Int J Interact Des Manuf (2009) 3:265–272
DOI 10.1007/s12008-009-0079-3

ORIGINAL PAPER

Methodology to build an assistance tool dedicated to preliminary design: application to compression springs

Manuel Paredes

Received: 24 March 2009 / Accepted: 24 August 2009 / Published online: 16 September 2009

Abstract This paper proposes a method to develop a design assistance tool dedicated to preliminary design. This tool increases interactivity and enables the design space to be explored. The user-friendly interface proposes to define design parameters with interval values. In addition, an objective has to be selected and the tool offers an optimum design. To take full advantage of all the exploration's capabilities, it is suggested that all the potential optimization functions be considered as design parameters and that each design parameter be considered as a potential objective function. This method was applied to build a software mock-up dedicated to compression spring design. The industrialization process leading to the commercial software is detailed.

Keywords Design space exploration · Optimization · Assistance tool · Mechanical springs · Industrialization

1 Introduction

The design of machines imposes the dimensioning of numerous common mechanical components (gears, cams, shafts, springs, etc.). These components have their own dimensioning rules and require specific manufacturing knowledge. The problem of designing a mechanical component is often solved by using tables and charts for certain pre-selected specifications and objectives. These calculations can be carried out manually, but without computer assistance, designers are often obliged to oversimplify the procedures, e.g. by assigning a value to certain parameters in order to reduce the number of problem variables to only 2 or 3 [1–3]. They are therefore unable to make the most of all the specification possibilities and consequently optimize design. Progress in Computer Assisted Design should lead to successful solutions to this kind of problem.

Many classes of assistance tools can be found. The greatest range of industrial software for component design can be considered as analyzing tools. Some tools can be used for the design of many mechanical components. MITCalc (www.mitcalc.com) contains both design and check calculations for many common tasks. KISSsoft is a calculation program consisting of a standard package and expert add-ons for the certification and design of machine elements (www.kisssoft.ch). Tumkor [4] presents an interactive website to assist in designing shafts and bearings. Other tools focus on the design of a specific mechanical component. As examples, Zakegear calculators (www.zakgear.com) are online tools related to the design of gears that apply the work of Dudley [5] and the Institute of Spring Technology provides the Spring Design and Validation Software (www.ist.org.uk).

All these analyzing tools provide a first level of assistance. The main idea is to avoid the user having to perform tedious calculations. The designer has to propose a design that is analyzed by the tool. As for the Spring Design and Validation Software from IST presented in Fig. 1, the tool interface commonly proposes to enter data in order to fully define a design. The proposed data are the most common in use. When several applications are proposed, the tool allows a choice to be made from within several sets of data. For a given application, the designer has to choose the most appropriate set of data to define his design. The tool then analyzes the solution. It calculates several parameters and can evaluate whether the solution is acceptable or not with regard to standards and/or the manufacturing process. The designer then has to himself analyze the other calculated parameters in order to

---

M. Paredes (✉)
INSA, UPS, Mines Albi, ISAE, ICA (Institut Clément Ader),
Université de Toulouse, 135, avenue de Rangueil,
31077 Toulouse, France
e-mail: manuel.paredes@insa-toulouse.fr

