

A Flexible Methodology for Optimal Helical Compression Spring Design.

Alistair Bentley¹, Tim Hodges², Jiahua Jiang³, Justin Krueger⁴, Saideep Nannapaneni⁵, Tianyu Qiu⁶

Problem Presenters: Jordan Massad⁷, Sean Webb⁸; Faculty Mentors: Ilse Ipsen⁹, Ralph Smith¹⁰

Abstract

The helical compression spring is one of the most commonly used mechanical components. As a result, spring design is a commonly performed task in industry and would benefit from the development of flexible tools that can answer the question of what makes a spring design optimal. To address this, we develop an intelligent supporting tool using an object-oriented design framework that provides users the robustness needed to define and solve any spring design problem they require. In addition to our tool interfacing with a graphic user interface, we supplement its flexibility with analytic tools. One such tool provides global sensitivity analysis, and another provides a feasibility plot which helps the user determine if their problem has a solution or if it should be redefined. We also include a model for stress relaxation, which is a novel inclusion in the list of considerations for spring specifications. Both its unique flexibility and capability and the analytic tools we provide to support it make our tool useful in defining, solving, and analyzing optimal spring design problems.

1 Introduction

1.1 Acceleration Switch Example

Springs have many everyday uses, such as in cars, homes, gyms, and industry. For example, springs are used in acceleration switches like the one seen in Figure 1. Their purpose in this instance is to close an electrical circuit once the object carrying the switch reaches a certain acceleration. This mechanism provides power to sensory equipment and data recorders for very precise amounts of time, and an application of its use is to collect data from rocket sled experiments [1].

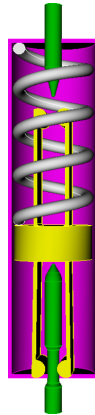


Figure 1: An example of an acceleration switch, see reference [1].

¹Mathematics, Clemson University

²Mathematics, Colorado State University

³Mathematics, University of Massachusetts Dartmouth

⁴Mathematics, Virginia Tech

⁵Civil & Environmental Engineering, Vanderbilt University

⁶Mathematics, University of Delaware

⁷Sandia National Laboratory

⁸Sandia National Laboratory

⁹Mathematics, North Carolina State University

¹⁰Mathematics, North Carolina State University

These rocket sled tests include very high velocities and accelerations, and the experiments end with their destruction upon impact. Since the test is not easily replicated, we must have the ability to consistently and correctly collect data. This means we require that the spring in the acceleration switch must function as expected to the forces exerted on it, and having a switch that is closed earlier or later than expected is undesirable [2]. Designing a spring to behave as desired is a complicated problem that requires satisfying numerous constraints on the spring’s physical properties. That the desired spring can vary depending on the application further complicates the problem.

Spring design under these conditions amounts to a constrained optimization problem where certain properties of the spring may need to be maximized or minimized subject to bounds on some or all of the spring’s other properties. Extending the scope beyond spring design for use in acceleration switches, which already cover a vast spread of specific spring designs, results in even more possibilities for design specifications. Rather than develop methods to solve these problems on an ad hoc basis, we develop to solve a wide variety of spring design problems.

1.2 Optimal Spring Design

Designing an optimal spring is not a new problem. Brake et al. considered the design of an acceleration switch with enabled uncertainty [2]. Alternatively, Sastry et al. implemented probabilistic response surface methodology to investigate the complications of uncertainty in designing a spring [3]. Others have considered the reduction of the optimization problem to focus on a single parameter [4]. Similar tools to the ones mentioned above also exist [5].

Motivated by a workflow for the design process, we present an intelligent tool for building an optimal spring. This tool is robust to a variety of spring design specifications and allows for optimization under uncertainty. We promote user interactivity by interfacing our tool with a graphical user interface and providing a tool to visualize the feasibility of a spring design. We also provide the ability to perform a global sensitivity analysis with respect to a spring’s design variables. Finally, we add stress relaxation as a optional specification for optimal spring design, which to our knowledge has never been included before.

In this manuscript we provide details for our novel and robust tool as follows. Section 2 formally defines the spring design problem for helical compression springs. Section 3 outlines the workflow for optimal spring design. Section 4 shares cases studies using our approach, and Section 5 summarizes our work and provides suggestions for future work.

2 The Problem

To develop the optimal spring design tool, we first need to identify and define the problem. This section explains the specific parameters and attributes relevant to optimal spring design, and then frames the optimization problem in a mathematical context.

