_{out}^{min}, x_{out}^{max}$
6	External stiffness	N/m	$k_{ext}^{min}, k_{ext}^{max}$

2 Selection and Re-design Methodology

Users who want to use single-input-single-output compliant mechanisms under static conditions usually would start with specifications such as the ones shown in Table 1. They reflect the fact that users usually prefer to specify a range rather than a particular value. The first question to answer is whether these specifications have a solution at all. A quick check on whether input energy is more than the output energy can answer this but only partially because there will be restrictions on space, strength, manufacturability, and other practical requirements. A method based on the SL model can provide useful information with respect to existing compliant mechanisms as explained next.

Figure (2) shows the SL model of a compliant mechanism along with its three 'properties', namely, input-side stiffness k_{ci}, output-side stiffness k_{co}, and intrinsic amplification of displacement from the input-side to the output-side n. These are physically meaningful and practically useful quantities that relate input-output forces and displacements as compared to the effective stiffness at input (k_{in}) and output (k_{out}), and the cross-stiffness ($k_{in/out}$).

$$\begin{Bmatrix} F_{in} \\ F_{out} \end{Bmatrix} = \begin{bmatrix} k_{in} & k_{in/out} \\ k_{in/out} & k_{out} \end{bmatrix} \begin{Bmatrix} x_{in} \\ x_{out} \end{Bmatrix} \quad (1a)$$

$$\begin{Bmatrix} F_{in} \\ F_{out} \end{Bmatrix} = \begin{bmatrix} k_{ci} + n^2 k_{co} & -n k_{co} \\ -n k_{co} & k_{co} \end{bmatrix} \begin{Bmatrix} x_{in} \\ x_{out} \end{Bmatrix} \quad (1b)$$

They are called 'properties' because they describe the intrinsic characteristics of compliant mechanisms independent of how they are used—just like material properties do for materials. This is an important analogy because the method of selection of compliant mechanisms is inspired by Ashby's method of selecting materials [12]. Shown in Fig. (3) is a k_{ci}-k_{co} map of compliant mechanisms in a database where each point represents the values of k_{ci} and k_{co} of a compliant

mechanism. If we now superimpose the feasible values of k_{ci} and k_{co} for user-specifications on it (say, points A to E), we can quickly ascertain if a particular compliant mechanism can satisfy the specifications. The feasible space is shown as the shaded region in Fig. (3). This is constructed by solving the equations of static equilibrium of the SL model and the inequalities imposed by user-specifications. The two degrees of freedom of the SL-model gives two equations that can be solved for k_{ci} and k_{co}.

$$k_{ci} = \frac{F_{in} - k_a u_{in} - n\left(k_{ext} u_{out} - F_{out}\right)}{u_{in}} \qquad (2a)$$

$$k_{co} = \frac{\left(k_{ext} u_{out} - F_{out}\right)}{n u_{in} - u_{out}} \qquad (2b)$$

In order to satisfy the minimum and maximum values of the six parameters specified by the user, Eqs. (2a) and (2b) need to be solved in conjunction with 12 inequalities.

$$F_{in}^{min} \le F_{in} \le F_{in}^{max} \qquad\qquad F_{out}^{min} \le F_{out} \le F_{out}^{max}$$

$$x_{in}^{min} \le x_{in} \le x_{in}^{max} \qquad\qquad x_{out}^{min} \le x_{out} \le x_{out}^{max} \qquad (3)$$

$$k_a^{min} \le k_a \le k_a^{max} \qquad\qquad k_{ext}^{min} \le k_{ext} \le k_{ext}^{max}$$
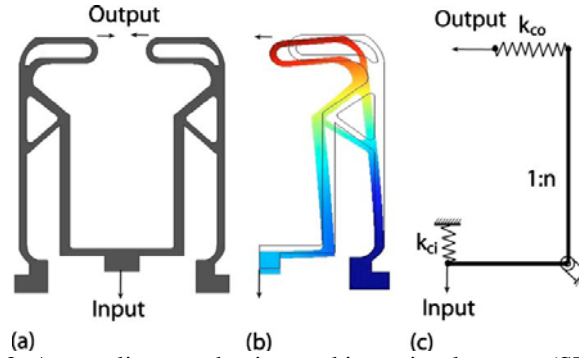