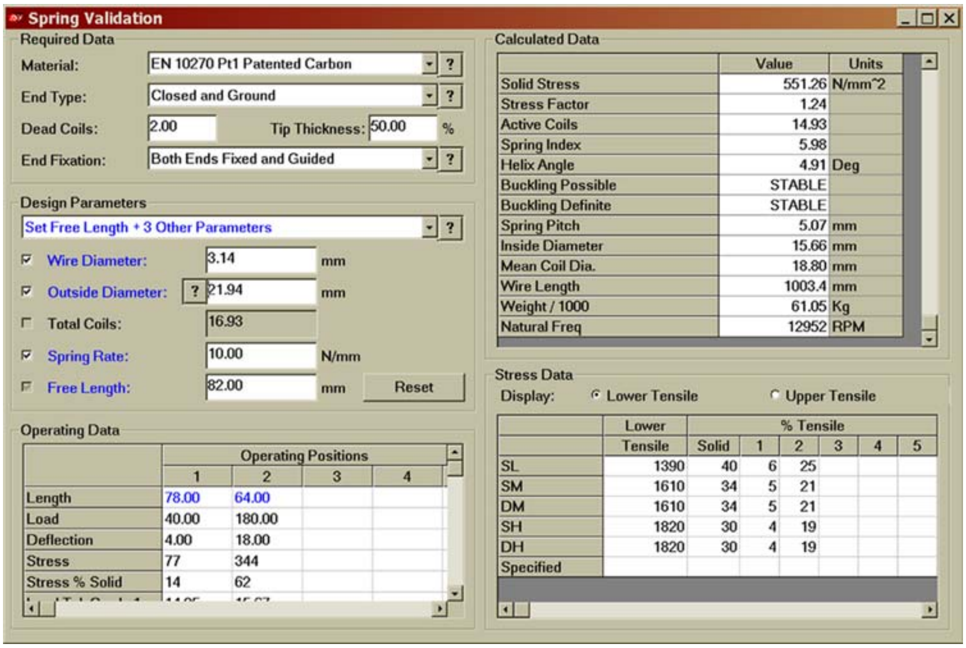


Fig. 1 Spring Validation Software from IST

Required Data

Material: EN 10270 Pt1 Patented Carbon

End Type: Closed and Ground

Dead Coils: 2.00 Tip Thickness: 50.00 %

End Fixation: Both Ends Fixed and Guided

Design Parameters

Set Free Length + 3 Other Parameters

☑ Wire Diameter: 3.14 mm

☑ Outside Diameter: 21.94 mm

☐ Total Coils: 16.93

☑ Spring Rate: 10.00 N/mm

☑ Free Length: 82.00 mm

Operating Data

	Operating Positions			
	1	2	3	4
Length	78.00	64.00		
Load	40.00	180.00		
Deflection	4.00	18.00		
Stress	77	344		
Stress % Solid	14	62		

Calculated Data

	Value	Units
Solid Stress	551.26	N/mm^2
Stress Factor	1.24	
Active Coils	14.93	
Spring Index	5.98	
Helix Angle	4.91	Deg
Buckling Possible	STABLE	
Buckling Definite	STABLE	
Spring Pitch	5.07	mm
Inside Diameter	15.66	mm
Mean Coil Dia.	18.80	mm
Wire Length	1003.4	mm
Weight / 1000	61.05	Kg
Natural Freq	12952	RPM

Stress Data

Display: ● Lower Tensile ○ Upper Tensile

	Lower	% Tensile					
	Tensile	Solid	1	2	3	4	5
SL	1390	40	6	25			
SM	1610	34	5	21			
DM	1610	34	5	21			
SH	1820	30	4	19			
DH	1820	30	4	19			
Specified							

see whether the proposed design satisfies his requirement. If the design is acceptable, then the designer usually stops the design process and validates the design. If the design is not valid, the designer modifies data, performing a trial and error process. The success of such a process primarily depends on the designer's background. An experienced designer will easily modify data in order to find an acceptable solution. Moreover, an analyzing tool can hardly be operated in the early design stages. Indeed, especially in the embodiment design phase [6], there is always a host of parameters that have not been yet been set, and it is therefore difficult to give the fixed values required as data for this kind of tool.

To improve design interactivity, synthesis tools can be used. In such tools, the designer defines his requirements and the tool proposes a satisfying design as a result when possible. The main difficulty for designers using synthesis tools is in defining their needs. It is not always easy for a designer to identify the relevant data, especially in the early design stages. It often arises that the requirements given by the designer lead to no acceptable solution. Thus, a tool that would help the designer to give data would be of great interest. Such a tool would increase interactivity during the design process. One solution is to build a tool that interactively provides an overview of the design space. When new data is given, the tool recalculates and presents the corresponding new design space. Using such a tool, the designer would be able to readily identify data that have the greatest effect on his design and maintain consistency all along the design process.

The Ohio State University proposes the Run Many Cases Program (www.gearlab.org) which provides a visualisation of the design space for the design of gears. Many researchers have worked on systems to represent the allowable design space or the Pareto solutions space [7,8]. Design space exploration has also often been applied to the design of processor architectures [9–11]. Negotiation indicators [6] can also be exploited to explore the design space by means of a class-based approach.

We propose a method to build an assistance design tool dedicated to preliminary design. This tool should be able to consider uncertain data, should perform design space exploration and also propose an optimal design. The first part of the paper presents the methodology of our proposed synthesis tool. The second part shows how it has been used to build a mock-up dedicated to compression spring design. The industrialization process from the research mock-up to the industrial software is also presented.

2 Methodology for the assistance tool

The goal is to allow an assistance tool dedicated to a mechanical component to be built. To do so, the window interface comprises several areas to enter data and show results.

Three areas are defined to enter data. All the fixed data are set in the first area. This area comprises all data that are to be set and that do not change during a calculation. It can be the material, a configuration or any other data that have to be fixed so as to be able to perform calculation. The title or reference of the calculation also has to be included in this area. It thus comprises check boxes, option boxes, combo boxes or text boxes.

The second area is the main one for data. This area includes a specification sheet where all the design parameters that can be considered appear. Dore [12] proposes an approach to identify design variables and criterion variables in preliminary design. Here, it is proposed to consider each variable (design or criterion) as a parameter that can be defined by a lower and/or an upper value. The list of considered parameters has to be exhaustive in order to cover the widest possible range of applications and be applied at any design step. Such a specification sheet enables a broad range of problems to be defined since the designer can readily give each parameter a lower limit, an upper limit, an interval value, or a fixed value by giving the same value to the lower and upper bounds, or nothing by leaving the bounds empty.

The third area enables the objective function for design to be defined.

