Mechanical Component Design via the Internet

Donald A. Smith, Colter S. Reed
University of Wyoming

Abstract

This paper reports initial progress to implement a new paradigm for students in a typical Machine Components Design course in Mechanical Engineering curricula. The basic idea is to have design algorithms for various mechanical components (springs, gears, power transmission shafts, cams, etc.) available to students as Applets on the Internet. The students are then focused on the constraints and functional requirements associated with the particular design problem, and the algorithms attempt to design the specific component using industrial strength design algorithms. The tedium of iterative calculations is placed on the algorithms, allowing the students to concentrate on the parameters appropriate to their specific design application.

I. Introduction

The design of a mechanical component typically involves an iterative approach to determine an appropriate set of parameters that satisfy the requirements and constraints associated with the specific design problem. For example, in the design of an extension or compression spring for a particular application, the engineer must select the following parameters:

1. Type of spring to be designed
2. The material to be used to fabricate the spring
3. One or more pair of force-displacement requirements
4. Other requirements as determined by the specific design problem

From the set of design requirements, the engineer then attempts to implement the design with appropriate selection of wire size and spring geometry so that operating stresses will result in satisfactory performance. These types of design problems tend to be highly iterative. This makes them prime candidates for computer implementation once appropriate design algorithms have been identified. If the component design capability is augmented by a user-friendly interface, these programs can serve to improve the students' understanding of the design process by allowing them to focus on design constraints and objectives rather than getting bogged down in the details of the iterative calculations. This strategy also permits the students to pursue various "What If ?" investigations of their design without incurring excessive time penalties.

If the component design programs are implemented as Java Applets, students connected to the World Wide Web can access and utilize them from their Internet browser. This strategy circumvents the need to store or recompile the algorithms for individual operating environments; the appropriate algorithm is called from a single server and executed inside the user's browser.

Any software developed for use by other than the developers should incorporate several common capabilities. The user interface should enable a first time user to obtain results without hindering the progress of a more experienced user. Incorporating a comprehensive Help section, along with examples of the most significant features of the program's capabilities, can accomplish this by offering as much or as little help as the user desires. Another requirement particularly appropriate for web-based programs is straightforward navigation between the various parts of the program; with input, results, and the Help section being most important.

The strategy outlined above has been applied to a specific mechanical component, resulting in a Java Applet which designs extension springs. The Applet has been tested using the students in a Mechanical Component Design class with encouraging results. The remainder of this paper will focus on aspects of the user interface for effective communication between the student with the design problem and the design algorithms which will attempt to generate solutions to that problem.

II. The Extension Spring Design Calculator - ESDC

The Extension Spring Design Calculator (ESDC) has its roots in a program from the industrial sector that was utilized extensively in the design of extension springs for various printers and other mechanical devices associated with the computer industry. In the mid 1980's, the mainframe version of the program was upgraded and implemented as a PC-resident program with a user interface that guided the first time user while allowing the experienced user to rapidly supply the necessary design requirements to the design algorithms. The interactive version of the program was used as the base from which to generate the current Java Applet version of the extension spring design program, ESDC.

In its present configuration, ESDC permits the user to select one of six possible design cases or the analysis of an existing spring for operation in an environment other than the one for which it was initially designed. The design options are most easily explained with the aid of Table 1.

Table 1 Required Data for the Design/Analysis Cases

Case	Max Load	Max Working Length	Min Load	Min Working Length	Max or Min Rate (Choice)	Rate (Numeric)	Wire Diameter	Index
1	**Yes**	**Yes**	**Yes**	**Yes**	No	No	No	No
2	No	**Yes**	**Yes**	**Yes**	**Yes**	No	No	No
3	**Yes**	No	**Yes**	**Yes**	**Yes**	No	No	No
4	**Yes**	**Yes**	No	**Yes**	**Yes**	No	No	No
5	**Yes**	**Yes**	**Yes**	No	**Yes**	No	No	No
6	**Yes**	No	**Yes**	No	No	**Yes**	No	No
7	**Yes**	**Yes**	**Yes**	**Yes**	No	No	**Yes**	**Yes**