2.1 Helical Compression Springs

Helical compression spring, such as the one in Figure 2, is the most recognizable type of spring and is the spring of choice for this work.

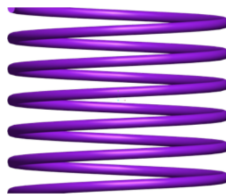


Figure 2: An example of a helical compression spring [1].

We can uniquely define a helical compression spring by a set of design parameters, which then determine physical attributes of the spring [1]. Figures 3a and 3b provide visual representations for a few of the basic parameters, and a complete list of parameters.



(a) The pitch, outer diameter, and length of a spring.

(b) The inner diameter of a spring.

Figure 3: A subset of basic parameters for a helical compression spring.

Spring Design Parameters

- Wire Diameter (d_w)** This is the diameter of the circular wire used to form the spring.
- Inner Diameter (d_i)** This is the diameter of the circle formed by the spring's interior excluding the wire. See Figure 3b.
- Outer Diameter (d_o)** This is the diameter of the circle formed by the spring including the wire, so $d_o = d_i + 2d_w$. See Figure 3a.
- Total Coils (N_t)** This is the total number of 360 degree rotations formed by the spring coil. The value does not have to be an integer.
- Active Coils (N_a)** This is the number of coils that contribute to the force opposing any compression force on the spring. The number of active coils depends on N_t and the spring's end conditions. If the end condition is *closed* the terminal coils of the spring are welded to their adjacent coils and $N_a = N_t - 2$. If the end condition is *open*, the terminal coils of the spring are free and $N_a = N_t - 1$. Note this relationship creates a lower bound on the total number of coils.
- Free Length (L_{free})** This is the spring's length without any force acting on it, i.e., the spring's natural length.
- Open Length (L_{open})** This is the spring's resting length in its application. In the acceleration switch, for example, it is the length of the spring in the switch when the switch is not in use.
- Hard Length (L_{hard})** This is the minimum length a spring can compress to under a specified force. For the acceleration switch example, the spring cannot compression beyond a certain point under the applied forces because additional compression would reopen the circuit.
- Solid Length (L_{solid})** This is the spring's length when its coils are completely compressed. The solid length is $L_{\text{solid}} = N_t d_w$. Also, the lengths defined must satisfy $L_{\text{solid}} \leq L_{\text{hard}} < L_{\text{open}} \leq L_{\text{free}}$.
- Deflection (s)** This is the amount the spring is compressed.
- Pitch (p)** This is the distance between coils. For coils with *closed* end conditions, $p = \frac{L_{\text{free}} - 2d_w}{N_a}$, and for coils with *open* end conditions, $p = \frac{L_{\text{free}}}{N_a - 1}$. See Figure 3a.
- Young's Modulus (E)** This describes the spring's response to uniaxial stress and is one defining aspect of the spring's rigidity. Young's modulus is also know as the modulus of elasticity.

- Poisson Ratio (ν)** This is a measurement of how much the spring deforms in any two dimensions when it is compressed in the third dimension.
- Shear Modulus (G)** This describes the spring's response to shear stress and is another defining aspect of the spring's rigidity. The shear modulus is also known as the modulus of rigidity and is $G = \frac{E}{2(1+\nu)}$.

This list of parameters is not exhaustive, but the parameter set given here defines many common spring attributes relevant to optimal spring design. We have a particular interest in the attributes listed below.

Spring Attributes

Spring Rate

$$k = \frac{G}{8N_a} \frac{d_w^4}{(d_i + d_w)^3} \quad (1)$$

The spring rate measures the rigidity or stiffness of the spring.

Spring Index

$$C = \frac{d_i}{d_w} + 1 \quad (2)$$

The spring index measures the ratio between the spring's inner diameter and wire diameter and often is proportional to the manufacturing cost for the spring [6].

Preload Force

$$F_{\text{open}} = (L_{\text{free}} - L_{\text{open}})k = (L_{\text{free}} - L_{\text{open}}) \frac{G}{8N_a} \frac{d_w^4}{(d_i + d_w)^3}$$

Preload Force is the force applied to the spring while it is in the open position. For the acceleration switch example, this is the force applied on the spring when the switch is not in use.

Coil Binding Gap

$$g = \frac{L_{\text{hard}} - L_{\text{solid}}}{N_t - 1} \quad (3)$$

Coil binding gap is the pitch of the spring when the spring is compressed to its hard length.