This area is often not featured in synthesis tools. Indeed, a synthesis tool can automatically stop the resolution process when an acceptable solution is found and propose this solution as a result. Some tools have an implicit, unique objective function. For example, the tool developed by Wong [13] for the design and optimization of industrial silencers merely proposes to minimize the total manufacturing cost and the Spring Design Software distributed by the Spring Manufacturers Institute was able to automatically adjust the wire diameter value in order to minimize spring weight. In our view, defining an objective function is of considerable interest for designers, as for a given set of requirements many different designs may be acceptable. The objective function enables the tool to determine the most suitable solution for the designer. Depending on the context this can involve mono- or multi-objective optimization. The main final objective functions usually consist of minimizing weight, reducing cost or maximizing reliability. Admittedly, it is often premature to define such an objective function from the functional requirements at the early design stages [8]. Nevertheless, we consider that defining intermediate objective functions can greatly help designers to draw up their specification sheet. For this reason, we suggest that each design parameter can be considered as an objective function to be minimized or maximized. The main advantage of such an idea is that it enables the designer to explore the design space by selecting an appropriate objective function and evaluate the corresponding result throughout the process of drawing up the functional requirements. Once the requirements have been fully defined, the final objective function can be applied to propose the final design. Designers may also wish to manage multi-objective optimization, but using the proposed methodology, only one unique objective function can be selected at a time. However, there are many ways to deal with multi-objective optimization [14]. To enable designers to manage multi-objective optimization easily, we propose to consider each potential objective function as a design parameter. Thus, bounds can be set for each potential objective and it becomes possible to explore one objective function while assigning bounds to the others. In so doing, designers can operate the tool to explore their multi-objective space and analyze the associated Pareto solutions. Figure 2 illustrates the proposed process for exploring the Pareto space for two objective functions that have to be minimized. Figure 2a shows the allowable multi-objective space resulting from the functional requirements and the Pareto solutions are shown by the thick line. An upper bound to the second objective function is added to the requirements in Fig. 2b. This reduces the solution space. The result of the optimization process in minimizing the first objective belongs to the Pareto solutions space. Modifying the upper bound value of the second objective enables the Pareto solutions to be explored.

To conclude, in order to make the most of the proposed design tool, each potential objective function has to be considered as a design parameter and all the design parameters must be added to the list of potential objective functions.

The results are shown in two areas. The first result area presents the proposed design that has to satisfy the requirements and correspond to the selected objective. This area contains data that are usually found as a result for an analyzing tool. Thus, the individual value of each design parameter shown in the specification sheet has to be given. Moreover, this result area can display the manufacturing parameters that correspond to the design or details of calculation that shows respect for standards. In this context, highlighting the objective function value enhances legibility.

Proposing a design that matches the requirements and respects standards and manufacturing constraints is not an easy matter. Moreover, in our suggested tool, the proposed design has not only to be acceptable but also has to minimize or maximize the objective function selected by the designer. To do so, powerful capabilities of optimization strategies can be harnessed as the requirements defined by the user can be translated into an optimization problem. The objective function is selected by the designer. The variables are then

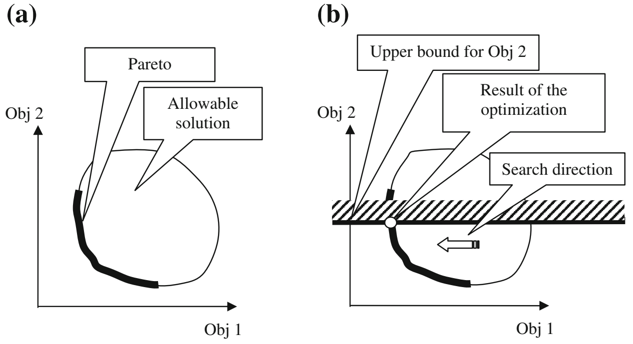


Fig. 2 Exploring the Pareto space using one objective function at a time

chosen in order to fully define a design. Each limit given in the specification sheet for a design parameter is used as a constraint for the optimization problem. In addition, the constraints related to standards or manufacturing knowledge have to be automatically considered.

Many techniques exist to solve an optimization problem. The main difficulty is to find a reliable method that is capable of solving the problem whatever the specifications. At this stage, it is impossible to introduce just one approach that would allow all problems to be solved. The resolution process mainly depends on the nature of the optimization problem. When continuous variables and functions (objective and constraints) are considered, mathematical programming methods [15] can be applied. When considering other types of variables (discrete, integer, Boolean, etc.) stochastic approaches are more appropriate, as with genetic algorithms [16], evolutionary strategies [17] or neural network algorithms [18]. Finally, for mixed variable problems, several strategies can be applied. Mathematical programming and stochastic algorithms can also be combined [19] or Metamodeling used [20].

Details for applying mathematical programming can be proposed. For continuous variables and functions, a direct method can be implemented as each step in the resolution process is based on a displacement inside the solution area. This kind of property is useful if the resolution process is interrupted before completion (when a full convergence proves to be difficult), the tool being more likely to provide an adequate though non-optimal solution. Direct methods require a starting point inside or close to the solution area. Indeed, the closer the starting point is to the final solution, the more likely the algorithm will be to converge towards the optimum solution. For this reason, we propose several strategies to find a good starting point. The first involves giving each considered variable the medium value of the bounds given in the specification sheet. This basic method is efficient when most variable bounds are defined. Clearly, it can lead to a non acceptable initial solution when there are few data available. To be more efficient, another strategy involves selecting the best existing design related to the requirements from a catalogue. Depending on the specifications and on the database, it may arise that no allowable design is found. The closest design to the specification can then be chosen as a starting point. This method can be improved by automatically defining a catalogue depending on the requirements. To do so, interval arithmetic [21,22] is then useful.