The "yes" and "no" entries in Table 1indicate the specific parameters which need to be specified for a particular design case. Cases 1 through 6 represent the various design situations that are covered by the design algorithms. For example, Case 1 requires that the user specify two load-length pairs. Cases 2 through 5 require one load-length pair, either the load or length at another partial pair, and the selection of a maximum rate or minimum rate spring. Required inputs for Case 6 are the two loads and a numeric value for the spring rate. In this case, the lengths at which the loads are to be produced are left as design variables. While this may seem odd, it is not uncommon in industrial practice to neglect to provide for enough physical space for a spring which must generate specific forces. This can lead to forced redesigns of the apparatus requiring the spring. Thus, case 6 provides a means of determining the space required for a spring that must provide specified forces. Case 7 provides for the fatigue analysis of a particular spring; spring index and wire diameter are specified by the user and a fatigue analysis is performed on the spring. In the design Cases 1 through 6, wire diameter and spring index are varied by the design algorithms as they attempt to effect a design compatible with the user's requirements.

In addition to selecting the design case appropriate to the problem at hand the user can select English or SI units; wire material (music wire, hard drawn wire, or oil tempered wire); end loop orientation (unimportant, 0, 90, 180 or 360 degrees); the number of standard end loops; and the end loop style. These selections are made from radio buttons or pull down menus, providing the user with a familiar operating environment based on standard Windows procedures. The familiar environment is maintained as the user specifies various parameters and choices by "graying out" options that are not available based on previous selections or specified values. The complete page for input is indicated in Figure 1.