Buckling Slenderness Ratio

$$\lambda = \frac{L_{\text{free}}}{d_i + d_w}$$

The buckling slenderness ratio measures the spring's susceptibility to buckling under compression.

Maximum Shear Stress

$$\tau = \frac{G(L_{\text{free}} - L_{\text{hard}})}{4\pi N_a} \left[\frac{d_w(4d_i^2 + 9.46d_i d_w + 3d_w^2)}{d_i(d_i + d_w)^3} \right] \quad (4)$$

This expression approximates the maximum shear stress applied to the spring.

Diametral Expansion

$$d_{\text{expand}} = d_w + \sqrt{(d_i + d_w)^2 + \frac{p^2 - d_w}{\pi^2}} \quad (5)$$

The diametral expansion is the measurement of how much the outer diameter of the spring increases as the spring is compressed. The discriminant of the square root is necessarily nonnegative, so diametral expansion can put additional constraints on the relationship between some parameters. Note also that the formula for diametral expansion depends on the spring's end conditions. The equation above is the diametral expansion of a spring with *closed* end conditions, and the diametral expansion for a spring with *open* end conditions is the solution of a cubic polynomial too complicated to show here.

Stress Relaxation

$$\phi = \frac{2\pi N_a (d_i + d_w)^2}{G s d_w^4} \int_0^{d_w} r^2 \left(\left(\frac{2Gsr}{\pi N_a (d_i + d_w)^2} \right)^{-n} + \frac{c}{k} G n t^k \right)^{-\frac{1}{n}} dr \quad (6)$$

Stress relaxation measures how much the spring force decays in high temperature conditions under constant compression for time t . This model formula is derived using the Norton-Bailey law where c , n , and k are temperature dependent material specific constants. The inclusion of stress relaxation as a spring attribute in optimal spring design is unique to our tool construction. We include a detailed derivation of this formula in Section 6.

Considering this list of spring parameters and attributes, optimal spring design amounts to optimizing a collection of parameters and attributes subject to the restrictions of design specifications. We formulate this objective as a constrained optimization problem.

2.2 Constrained Optimization

Given an input of objectives and constraints on those objectives which are both defined by a set of design variables, we can formulate the challenge of designing an optimal spring as a constrained optimization problem. Without loss of generality, the formulation of such a problem is

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{G}(\mathbf{x}) \leq \mathbf{0}, \end{aligned} \quad (7)$$

Here, \mathbf{x} is the set of independent variables for the problem. In the context of spring design the independent variables are the spring parameters, which we also call design variables in the context of optimization. The function $F(\mathbf{x})$ is the objective function that we try to optimize. For spring design this function can consist of a single spring attribute or any weighted linear combination of attributes. Lastly, the function $\mathbf{G}(\mathbf{x})$ is the set of constraint functions restricting the optimization problem. For spring design, this set can consist of any combination of bounds on the design variables and bounds on the spring attributes.

To solidify this formulation, consider the acceleration switch example provided in the introduction. Suppose we wish to minimize the weighted sum of the spring rate (k) and the spring index (C) with respect to the inner diameter (d_i) and wire diameter (d_w) of the spring. Suppose we must satisfy a lower bound on the wire diameter ($d_{w,\min}$), an upper bound on the inner diameter ($d_{i,\max}$), and an additional constraint that the coil binding gap (g) is greater than the value g_{\min} . Composing this problem mathematically gives

$$\begin{aligned} \min_{d_i, d_w} \quad & w_1 k + w_2 c \\ \text{subject to} \quad & g + g_{\min} \leq 0, \\ & -d_w + d_{w,\min} \leq 0, \\ & d_i - d_{i,\max} \leq 0, \end{aligned} \quad (8)$$

or equivalently,

$$\begin{aligned} \min_{d_i, d_w} \quad & w_1 \left(\frac{G}{8N_a} \frac{d_w^4}{(d_i + d_w)^3} \right) + w_2 \left(\frac{d_i}{d_w} + 1 \right) \\ \text{subject to} \quad & -\frac{L_{\text{hard}} - L_{\text{solid}}}{N_t - 1} + g_{\min} \leq 0, \\ & -d_w + d_{w,\min} \leq 0, \\ & d_i - d_{i,\max} \leq 0, \end{aligned} \quad (9)$$

where w_1 , w_2 , c_{\max} , $d_{w,\min}$, $d_{i,\max}$ are known values.