Figure 2: A compliant mechanism and its spring-leverage (SL) model
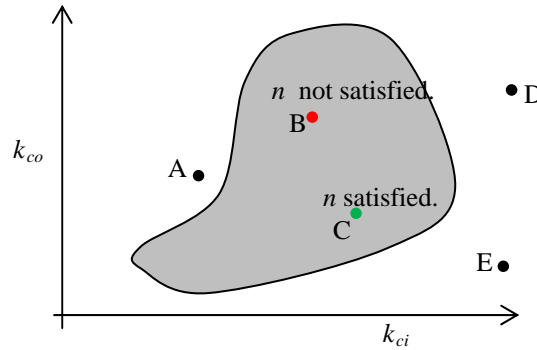

Figure 3: Schematic of the feasible k_{ci}-k_{co} map created using user-specifications with mechanisms of the database plotted as points in it.

The solution procedure is presented in [13]. It involves a few optimization problems to be solved and some algebraic manipulation. But it can be done in real time that is suitable for interactive use.

If a mechanism is inside the feasible region (points B and C in Fig. (3)), then it means that k_{ci} and k_{co} values of that mechanism are within the allowed values for k_{ci} and k_{co}. But then the value of n also needs to be checked. For given values of k_{ci} and k_{co}, permissible values of n can be obtained by solving Eqns. (2a)-2(b) and inequalities in Eqn. (3) [13]. By checking if n of the mechanism lies within that range, it can be concluded that the mechanism satisfies the user-specifications. Point C is shown to satisfy this condition in Fig. (3). If not or if the point corresponding to a mechanism lies outside the feasible region, the mechanism can be re-sized, its cross-section parameters modified, its thickness changed, or its material changed. All this can be done interactively in real time as explained in the next section. The related calculations involve finite element analysis of the mechanism based on the mechanism's information stored in the database. This can be done for both small displacement and large displacement (i.e., geometrically nonlinear) analysis. Details of this are not presented here due to paucity of space. Additionally, maximum stress of the chosen mechanism is also displayed to the user. The users can also specify the manufacturing constraints (e.g. minimum width) for a chosen process. The interactive programme will obey these constraints to ensure that the mechanism remains practical as the users re-design it. Since all mechanisms in the database are verified to be practical, when user-specifications are readily satisfied by a mechanism of the database, no further modification is required if the user selects it.

3   Online Implementation

Java programming language was chosen to implement the calculations discussed in the preceding section. Any computer with a web-browser, internet access, and the Java JRE (Java Runtime Environment) plug-in can access and run the program. The plug-in is a small, easy-to-install package freeware from Sun Microsystems, which allows a user to quickly download small 'applet' programs. These applets can run on any operating system without changing the interface. To create graphical interface in JAVA, predefined library classes provided by Sun Microsystems are used. The 'swing' library helped to create panels and buttons with the appropriate functionality. Other classes available in the standard Java libraries allowed functionality for animation, error-checking, threading, and file operations. The GUI of *CMdesign* in Fig. (4) shows the arrangement of the graphic features.

The algebraic manipulations involved in solving equations and inequalities in Eqns. (2) and (3), finite element analysis using 2D beam elements, and other calculations are all coded in Java.

Once the user submits the specifications in the User-input panel in Fig. (4), the code checks if the entered data is valid by checking if alphanumeric characters and are present and if minimum values do not exceed their corresponding maximum values. The same numbers can be entered for minimum and maximum if the user intends to enter a particular value. Zero can be entered for minimum and maximum if the user does not want to specify it. The stiffness map is then drawn in the graphics