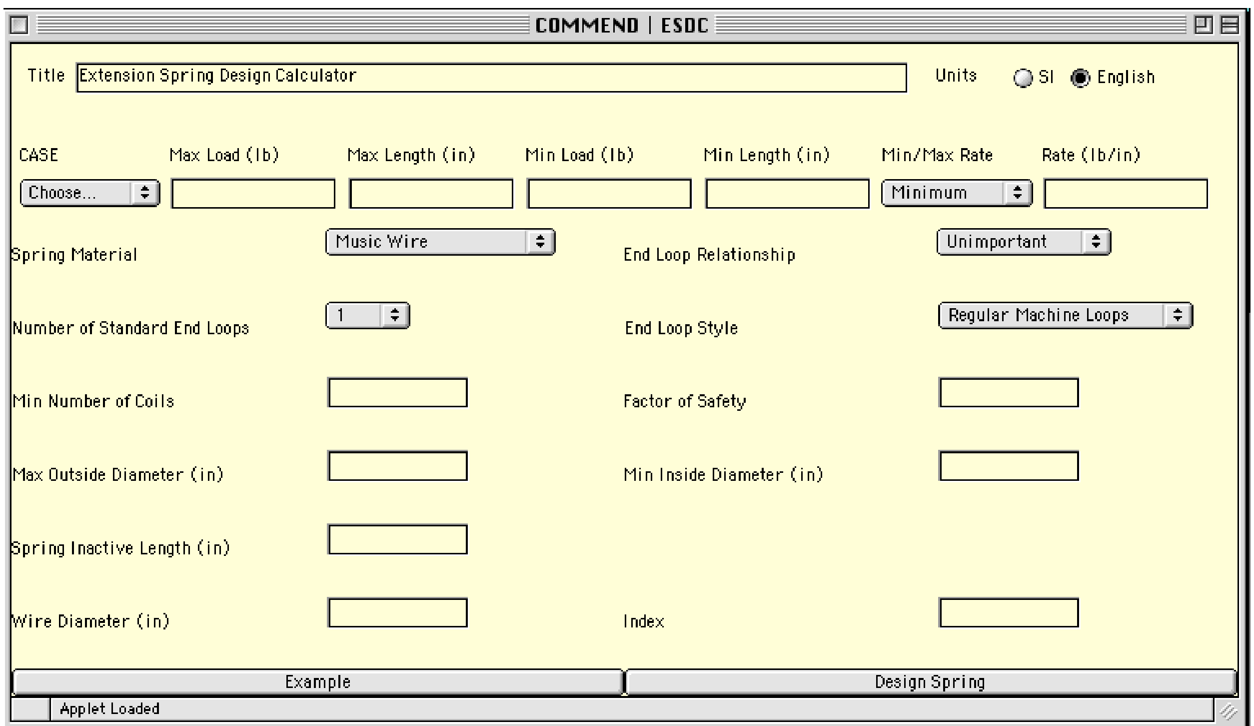


Figure 1. The Data Input-Specification Dialog Box

When the Applet first comes up, all the possibilities for input are displayed as per Figure 1. As the user starts to make choices and specify parameter values inappropriate fields and alternatives are "grayed out"--a technique utilized with most modern operating systems. For example, if the

Number of Standard End Loops is set at 2, the text "Spring Inactive Length" will be grayed out and the associated box will become inactive; i.e., the user cannot enter data in that field. Similarly, if design Cases 1 through 6 are selected, Wire Diameter and Index will be grayed out and their data fields will not be accessible. Wire Diameter and Index are varied by the design algorithms as they attempt to satisfy the user input requirements and constraints.

An example of a completed input data page is shown in Figure 2. The user has specified parameters for a Case 1 design, using SI units. The factor of safety has been selected so that only one design is possible in the range of wire diameters considered, 0.125 to 3.25 mm.