3 The Approach

Our approach follows the optimal spring design workflow illustrated in Figure 4. This is an iterative process that receives a user specified problem, reformulates and solves the problem, and then returns a result for the user to accept or decline. The remainder of this section expands upon the details of each element in this diagram.

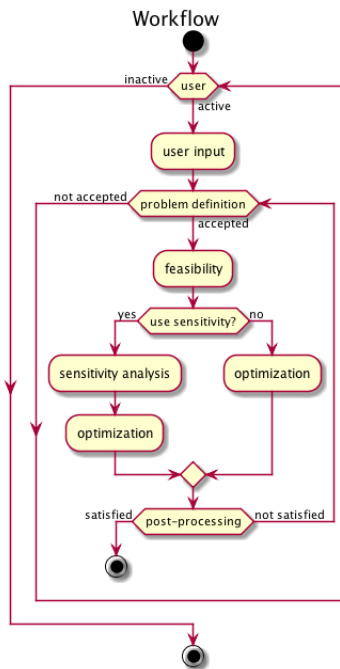


Figure 4: A workflow demonstrating the problem approach.

3.1 User Input

During the first step of our work flow, a spring designer (herein known as the user) presents an arbitrary spring design problem. For instance, the user might be interested in building a spring to use in the acceleration switch described in the introduction. Alternatively, the user might be interested in designing a spring for shock absorption on a car. During this initial phase, any spring used in an industrial application could be given as input.

3.2 Problem Definition

Generally, the user would like to optimize some of the spring attributes introduced in Section 2.1, while satisfying the problem’s physical design constraints. In the case of an acceleration switch example in Section 2.2, the user would like to achieve the smallest possible weighted sum of spring rate (k) and spring index (C), while satisfying the physical dimensions of the spring and the constraint on the coil binding gap g . Thus, the next step in our workflow is to take the user input and formulate an optimization problem that includes an objective, a set of design variables and constraints. For the acceleration switch, this results in the problem defined in (9).

In order to run the design analysis, we place some restrictions on the form of this optimization problem. First, the optimization’s objective function must be specified as a weighted sum of minimums and maximums. While other methods for multi-criteria optimization exist, including these in our scheme has been left to future work.

Table 1: Variables and attributes that define a spring design.

Permissible Design Variables	Permissible Attributes for Optimization
The spring's natural length (L_{free})	Spring Rate (k)
The spring's resting length (L_{open})	Spring Index (C)
Spring's minimum compression length (L_{hard})	Preloaded Force (F_{open})
Total spring coil (N_t)	Coil Binding Gap (g)
Spring's inner diameter (d_i)	Buckling Slenderness Ratio (λ)
Spring's width diameter (d_w)	Shear Stress (τ)
Shear modulus (G)	Diametral Expansion (d_{expand})
	Stress Relaxation (ϕ)
	Outer Diameter (d_o)

The second restriction, based on discussions with engineers experienced with spring design, is to reduce the set of design variables and optimization attributes to those in Table 1. The decision to limit the space of design variables and objective functions is not a limitation of this method. Rather, it represents the set of options currently available in our design library. If the user wishes to constrain or optimize an attribute that is not part of the existing library, the flexibility to add it is available.

An example of this flexibility is our team's integration of stress relaxation into the design process, which, to our knowledge, no other spring design system has. After our team found an acceptable mathematical formulation for stress relaxation as shown in 2.1, the function library was extended, allowing the user to incorporate relaxation into any problem formulation.

In addition to the design variable and spring attribute libraries, we employed an object-oriented approach to our software design to maximize the flexibility in our workflow. Object-oriented programming is an approach to software design that centers around classes, which are data containers capable of performing a predefined set of tasks. For example, our program has a *Constraint* class that includes the constraint's function, and a way to determine if the constraint is violated at a given point. For a few examples of classes, see Figures 5a and 5b.



(a) *Spring* class.

(b) *Objective* and *Constraint* classes.

Figure 5: Examples of classes.

Using classes allow flexibility because the program's algorithm are designed to operate on the classes. For example, when the user inputs a new problem, the program generates an object with a specific objective function and set of constraints, and proceeds to use this object for the remainder of the program run time. Thus, our software can automatically convert the user problem into objects need for the optimizer to run.