window with domain and range axes as k_{ci} (N/m) and k_{co} (N/m) respectively. The mechanisms in the database are shown in a drop-down list, arranged in ascending order of their proximity from the feasible space. A mechanism will be plotted with a black circular annotation around it on the map when a user selects it from the list. By performing a double-click mouse operation on the selected mechanism, a user can view its current details, such as Minimum In-Plane width, Thickness, Size in X, Size in Y, Material's Young's Modulus, maximum stress, its image and animation as well as the source for the mechanism. There are currently 20 mechanisms in the database. Depending on the material, possible manufacturing processes are shown. The user can view the default values of the manufacturing process parameters and can alter them if required. If satisfactory mechanism(s) are readily found (i.e. k_{ci}, k_{co}, and n values of the mechanism(s) match the user-specifications as explained in Section 2), the user may just stop there and save the mechanism and receive the mechanism details by e-mail. If not, re-design can be attempted.
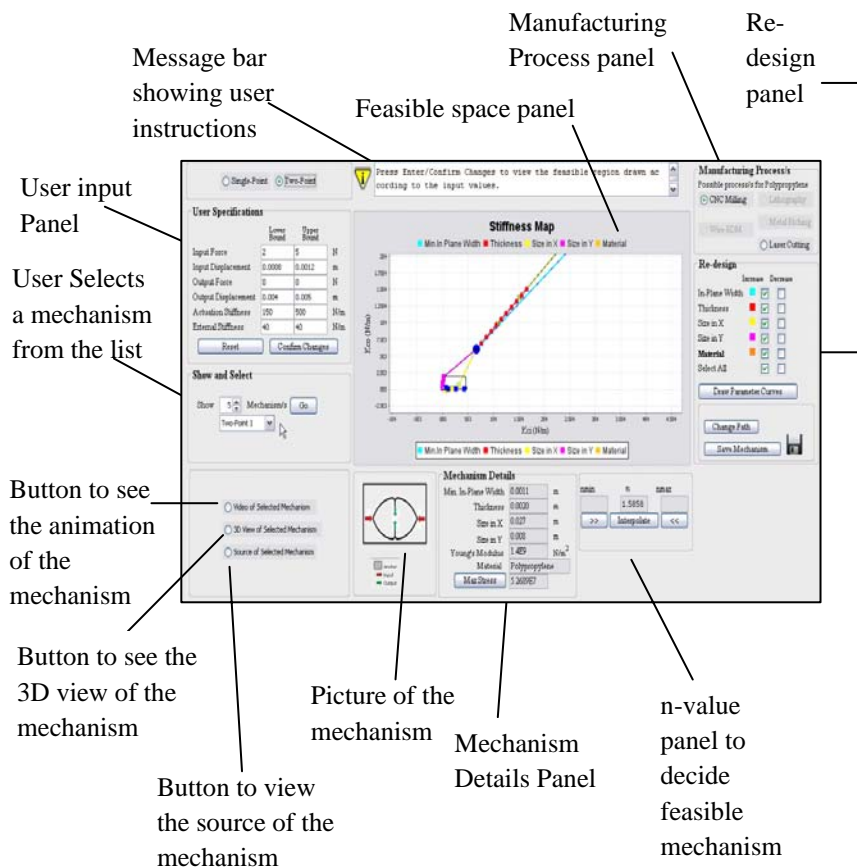

Figure 4: Graphical User Interface of CMdesign online program

The Re-design panel allows the user to modify the chosen mechanisms to meet user specifications through 'parameter curves', 'path-change' and interpolation steps. While the user can readily see if the parameter curves enter the feasible space or not, the checking of n value is enabled by displaying the valid values of n_{min} and n_{max} for any mouse-click on the parameter-curve. When n value matches, the point turns green indicating that a feasible mechanism is found. The user can then save the mechanism details. If the user is not successful in finding a suitable mechanism along any parameter curve from the mechanism's original point, a new point can be chosen to draw the new parameter curves. This is referred to as 'Change Path' operation. Maximum stress and changing size and shape of the mechanism is continuously displayed to the user. Parameter curves obey the manufacturing process parameters by terminating the path. Many other checks are ensured to make the program user-friendly. 'The data-flow of the entire program is shown in Fig. 5. A case-study is presented next to clarify how the program can be used to find a suitable and practically feasible compliant mechanism for a real problem at hand.

4 A Case-study

Let us assume that a user wants to design a Displacement-amplification Compliant Mechanism (DaCM) as per the specifications shown in Table 2. The feasible space for these is quickly created by the code as shown in Fig. (6a). Figure (6b) shows the mechanism selected by the user as it falls within the feasible region. It is Force_Sensor 1 with K_{ci} = 4.27E3 N/m and K_{co} = 7.02E1 N/m whose details are shown in Table 3. If the user selects CNC-milling as the manufacturing process, its default parameters, shown in Table 4, are displayed.