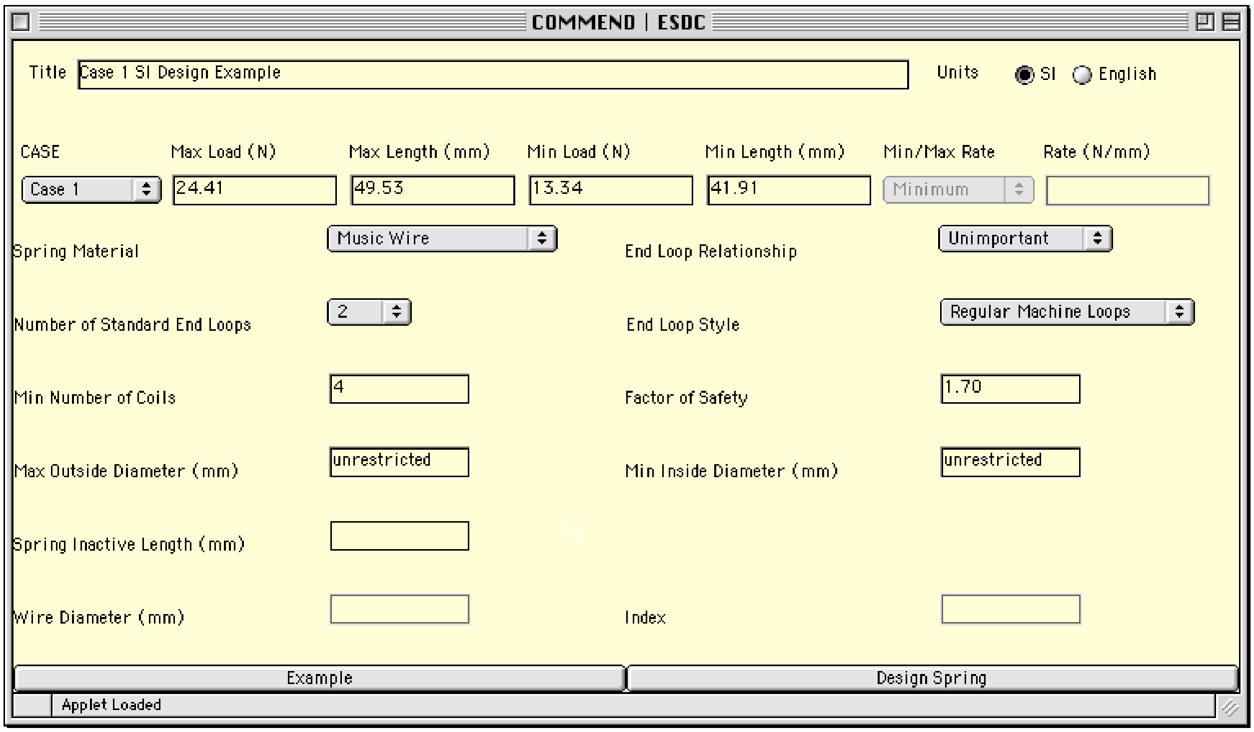
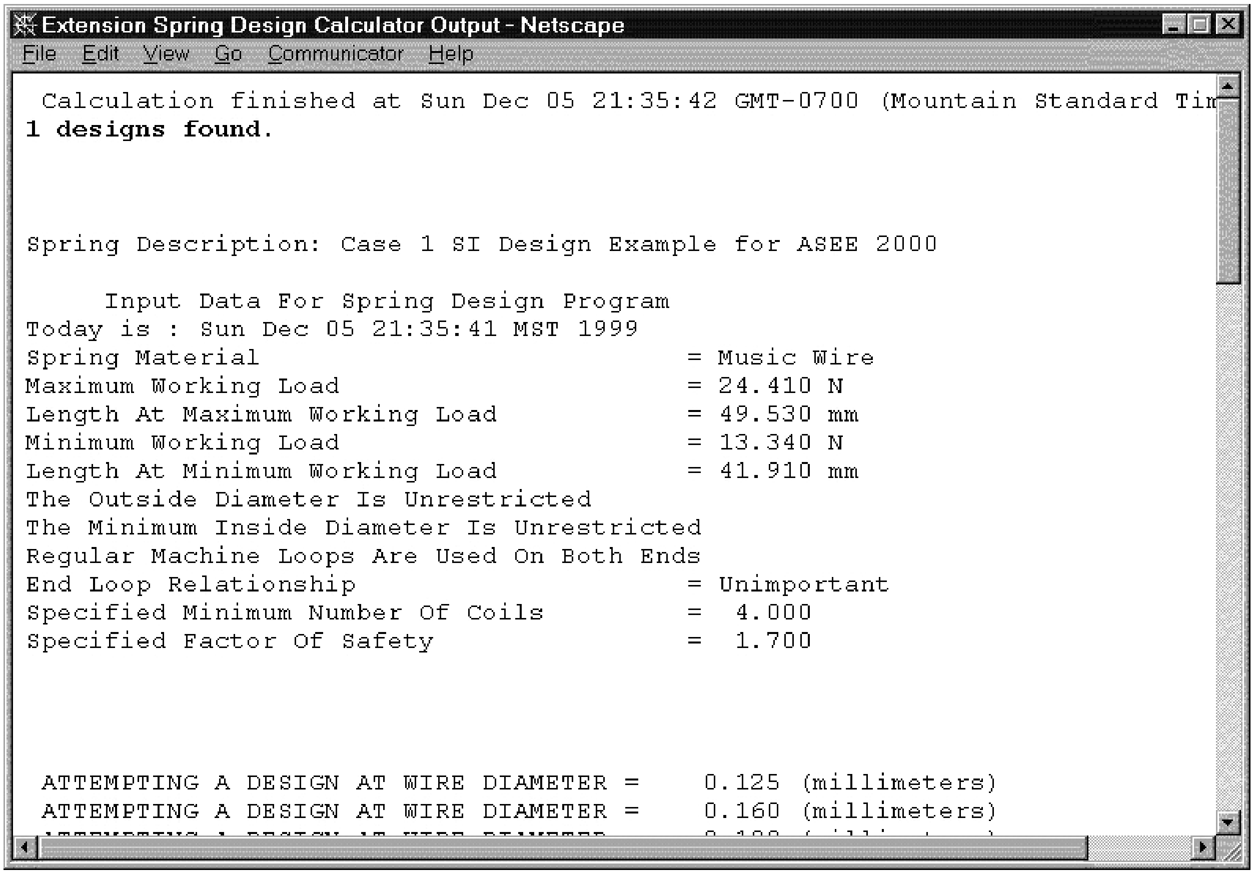