The second result area shows a higher level of assistance. This area proposes an overview of the design space. This can be obtained by presenting the acceptable bounds of each design parameter considered in the specification sheet. This area can be updated at each data modification. In so doing, the designer has an interactive overview of the design space. Showing the acceptable bound of a parameter involves representing the final value of an objective function as a result of the optimization process shown below where the parameter bound is considered as the objective [23]. Thus, an optimization problem has to be solved for each bound to be presented. This process can be time consuming and the approach can therefore be applied most successfully when analytical formulae or low numerical processes can be implemented. The design space is thus represented by it bounds (not only on the manufacturing variables but on each design parameter) but no potential sub-spaces can be highlighted. Internal exploration can be manually performed step by step by modifying the bounds in the specification sheet.

Finally, this level of assistance also enables the resolution process to be facilitated by mathematical programming. The database from which the initial design is selected can be increased by keeping the design results for each run performed to show the design space. Indeed, interactively displaying the design space means finding the minimum and maximum allowable values for each design parameter. When a new data item is given, the design space can be reduced, but at least one design remains acceptable. For example, when the maximum value for a parameter is given (or modified) by the designer then the design corresponding to the opposite value of this parameter (the minimum) remains acceptable. This method means an acceptable design to initialize variables is sure to be found.

3 Software mock-up for compression spring design

The method proposed to define an assistance tool was applied to cylindrical compression springs with constant pitch. Defining a compression spring design means finding values for manufacturing parameters to define spring geometry and operating parameters to define its use. The main parameters are shown in Figs. 3 and 4.

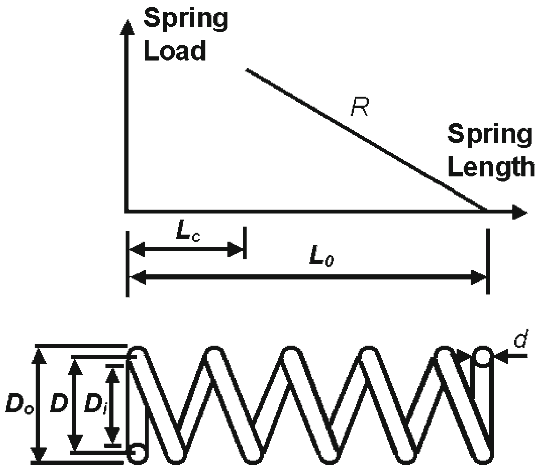


Fig. 3 Manufacturing parameters

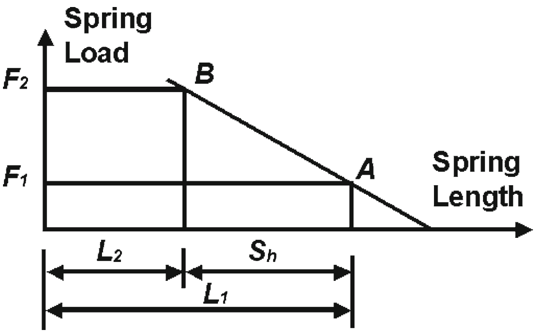


Fig. 4 Operating parameters

The other considered design parameters are:

- Spring mass
- Natural frequency of surge waves
- Energy stored during travel
- Housing diameter
- Overall volume at free length
- Overall volume when compressed

Parameters to respect standards and manufacturing requirements are also considered:

- Minimum allowable length
- Spring index
- Helix angle
- Fatigue life factor
- Stress at solid
- Buckling length

To fully define a design some other characteristics have to be set:

- Objective function
- Material
- Type of ends
- Number of dead coils
- Number of cycles
- Selected standard to calculate the minimum allowable length
- Allowable stress at solid
- The end fixation factor to calculate the buckling length
- Whether the helix angle has to be checked
- Whether the stress correction factor has to be applied
- Required level of assistance

Considering this, an optimization problem can be automatically built. Optimum compression spring design has often been applied to illustrate mixed variable optimization techniques [24–26]. The authors consider that the number of coils is an integer variable, that the wire diameter is a discrete variable and that the other variables are continuous. From a practical point of view, all the variables can be considered as continuous. Indeed, spring manufacturers can build springs with a continuous number of coils. Furthermore, while standard wire diameters do exist, it is possible to ask the wire manufacturer to provide a specified wire diameter. The optimization problem thus comprises six continuous variables that enable not only the spring itself (four manufacturing parameters) but also its use (two operating parameters) to be defined together with a maximum of 47 constraints. The objective function is selected from the considered list of parameters. It is thus analytically defined (all the considered parameters are defined analytically).

This optimization problem has to be solved automatically. The goal here is to find a fast, reliable and comprehensive method to solve our problem automatically, whatever the specifications. As continuous variables and functions have to be managed, mathematical programming is used and, more specifically, the DOT algorithm [15]. This algorithm applies the Sequential Quadratic Programming method and appears to be extremely efficient. In order to initialize variables, interval arithmetic [21] has been used to build a virtual catalogue of springs [27]. The algorithm proposed by Paredes [28] allows the optimal values of operating variables for each virtual spring to be calculated automatically and finally the best design to initialize the variables to be selected.