Table 2: User-specifications

	lower bound	upper bound
Input Force	2 N	5 N
Input Displacement	0.0008 m	0.0012 m
Output Force	0 N	0 N
Output Displacement	0.004 m	0.005 m
Actuation Stiffness	150 N/m	500 N/m
External Stiffness	40 N/m	40 N/m

Table 3: Details of the mechanism displayed to the user

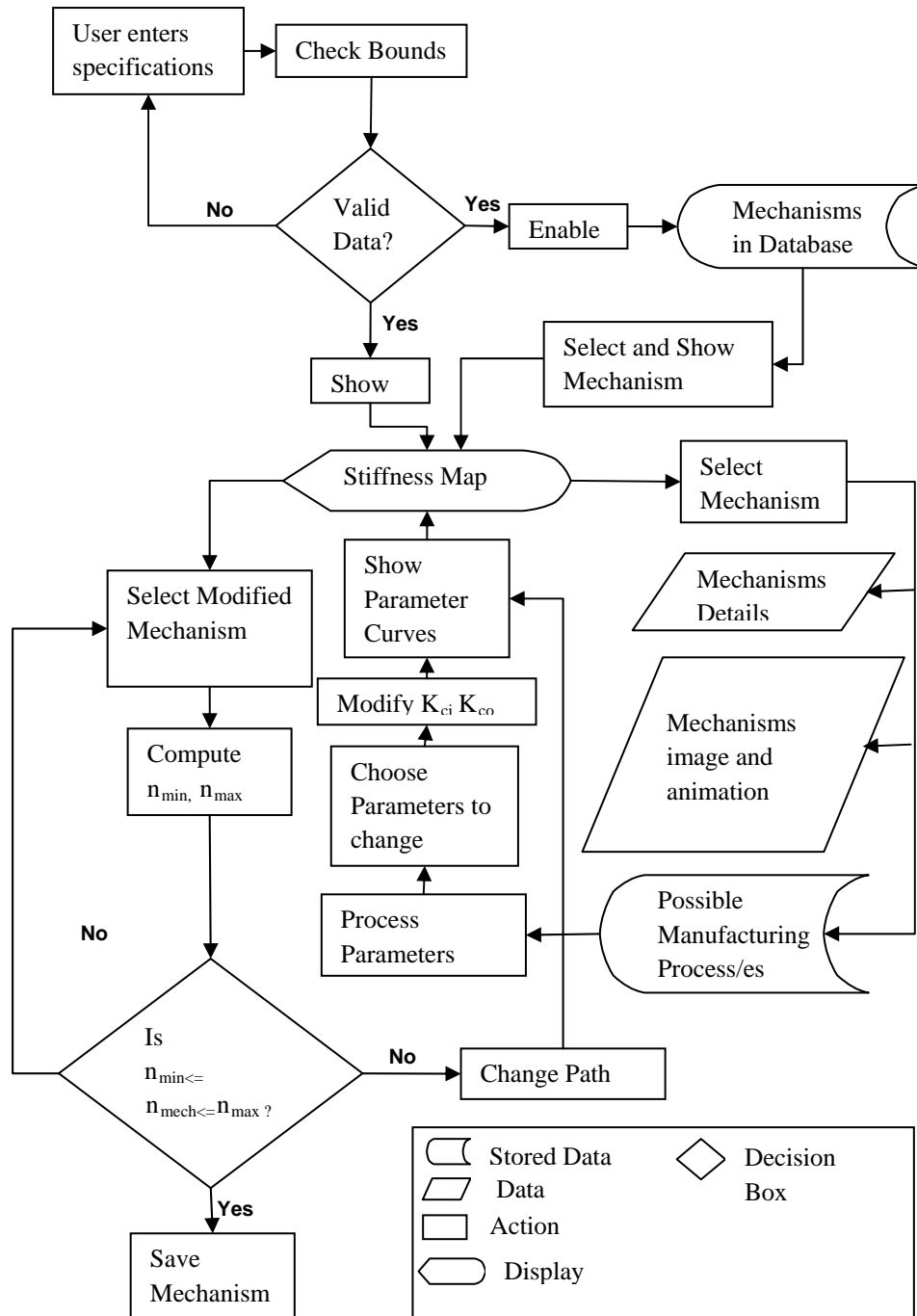| Minimum In Plane Width	2.5e-4 m
Thickness	3e-4 m
Size in X	0.103 m
Size in Y	0.063 m
Young's Modulus	2e11 N/m^2
Material	Spring steel
Maximum Stress	213.32E6 N/m^2