Figure 2. The Data Input Page Just Prior to Invoking the Design Algorithms

The design algorithms attempt to design springs consistent with the user specified requirements, the fatigue properties of the material selected[1] and the initial tension which is possible to wind into extension springs[2]. For the data of Figure 2, only one spring is possible and the output from the design algorithms is displayed along with an echo of the input data. The top portion of the output page is indicated in Figure 3.

The text "Attempting a design at …" was originally included to provide the user with feedback from the design algorithms, indicating that the program was progressing through the range of wire sizes appropriate to the design requirements. With Netscape 3.0, the virtual Java machine in the browser was so slow that this feedback was important. Current browsers are considerably faster than earlier versions and the incremental feedback from the design algorithms is not necessary. However, the "Attempting a design at …" text was left in the output so the user would know what was processed by the algorithms.



Extension Spring Design Calculator Output - Netscape

File Edit View Go Communicator Help

```
 Calculation finished at Sun Dec 05 21:35:42 GMT-0700 (Mountain Standard Tim
1 designs found.



Spring Description: Case 1 SI Design Example for ASEE 2000

     Input Data For Spring Design Program
Today is : Sun Dec 05 21:35:41 MST 1999
Spring Material                           = Music Wire
Maximum Working Load                      = 24.410 N
Length At Maximum Working Load            = 49.530 mm
Minimum Working Load                      = 13.340 N
Length At Minimum Working Load            = 41.910 mm
The Outside Diameter Is Unrestricted
The Minimum Inside Diameter Is Unrestricted
Regular Machine Loops Are Used On Both Ends
End Loop Relationship                     = Unimportant
Specified Minimum Number Of Coils         =  4.000
Specified Factor Of Safety                =  1.700



 ATTEMPTING A DESIGN AT WIRE DIAMETER =    0.125 (millimeters)
 ATTEMPTING A DESIGN AT WIRE DIAMETER =    0.160 (millimeters)
```

Figure 3. Output from ESDC for the Data of Figure 2

III. A Typical Utilization of the ESDC

A preliminary version of ESDC was used as part of a homework exercise in a Machine Components class during the Spring 1999 semester. The particular problem was 12-31 from the Juvinall-Marshek text[3]. In this problem, the students are to design a spring for a wheelchair brake. The problem specifications allow the students to determine a force required at a maximum extension of the spring and the minimum extension the spring will experience in its operating environment. Additionally, a constraint is placed on the maximum outside diameter of the spring. With the maximum load-length pair and the minimum operating length specified, this situation is appropriate for a Case 4 design with a minimum rate requirement to minimize the alternating component of stress. The input parameters yield several designs and the students can then opt for either maximum factor of safety or a minimum cost designs. Thus, the students have to make decisions appropriate to a realistic design situation, but they are not unduly tasked with calculations. A requirement of the assignment was to verify the computer-generated design by an analysis of the final spring.

IV. FAQ

With the background and details of the ESDC above, it is appropriate to address the rationale for this project. Several questions have been posed during the development of the ESDC and the most pertinent will be addressed in this "Frequently Asked Questions" section.

Why do this for mechanical components, i.e., the components typically studied in a Machine Component Design course?

The analysis of a particular machine component is rather straightforward. Given the loads, operating environment, material, and geometry the stresses can be calculated and various failure criteria applied to determine whether or not the component can be expected to perform satisfactorily. The design problem is far less straightforward. Given the operating environment and (hopefully) the loads acting on the part, the engineer is faced with the task of developing appropriate combinations of material and geometry that will meet the design requirements. The design process tends to be rather iterative, a task appropriate to a computer. Additionally, the contemporary engineering student has many demands on his or her time and becoming involved with repetitive calculations does not represent a particularly appropriate use of that time. Thus, the tedious iterative search for suitable geometry and/or material combinations is subcontracted to the computer and the student can focus on establishing appropriate constraints and parameters for the design.