As the resolution process is extremely rapid, the highest level of assistance was implemented. The result area shows the values of all design parameters as well as constraint parameters. An overview of the design space is also proposed showing the allowable bounds of all the design parameters.

The final proposed window interface is shown in Fig. 5.

In addition, sensitivity analysis can be performed in order to check the robustness of the design [9]. Paredes [27] presents details of the sensitivity study and on building of the software mock-up.

4 From the mock-up to the industrial software

The research mock-up dedicated to compression spring design was presented to a set of industrial designers from our partner, the Schneider-Electric company. Their feed-back was very instructive. It appears that even expert designers have never been able to fully define their specifications (constraints and objective function) in one run and that the results given by the tools greatly help them. The tool been very quick, an interactive dialog between the designer and the tool takes place and the final design is mostly obtained after three or four runs. The first wish is thus reached as the proposed mock-up enables increasing design interactivity.

It also appears that almost all the designers quickly wish to manage multi-objective optimization. As this functionality

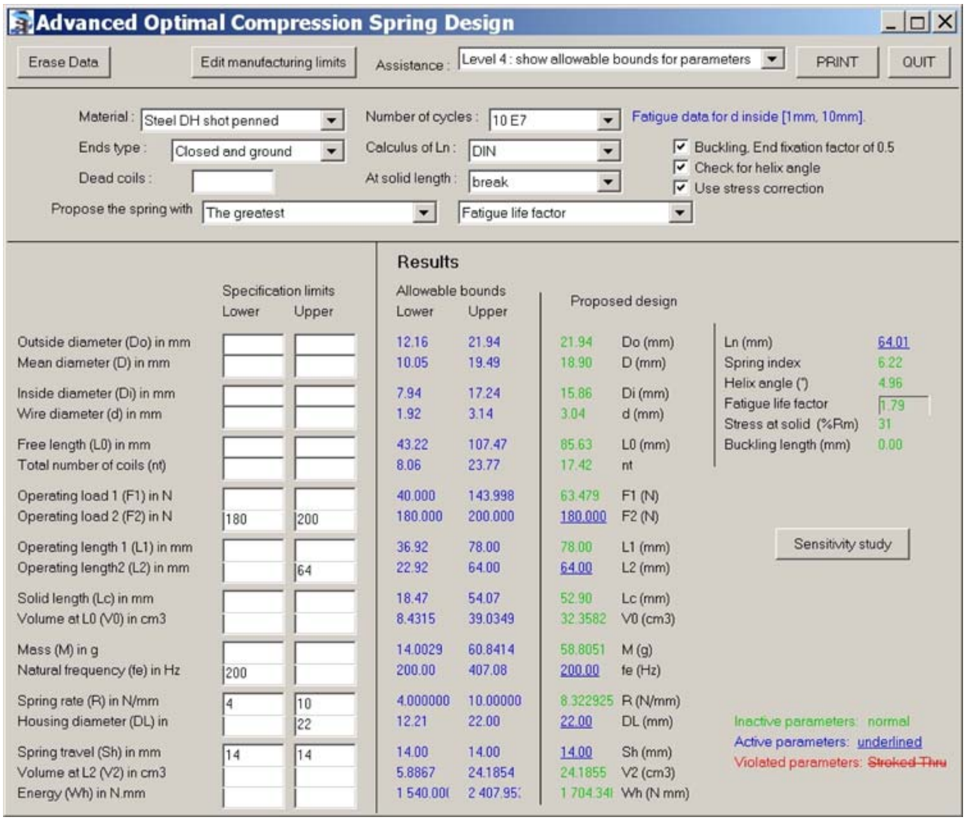


Fig. 5 Software mock-up for optimum spring design

Assistance : Level 4 : show allowable bounds for parameters

Material : Steel DH shot penned
Ends type : Closed and ground
Dead coils :
Propose the spring with The greatest Fatigue life factor

Number of cycles : 10 E7
Calculus of Ln : DIN
At solid length : break

Fatigue data for d inside [1mm, 10mm].

☑ Buckling. End fixation factor of 0.5
☑ Check for helix angle
☑ Use stress correction