Figure 5: Data flow for the program

Table 4: Default values for CNC Milling shown to the user

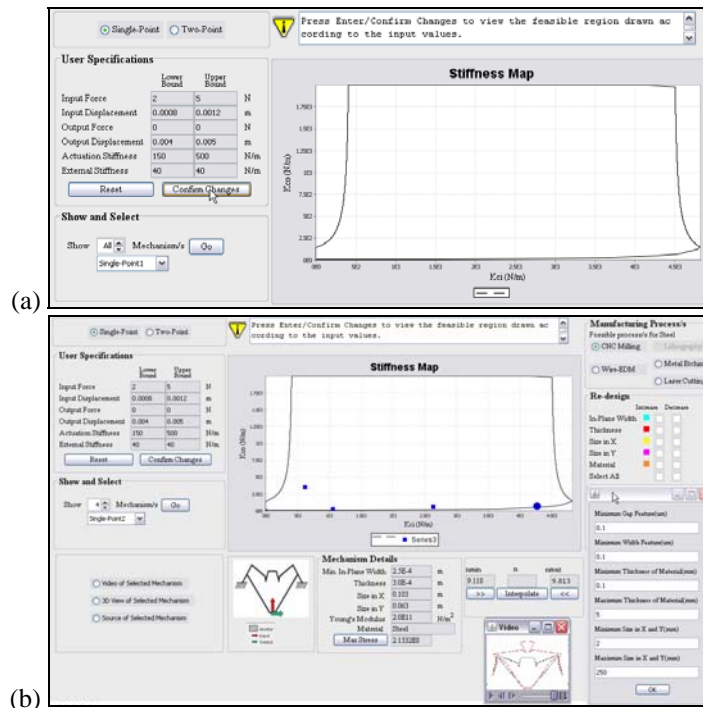| Minimum Gap Feature	0.1 mm
Minimum Width Feature	0.1 mm
Minimum Thickness	0.1 mm
Maximum Thickness	5 mm
Minimum Size in X and Y	2 mm
Maximum Size in X and Y	250 mm

(a)

(b)

Figure 6: (a) A screen-shot of the feasible map for user-specifications of Table 2; (b) with one mechanism plotted as per user's selection with the details of the chosen manufacturing process indicated in a pop-up window.

The user finds that this mechanism, although is inside the feasible region, is not suitable because its n value is not within the permissible range for that point. Hence, the user begins to explore using parameter curves and change-path until n is within the range, all the while watching the changed mechanism (see Fig. 7). The details of the re-designed mechanism are shown in Table 5.

Table 5: Mechanism's details after re-design

Minimum In Plane Width	2.25e-4 m
Thickness	3e-4 m
Size in X	0.103 m
Size in Y	0.063 m
Young's Modulus	2e11 N/m2
Material	Spring steel