Doesn't the student lose part of the educational experience associated with assuming values for the design parameters and having to reassign those parameters when the initial set or sets do not result in a satisfactory design?

The short answer is "No". When the Mechanical Engineering student enters the industrial workplace, he or she will most likely have access to in-house or third-party component design programs. One must remember that in the design of a complete machine, the design of each of the individual components is a very small percentage of the overall task. The design of a spring, gear, cam, or power transmission shaft may be an appropriate homework assignment for a Machine Components Design class, but it cannot take a significant amount of time in the industrial environment if the product is to generated in a timely fashion. Students utilizing computer-aided design tools such as ESDC are obtaining an industrial perspective during the course of their undergraduate program. Additionally, if the course instructor insists that the students verify the results of a computer generated design, the students are forced to do an analysis that will involve all the parameters associated with the design of the component, thereby reinforcing the interrelationships of the various parameters.

Why a Java Applet?

The attractiveness of implementing the design algorithms as Java Applets is associated with the Java Virtual Machine located in current web browsers. Anyone connected to the web can download the Applets into their browser and run the design algorithms locally, on their client machine. When the design task is completed, the Applet does not need to be stored locally; it will be available from the remote server whenever needed. The Applet implementation provides a one time, one way, download of the design algorithms. There is no need for two-way traffic over the 'net. A major benefit of this strategy is that there is only one copy of the source code to deal with for enhancements or the inevitable glitches.

V. Conclusion

The spring design Applet ESDC represents the first component design algorithm of what should evolve into a library of Java based programs for the design of other components. Efforts are already underway for spur gear and compression spring design programs.

Anyone involved with the instruction of a Mechanical Component Design course is welcome to utilize ESDC, which is available at http:// wwweng.uwyo.edu/commend/.

VI. Acknowledgements

The authors would like to acknowledge the contributions of all the other individuals who were instrumental in the development of the current level of the ESDC. The project was initiated in the spring of 1998 with Dr. Aleksander Malinowski (now at Bradley University) and undergraduate students Shawn Cretti, Katarzyna Wilamowska, and Alex Dickson. Shawn and Kasia completed the design algorithms for the various cases and initiated work on the user interface. Hongze Lai, a former graduate student, developed the Applet that displays the Modified Goodman Diagram for the Case 7 Analysis capabilities. Huiyuan Ma, a current U.W. graduate student, accomplished the final debug of the graphic applet and implemented the graying out feature of the input screen. The first author's Spring 99 class in Machine Component Design contributed many constructive suggestions regarding the interface, Help section, and other aspects of the program. Everyone's contributions are very much appreciated.

Bibliography

1. *Manual on the Design and Application of Precision Mechanical Springs*, Michigan Spring Company, Muskegon, Michigan, July 1984.
2. Chironis, N., *Spring Design and Application*, McGraw-Hill Book Company, Inc. New York, 1961.
3. Juvinall, R.C. & Marshek, K.M., *Fundamentals of Machine Component Design*, 2nd ed., John Wiley & Sons, 1991.

DONALD A. SMITH
Don Smith is currently a Professor of Mechanical Engineering at the University of Wyoming. He received his BSME and M.Sc. degrees from The Ohio State University and his Ph.D. from the University of Michigan. Prior to joining the University of Wyoming, he was employed with Xerox and IBM. His academic interests are in the areas of Mechanical Component Design, Kinematics and Dynamics of Machinery, and Computer Aided Design. Current research activity is focused on web based mechanical component design.

COLTER S. REED
Colter Reed is completing his BS in Computer Science and Electrical Engineering with the Computer Engineering Option at the University of Wyoming. He has worked in the College's Engineering Science Interactive Graphics Laboratory for several years while pursuing his undergraduate degrees.