	Specification limits		Allowable bounds		Proposed design	
	Lower	Upper	Lower	Upper		
Outside diameter (Do) in mm			12.16	21.94	21.94	Do (mm)
Mean diameter (D) in mm			10.05	19.49	18.90	D (mm)
Inside diameter (Di) in mm			7.94	17.24	15.86	Di (mm)
Wire diameter (d) in mm			1.92	3.14	3.04	d (mm)
Free length (L0) in mm			43.22	107.47	85.63	L0 (mm)
Total number of coils (nt)			8.06	23.77	17.42	nt
Operating load 1 (F1) in N			40.000	143.998	63.479	F1 (N)
Operating load 2 (F2) in N	180	200	180.000	200.000	180.000	F2 (N)
Operating length 1 (L1) in mm			36.92	78.00	78.00	L1 (mm)
Operating length2 (L2) in mm		64	22.92	64.00	64.00	L2 (mm)
Solid length (Lc) in mm			18.47	54.07	52.90	Lc (mm)
Volume at L0 (V0) in cm3			8.4315	39.0349	32.3582	V0 (cm3)
Mass (M) in g			14.0029	60.8414	58.8051	M (g)
Natural frequency (fe) in Hz	200		200.00	407.08	200.00	fe (Hz)
Spring rate (R) in N/mm	4	10	4.000000	10.00000	8.322925	R (N/mm)
Housing diameter (DL) in		22	12.21	22.00	22.00	DL (mm)
Spring travel (Sh) in mm	14	14	14.00	14.00	14.00	Sh (mm)
Volume at L2 (V2) in cm3			5.8867	24.1854	24.1855	V2 (cm3)
Energy (Wh) in N.mm			1 540.00(	2 407.95:	1 704.34I	Wh (N mm)

Ln (mm)	64.01
Spring index	6.22
Helix angle (°)	4.96
Fatigue life factor	1.79
Stress at solid (%Rm)	31
Buckling length (mm)	0.00

Inactive parameters: normal
Active parameters: underlined
Violated parameters: ~~Stroked Thru~~

doesn't directly appear in the tool, we had to explain our strategy that enables selecting one objective at a time (we have to explain here that designers are not aware of optimization techniques).

Finally, the mock-up proved efficient enough to justify the Schneider-Electric company supporting the transfer from the research mock-up to an industrial software application.

For an industrial user, the software has to respect certain key issues:

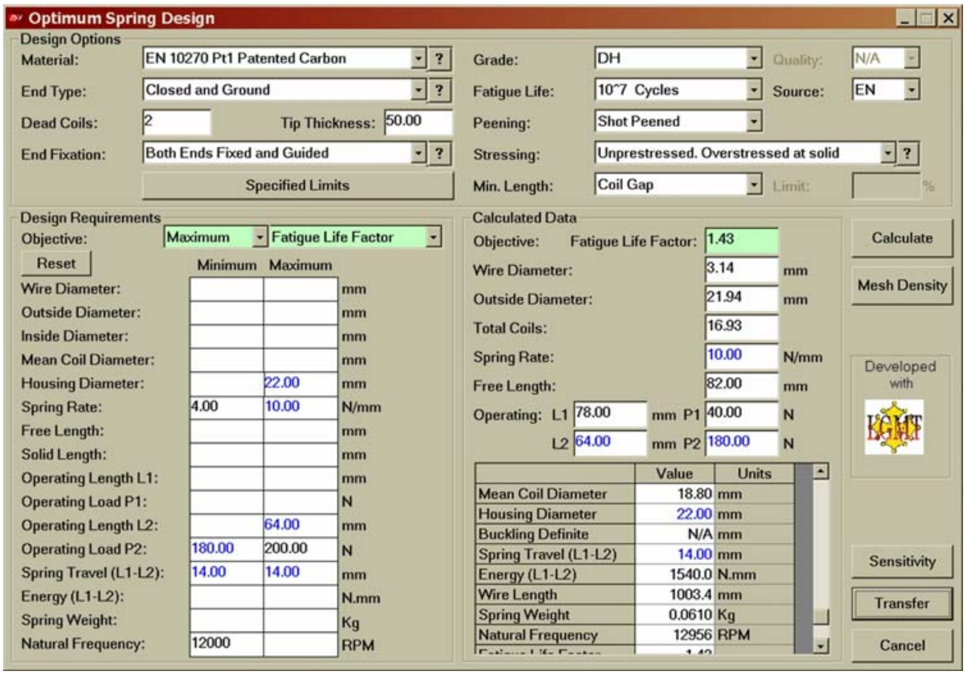
- It must remain effective whatever changes made in computer systems
- It has to be compatible with other existing software
- Its interface has to be capable of being modified to respect designer requirements
- It has to be guarantee a service life over a minimum number of years

In addition, the tool is expected to be operated worldwide. Thus, the material database has to be significantly refined and expanded.

For these reasons, co-operation was pursued with the Institute of Spring Technology. Indeed, this institute already develops and sells validation software related to spring design. Thus, the Institute has taken responsibility for software development and distribution problems that may arise. IST also undertakes long-term research projects into fundamental aspects of spring technology and high strength materials. They thus maintain a substantial material database that takes European, US and Japanese standards into account.

The main difficulty that has to be overcome relates to the optimization core. Indeed, the optimization algorithm applied in the software mock-up belongs to the Vanderplaats Research and Development Company. Unfortunately, the licence price for this efficient algorithm for the industrial tool would have been higher than the expected selling cost of the entire final software. At this stage, a large number of algorithms have been tested, the determining criteria being that they have to be efficient, easily implemented and have an acceptable cost.

The solution we found involves using the *fmincon* function from the Matlab Optimization Toolbox (www.mathworks.com/products/optimization). *fmincon* is a numerical solver that finds the minimum (or maximum) value of constrained nonlinear multivariable functions. It uses a Sequential Quadratic Programming method. In this method, a Quadratic Programming sub-problem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula [29,30]. This kind of algorithm is capable of solving our optimisation problem. Several tests were performed in order to ensure its accuracy.