3.3 Feasibility

Once the user has set a problem, the next step is determining if a solution exists for the problem. This entails determining if there is a region in the space of design variables such that all the constraints defined in the problem are satisfied by points in this region. We call this region of interest the feasible region. Returning to the acceleration switch example, the feasible region is the region defined by the three constraints in 9, i.e., $-\frac{L_{\text{hard}}-L_{\text{solid}}}{N_t-1} + g_{\text{min}} \leq 0$, $-d_w + d_{w,\text{min}} \leq 0$, and $d_i - d_{i,\text{max}} \leq 0$. If the constraints are such that no feasible region exists, then the problem is poorly defined and the user should modify it. To assess feasibility, we use a Latin hypercube sampling method, which is designed to ensure sampling coverage over the entire domain. This approach is used because the user defined constraints are very general in nature, e.g., they can be linear or non-linear. Additionally, a graphical tool has been added to help the user analyze the problem's feasible region and the value of the objective function over the region. Using this tool, a plot with two or three state variables displays the problem's overall feasible region and the feasible region each individual constraint forms. This allows users to identify the nature of the constraints in their problem.

3.4 Optimization

3.4.1 Global Optimization

Without loss of generality, we can formulate a constrained design optimization problem as it is written in (7). Several optimization algorithms (both local and global) are available to solve the above optimization problem. DIRECT is one of the most popular global search optimization algorithm [7, 8] and is what we use this spring optimization problem.

The main idea of the DIRECT algorithm is that we normalize the domain to be a unit hypercube and evaluate the function at the center point c of this hypercube and at points $\{c + \delta, c, c - \delta\}$ where δ is the size of the hypercube. Then we choose the hypercube that contains the smallest function value and further divide it into smaller hypercubes until we reach the global optimum. Because DIRECT is a sampling optimization algorithm, it requires no additional information from the objective function, such as its gradient or Hessian, and since it is a global algorithm, DIRECT does not require an initial point to begin optimization. Finally, DIRECT is easy to implement in an automated software framework since it does not require the segregation of linear and non-linear constraints.

3.4.2 Optimization Under Uncertainty

It is also essential to account for the variability in the manufacturing process, i.e., tolerances, in the design of springs. We also refer to tolerance as error or uncertainty in the design variables. The tolerance for each of the spring parameters is nominally assumed to be equal to one percent of the value of the variable. Thus, each variable has a uniform distribution with unknown mean and variance, which dependent on the mean value. The optimization formulation after accounting for tolerances can now be written as

$$\begin{aligned} & \min_{\mathbf{x}} E(F(\mathbf{x}, d)) \\ & \text{subject to} \quad -Pr(g_i(\mathbf{x}, d) \leq 0) + p_t^i \leq 0, \\ & \quad \quad \quad -Pr(\mathbf{x} \geq lb_{\mathbf{x}}) + p_{lb} \leq 0, \\ & \quad \quad \quad -Pr(\mathbf{x} \leq ub_{\mathbf{x}}) + p_{ub} \leq 0 \end{aligned} \tag{10}$$

where \mathbf{x} and d represent the design variables with tolerances and non-design variables with tolerances, respectively. The first constraint represents the probabilistic inequality constraint and the other constraints represent the bounds for the design variables. Optimization with tolerance is a nested double loop process where we carry out optimization in the outer loop, and in each iteration of optimization, we carry out reliability analysis in the inner loop using Monte Carlo sampling to check the probabilistic constraints. After obtaining the optimum values of the means of the design variables, we obtain their corresponding probability distributions since it is assumed that the variables follow uniform distributions with variances dependent on the mean values.

Since the optimal design parameters are stochastic, the optimum value of the objective function, which is a function of the optimum design variables is also stochastic. We obtain the probability distribution of the

optimum value objective function from the probability distributions of the design variables through uncertainty propagation analysis using Monte Carlo sampling. One random realization of each of the design parameters when passed through the objective function provides one realization of the optimum value of the objective function. Thus, we obtain several realizations of design variables, which result in several realizations of the optimum value of the objective function. Using the realizations, we use the Gaussian kernel density approach to construct the probability distribution.