Finally, the user can save a file containing information about user-specifications, the chosen manufacturing process's details, and the finalized mechanism details. To achieve this, Java is interfaced with PHP (Hypertext Preprocessor). Java web applets cannot normally read or write any files from the user's computer. So, the applet needs to have permission to access the user's computer. Moreover, when the files are on the server, each time a HTTP connection is opened in order to read from files. To avoid this, files are combined within an archive which is downloaded on the client's machine while a call from the browser to the applet is made. After clicking the 'save' icon, a text file with all the details will get saved on the client's computer in the desired location specified by the user.
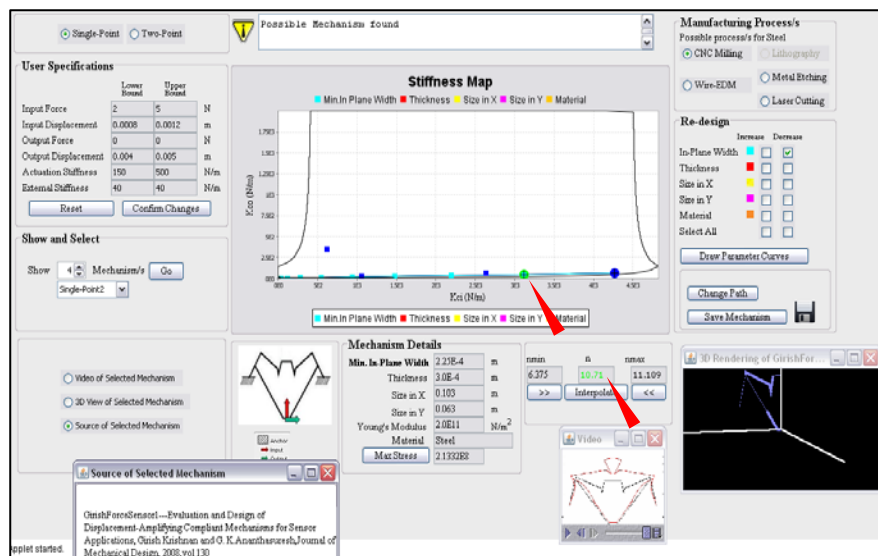

Figure 7: Screenshot of the width-parameter curve during interactive re-design. A suitable mechanism is found as indicated by one red pointer to the box showing the n value (= 10.71) in green because it is within the range (6.375 and 11.109). The other red pointer points to the green dot of the re-designed mechanism.

5   Conclusions

Systematic and easily usable design techniques help the growth of a field. By noting that compliant mechanism design techniques developed so far focus only on conceptual design or initial geometric form rather than practical constraints, this paper attempts to present a pragmatic design method via an online Java program. The purpose of this paper is to subject the free online program to review by experts in the field to have its correctness and usefulness evaluated. The method is based on recently developed spring-leverage model that enables users to select suitable compliant mechanisms from a database and interactively modifying them as needed. The details of the calculations, which involve solving two equations and 12 inequalities as well as beam element-based finite element analysis, are not included in this paper due to paucity of space. The design method accounts for space constraints, stress limits, selection of material and manufacturing technique, and

restrictions on manufacturing process parameters. The online program helps grow the database as the solutions generated by the users are added. Enhancing the collection of compliant mechanisms and classifying them as per functionality are the next steps.

Acknowledgement

References

[1] www.flxsys.com
[2] G. K. Ananthasuresh and L. L. Howell, "Mechanical design of Compliant Microsystems-A Perspective and Prospects," ASME Journal of Mechanical Design, Vol. 127 (4), 2005, pp. 736-739,
[3] L. L. Howell, *Compliant Mechanisms*, John Wiley, New York, 2001.
[4] M. P. Bendsøe and O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer, Berlin, 2003.
[5] G. K. Ananthasuresh (ed.), *Optimal Synthesis Methods for MEMS*, Kluwer, Amsterdam, 2003.
[6] C. J. Kim, Y. M. Moon, and S. Kota, "Building Block Approach towards the Conceptual synthesis of Compliant Mechanisms Utilizing Compliance and Stiffness Ellipsoids," ASME Journal of Mechanical Design, Vol. 130(2), 2008, p. 022308.
[7] www.mecheng.iisc.ernet.in/~m2d2/YinSyn
[8] www.topopt.
[9] G. N. Sandor, A. G. Erdman, and S. Kota, *Mechanism design: Analysis and Synthesi,* Prentice Hall, Upper Saddle River, NJ, 2001.
[10] S. Hegde and G. K. Ananthasuresh, "Design of Single-Input-Single-Output Compliant Mechanisms for Practical Applications using Selection Maps," ASME Journal of Mechanical Design, Vol. 132, 2010, pp. 081007-1-8.
[11] S. Hegde and G. K. Ananthasuresh, "A Spring-mass-lever Model, Stiffness and Inertia Maps for Single-Input-Single-Output Compliant Mechanisms," Mechanism and Machine Theory, (accepted).
[12] M. F. Ashby, Materials Selection in Mechanical Design, 2nd Ed., Butterworth Heinemann, New York, 1999.
[13] S. Hegde, *Design of Compliant Mechanisms using Selection Maps*, PhD Thesis, Indian Institute of Science, Bangalore, India, 2011.