Optimum Spring Design

Design Options

Material: EN 10270 Pt1 Patented Carbon
End Type: Closed and Ground
Dead Coils: 2
Tip Thickness: 50.00
End Fixation: Both Ends Fixed and Guided
Grade: DH
Quality: N/A
Fatigue Life: 10^7 Cycles
Source: EN
Peening: Shot Peened
Stressing: Unprestressed. Overstressed at solid
Min. Length: Coil Gap
Limit: %

Design Requirements

Objective: Maximum Fatigue Life Factor

	Minimum	Maximum	
Wire Diameter:			mm
Outside Diameter:			mm
Inside Diameter:			mm
Mean Coil Diameter:			mm
Housing Diameter:		22.00	mm
Spring Rate:	4.00	10.00	N/mm
Free Length:			mm
Solid Length:			mm
Operating Length L1:			mm
Operating Load P1:			N
Operating Length L2:		64.00	mm
Operating Load P2:	180.00	200.00	N
Spring Travel (L1-L2):	14.00	14.00	mm
Energy (L1-L2):			N.mm
Spring Weight:			Kg
Natural Frequency:	12000		RPM

Calculated Data

Objective: Fatigue Life Factor: 1.43
Wire Diameter: 3.14 mm
Outside Diameter: 21.94 mm
Total Coils: 16.93
Spring Rate: 10.00 N/mm
Free Length: 82.00 mm
Operating: L1 78.00 mm P1 40.00 N
L2 64.00 mm P2 180.00 N

	Value	Units
Mean Coil Diameter	18.80	mm
Housing Diameter	22.00	mm
Buckling Definite	N/A	mm
Spring Travel (L1-L2)	14.00	mm
Energy (L1-L2)	1540.0	N.mm
Wire Length	1003.4	mm
Spring Weight	0.0610	Kg
Natural Frequency	12956	RPM

Fig. 6 Optimum Spring Design Software from IST

The Matlab function *fmincon* was chosen for another specific and strategic reason. The IST industrial software is coded using Visual Basic. The Matlab Builder (www.mathworks.com/products/netbuilder/) automatically generates independent Common Object Model (COM) objects from Matlab functions such as *fmincon* and a COM object can be called from any COM-compliant technology, such as Visual Basic. As it is possible to convert a Matlab program into a self-contained application or a software component with the Matlab Compiler (www.mathworks.com/products/compiler/), it is also possible to link a component compiled from a Matlab function to a Visual Basic program. Moreover, applications using Matlab Builder components do not require Matlab to be installed, and there is no additional cost for distributing software that uses Matlab functions. Thus, the spring design optimization problem solved with the efficient Matlab function *fmincon* is implemented in Visual Basic to be compatible with existing IST software. Deployment and marketing of the resulting software is subject to no fee to the Matlab editor, and the software user does not even need to have a Matlab licence.

When building the software, it appears that *fmincon* is efficient but has a slow resolution process. In order to maintain acceptable interactivity of the software, it was decided not to implement the highest level of assistance. Indeed, displaying the design space requires 38 optimization problems to be solved. Thus, this functionality has not yet been implemented. However, progress in computer power suggests that this will be possible in the near future. At present, the tool allows the design space to be explored by selecting an appropriate objective function. Instead of viewing the entire design space, the designer can select a parameter and a bound (minimum or maximum) as an objective function to see what influence an input has on that bound. This provides a partial view of the design space. Even though it appears to be time consuming, sensitivity analysis has been implemented in order to enable the designer evaluate the robustness of the proposed design when necessary.

Figure 6 shows the final window interface of the software. A link has been added to transfer the optimum design to the analyzing tool previously developed by IST (see Fig. 1). Doing so, designers are able to see details of the design such as fatigue life diagrams or manufacturing tolerances.

5 Conclusion

In the present paper, a strategy was introduced to build an assistance tool to be used in preliminary design. The main idea was to provide a window interface as a specification sheet where each parameter can be defined by a lower and an upper bound in order to manage uncertainty.

The requirements are translated into an optimization problem that is solved automatically. As each data item is entered, a full overview of the design space can be provided showing the allowable bounds for each parameter. This can be obtained by setting each bound as the objective function of the optimization problem. Solving the optimization problem whatever the specifications is not an easy matter. Several solutions are proposed depending on the nature of the optimization problem. Details are given for solving an optimization problem with continuous variables and functions.

The methodology is then applied to compression spring design. The software mock-up uses the DOT algorithm from Vanderplaats. This algorithm appears to be extremely quick and efficient. Thus, the highest level of assistance was implemented. At each new entry from the designer, the tool immediately shows the design space and proposes an optimum design. This tool has been tested in an industrial context and proved to increase interactivity in design as this approach helps the designer build-up his specifications.