3.5 Sensitivity Analysis

As the dimension of design variable space increases, the computational expense of the optimization procedure increases. To reduce the computational expense, we try to shrink the design variable space by removing the variables that have very little influence on the objective function. Thus, we require a dimension reduction strategy. Dimension reduction approaches have been divided into two categories, the filter approach [9] and the wrapper approach [10]. In the filter approach, we rank the input variables according to a ranking criterion, and we select the most dominant variables by assuming a threshold influence value. In the wrapper approach, we select a subset of variables from the list of all possible subsets of the input variables that best estimate the output variable. We use sensitivity analysis, a filter approach, in this work for dimension reduction. Two types of sensitivity analysis, local sensitivity analysis and global sensitivity analysis, have been developed in the literature. The local sensitivity index of a variable measures the sensitivity of the model output when the variable is fixed at a single value whereas the global sensitivity (GSA) index measures the variation of model output when the variable is varied over its range [11]. We choose to use GSA as it considers the entire range instead of conditioning at a point in computing the sensitivity to the output. Note that the input variables represent the design variables and model represents the objective function. Consider an objective function, F with n design variables x_1, x_2, \dots, x_n given by

$$Y = F(x_1, x_2, \dots, x_n). \quad (11)$$

In GSA two types of indices can be calculated for each variable - first order index and total effects index. The first-order index (S_i^I) quantifies the uncertainty contribution of an input variable, without considering its interactions with other variables, to the output variable uncertainty. Similarly, the total effects index (S_i^T) quantifies the uncertainty contribution of an input variable by considering its interactions with all variables, to the output uncertainty. The expressions for the two sensitivity indices are given below as

$$S_i^I = \frac{V_{x_i}(E_{x_{-i}}(Y|x_i))}{V(Y)} \quad (12)$$

$$S_i^T = \frac{E_{x_{-i}}(V_{x_i}(Y|x_{-i}))}{V(Y)} \quad (13)$$

Given a design range (lower and upper bounds), a variable can be assumed to be uniformly distributed in the design range. For each variable, the first-order index is calculated and if it is less than an assumed threshold value, then that variable is assumed insensitive and removed from the optimization procedure. A nested double loop Monte Carlo sampling approach is adopted for computation of sensitivity indices. Note that sensitivity analysis requires a considerable computational expense and therefore may be carried out in high dimensional design optimization problems. In problems with lower number of design variables, the sensitivity analysis can be avoided and directly perform design optimization.

3.6 Post Processing

Once the optimization process and perhaps sensitivity analysis are complete, the user must decide how to move forward. If the results can satisfy the user, the process can move forward to the manufacturing of the spring. If the results are not acceptable, the user must either decide to move on from the problem or return to a previous step in the workflow and make modifications. Returning to a point in the workflow, could mean performing more analysis on the problem as defined and returning to the optimization step with slight modifications. It could also mean redefining the problem completely and starting the optimization process from the beginning.

4 Computational Experiments

To demonstrate the workflow for optimal spring design and the robustness of our tools, we include two case studies. We present enough information in each case study to replicate the studies if desired.

4.1 Case 1:

Objectives: Minimize spring rate and spring index.

Constraints: Inner diameter and outer diameter relation, relation on the inner, wire, and outer diameters, coil binding gap, buckling slenderness, and maximum shear stress.

Design Variables: d_i , d_w , and N_t

4.1.1 User Input

Table 2: Fixed parameters for Case 1.

Name	Value
G	77 GPa
UTS	.7GPa
g_{\min}	.5 mm
k_{\max}	20
C_{\max}	10
L_{free}	85.5 mm
L_{hard}	20 mm

4.1.2 Problem Definition

$$\begin{aligned}
 & \min_{d_i, d_w, N_t} 0.5 \times \left(\frac{k}{k_{\max}} \right) + 0.5 \times \left(\frac{C}{C_{\max}} \right) \\
 & \text{subject to} \\
 & \quad d_i + 2 \times d_w \leq 60 \times 10^{-3} \text{mm} \\
 & \quad g_{\min} - \frac{L_{\text{hard}} - N_t d_w}{N_t - 1} \leq 0 \\
 & \quad \frac{L_{\text{free}}}{d_i + d_w} - \pi \sqrt{\frac{2(2\nu + 1)}{\nu + 2}} \leq 0 \\
 & \quad \frac{G(L_{\text{free}} - L_{\text{hard}})}{4\pi(N_t - 2)} \left[\frac{d_w(4d_i^2 + 9.46d_i d_w + 3d_w^2)}{d_i(d_i + d_w)^3} \right] - UTS \leq 0 \\
 & \quad 20 \times 10^{-3} \text{mm} \leq d_i \leq 30 \times 10^{-3} \text{mm} \\
 & \quad 1 \times 10^{-3} \text{mm} \leq d_w \leq 5 \times 10^{-3} \text{mm} \\
 & \quad 9 \leq N_t \leq 17
 \end{aligned} \tag{14}$$

In this case study, \mathbf{x} is a 3×1 vector, where x_1 , x_2 and x_3 represents the inner diameter (d_i), wire diameter (d_w) and total number of coils N_t respectively.