Once the software mock-up was validated by industrial designers, the corresponding industrial software was developed in collaboration with the Institute of Spring Technology. Because of distribution costs, the software uses the Matlab *fmincon* function to solve the optimization problem. This algorithm is quite slow so the highest level of assistance has not been implemented. However, designers can obtain a partial view of their design space by selecting the appropriate objective before a run. The compression spring optimization module has now been distributed by IST. Other modules for the optimal design of extension, conical and torsion springs are under development.

Acknowledgments The author would like to thank Schneider Electric Industries and especially A. Samperio for his support to this research project, the Institute of Spring Technology for the highly relevant work of J. Desforges, J. Slade and R. Dignall, and finally E. Rodriguez for his work related to his PhD Thesis.

References

1. Johnson, R.C.: Optimum Design of Mechanical Elements. Wiley, New York (1980)
2. Metwalli, S., Radwan, A., Elmeligy, A.A.: CAD and optimization of helical torsion springs. Comput. Eng. **2**, 767–773 (1994)
3. Deford, R.: Iterative logic for nested compression spring design. Int. Mag. Spring Des. Manuf. June, 27–28 (2003)
4. Tumkor, S.: Internet-based design catalogue for the shaft and bearing. Res. Eng. Des. **12**, 163–171 (2000)
5. Dudley, D.W.: When splines need stress control. Prod. Eng. NY **28**, 56–61 (1957)
6. Serna, L., Fischer, X., Bennis, F.: Cognitive virtual exploration for optimization model reduction. Int. J. Appl. Sci. Eng. Technol. **4**, 79–87 (2008)
7. Palesi, M., Givargis, T.: Multi-objective design space exploration using genetic algorithms. In: CODES'02, Estes Park (2002)
8. Yannou, B., Simpson, T.W., Barton, R.R.: Towards a conceptual design explorer using metamodeling approaches and constraint programming. In: Proceedings of the 2003 Design Engineering Technical Conference, Chicago (2003)
9. Rajagopal, S., Cavallaro, J.R., Rixner, S.: Design space exploration for real time embedded stream processors. IEEE Micro Special Issue Embed. Syst. **24**(4), 54–66 (2004)
10. Ekpanyapong, M., Lim S.K., Ballapuram, C. Lee, H.-H.S.: Wire-driven microarchitectural design space exploration. In: IEEE International Symposium on Circuits and Systems, ISCAS 2005, 23–26 May 2005, vol. 2, 1867–1870 (2005)
11. Kunzli, S.: Efficient design space exploration for embedded systems. Shaker Verlag, Aachen (2006)
12. Dore, R., Pailhes, J., Fischer, X., Nadeau, J.P.: Identification of design variables and criterion variables towards the integration of user requirements into preliminary design. Int. J. Prod. Dev. **4**(5), 508–529 (2007)
13. Wong, L.M., Wang, G.G.: Development of an automatic design and optimization system for industrial silencers. In: Proceedings of the 2002 Design Engineering Technical Conference, Montreal (2002)
14. Deb, K.: Multi-objective evolutionary optimization: past, present, and future evolutionnary design and manufacture, selected papers from ACDM'00. pp. 225–236. Springer (2000)
15. Vanderplats, G.N.: Numerical Optimization Techniques for Engineering Design. McGraw-Hill, New York (1984)
16. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
17. Back, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)
18. Haykin, S.: Neural Networks, A Comprehensive Fundation. Prentice Hall PTR, Upper Saddle River (1994)
19. Giraud-Moreau, L., Lafon, P.: A comparison of evolutionary algorithms for mechanical design components. Eng. Optim. **34**, 307–322 (2002)
20. Meckesheimer, M., Barton, R.R., Simpson, T., Limayem, F., Yannou, B.: Metamodeling of combined discrete/continuous responses. AIAA J. **39**(10), 1950–1959 (2001)
21. Moore, R.E.: Methods and Applications of Interval Analyzis. SIAM, Philadelphia (1979)
22. Hyvonen, E.: Constraint reasoning based on interval arithmetic: the tolerance propagation approach. Artif. Intell. **58**, 71–112 (1992)
23. Yao, Z., Johnson, A.L.: On estimating the feasible solution space of design. Comput. Aided Des. **29**, 649–655 (1997)
24. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. J. Mech. Des. **112**, 223–229 (1990)
25. Kannan, B.K., Kramer, S.N.: An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its application to mechanical design. J. Mech. Des. **116**, 405–411 (1994)
26. Deb, K., Goyal, M.: A flexible optimization procedure for mechanical component design based on genetic adaptative search. J. Mech. Des. **120**, 162–164 (1998)
27. Paredes, M., Sartor, M., Daidié, A.: Advanced assistance tool for optimal compression spring design. Eng. Comput. **21**, 140–150 (2005)
28. Paredes, M., Sartor, M., Fauroux, J.C.: Stock spring selection tool. Int. Mag. Spring Des. Manuf. **39**, 53–67 (2000)
29. Coleman, T.F., Li, Y.: An interior, trust region approach for nonlinear minimization subject to bounds. SIAM J. Optim. **6**, 418–445 (1996)
30. Powell, M.J.D.: A fast algorithm for nonlineary constrained optimization calculations, numerical analyzis, Lecture Notes in Mathematics, vol. 630 (1978)