4.1.3 Feasibility

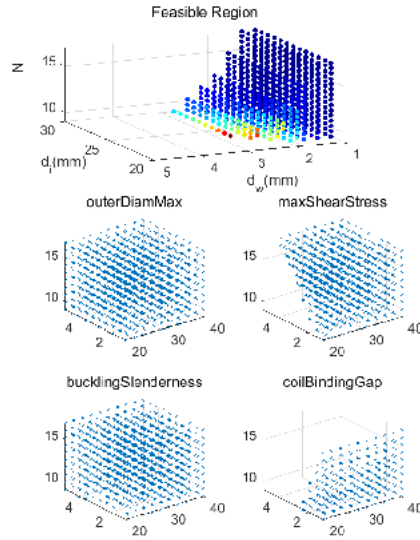


Figure 6: Feasibility region for case 1.

4.1.4 Sensitivity

Table 3: Sobol Indices for Case 1.

Variable	Sobol Index
d_i	0.16
d_w	0.61
N_t	0.02

From the sensitivity indices, it can be observed that all the design variables have significant influence on the objective function; therefore no dimension reduction is implemented for design optimization.

4.1.5 Optimization

As stated in Section 3.4, DIRECT algorithm is used for optimization and the optimum design point is obtained as

$$\mathbf{x}_{\text{opt}} = \begin{bmatrix} .03 \\ .001 \\ 17 \end{bmatrix}$$

Therefore, the optimum inner diameter(d_i), optimum wire diameter (d_w), and optimum number of coils (N_t) are $30mm$, $1mm$ and 17 , respectively.

4.2 Case 2:

Objectives: Minimize spring rate and spring index.

Constraints: Relation on inner, wire, and outer diameter, diametral expansion, coil binding gap, buckling slenderness, and maximum shear stress, stress relaxation.

Design Variables: d_i , d_w , and N_t

4.2.1 User Input

Table 4: Fixed parameters for Case 2.

Name	Value
G	77 GPa
UTS	.7GPa
g_{\min}	.5 mm
k_{\max}	20
C_{\max}	10
L_{free}	85.5 mm
L_{hard}	20 mm
Norton Bailey c	3.5×10^{-6}
Norton Bailey n	1.5
Norton Bailey k	1
Minimum Stress Relaxation	0.85
Deflection s	30 mm
Time Stress Relaxation t	3×10^6
Stress Relaxation G_{sr}	60 GPa

4.2.2 Problem Definition

$$\begin{aligned}
 & \min_{d_i, d_w, N_t} 0.5 \times \left(\frac{k}{k_{\max}} \right) + 0.5 \times \left(\frac{C}{C_{\max}} \right) \\
 & \text{subject to} \\
 & d_i + 2 \times d_w \leq 60 \times 10^{-3} \text{mm} \\
 & g_{\min} - \frac{L_{\text{hard}} - N_t d_w}{N_t - 1} \leq 0 \\
 & \frac{L_{\text{free}}}{d_i + d_w} - \pi \sqrt{\frac{2(2\nu + 1)}{\nu + 2}} \leq 0 \\
 & d_w + \sqrt{(d_i + d_w)^2 + \frac{p^2 - d_w}{\pi^2}} - d_o^{\max} < 0 \\
 & \frac{G(L_{\text{free}} - L_{\text{hard}})}{4\pi(N_t - 2)} \left[\frac{d_w(4d_i^2 + 9.46d_i d_w + 3d_w^2)}{d_i(d_i + d_w)^3} \right] - UTS \leq 0 \\
 & \frac{4}{G_{\text{sr}} \theta d_w^4} \int_0^{d_w} r^2 \left((G_{\text{sr}} \theta r)^{-n} + \frac{c}{k} G_{\text{sr}} n t^k \right)^{-\frac{1}{n}} dr \geq .85 \\
 & \text{where, } \theta = \frac{2s}{\pi N_a (d_i + d_w)^2} \\
 & 20 \times 10^{-3} \text{mm} \leq d_i \leq 30 \times 10^{-3} \text{mm} \\
 & 1 \times 10^{-3} \text{mm} \leq d_w \leq 5 \times 10^{-3} \text{mm} \\
 & 9 \leq N_t \leq 17
 \end{aligned} \tag{15}$$

Similar to the first case study, \mathbf{x} is a 3×1 vector, where x_1 , x_2 and x_3 represents the inner diameter (d_i), wire diameter (d_w) and total number of coils N_t , respectively.

4.2.3 Feasibility

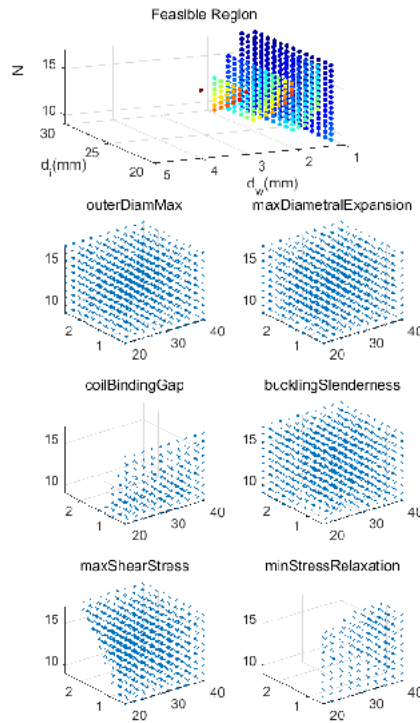


Figure 7: Feasibility region for case 1.

4.2.4 Sensitivity

Table 5: Sobol indices for Case 2.

Variable	Sobol Index
d_i	0.19
d_w	0.64
N_t	0.02

From the sensitivity indices, it can be observed that all the design variables have significant influence on the objective function; therefore no dimension reduction is implemented for design optimization.

4.2.5 Optimization

As stated in Section 3.4, DIRECT algorithm is used for optimization and the optimum design point is obtained as

$$\mathbf{x}_{\text{opt}} = \begin{bmatrix} .03 \\ .001 \\ 17 \end{bmatrix}$$

Therefore, the optimum inner diameter (d_i), optimum wire diameter (d_w), and optimum number of coils (N_t) are 30 mm, 1 mm and 17, respectively. Even though an additional constraint about stress relaxation is added, the optimum solution remained the same. Therefore, the assumed stress relaxation constraint does not influence the optimum solution.

5 Summary and Future Work

We developed an intelligent tool for the design of helical compression springs. The key highlights of this tool are: One, it allows for flexible optimization, which enables design of springs with interchanging objective functions and constraints. Two, it can incorporate stress relaxation in the design process, which to our knowledge has never been done. Three, it provides a feasibility design space given the constraints. Four, it can provide the global sensitivity indices of the design variables to the objective function.

Some limitations of the approach outlined are as follows. The choice of optimization and sensitivity analysis are fixed, however, they are modularized to allow a different optimization routine and sensitivity analysis to be incorporated. Given the amount of flexibility that is enabled, a user will have to be able to decide if a infeasible solution is due to user error.

More analysis of the stress relaxation and creep could result in better performance when using those conditions. Analysis on models of stress relaxation and their performance in our model would be beneficial for both our design, and could further understanding of stress relaxation and creep.

In this work, the DIRECT algorithm for global optimization is used for the design of springs. Also, computational experiments have been carried to perform design optimization considering tolerances in design variables but not incorporated in the design tool; this is an future work of importance. Other future work is to considering faster single-loop techniques, faster analytical and sampling techniques such as First Order Reliability Methods (FORM), and Importance sampling need to be considered

6 Appendix

6.1 Basics Regarding Stress Relaxation and Creep

A rocket mounted spring is expected to deliver normal mechanical output under extreme temperatures. The analysis of creep behavior helps us design a spring to meet the requirements under rough conditions.

Creep is the strain increase under constant stress, usually activated by high temperatures. Three mechanisms have been proposed to account for the phenomenon: dislocation slip and climb, grain boundary sliding and diffusional flow. We focus on the creep caused by the dislocation mechanism. The material's creep tendency is determined by a creep test. During a test, the spring is kept at $0.3 - 0.5T_m$ (T_m is the melting temperature of the material) and loaded by a constant tensile force.

Conventionally, creep can be divided into 3 stages. In the first stage (primary/ reduced/transient), the creep strain rate decreases to a certain value (minimum creep rate). The second stage (secondary/steady/stationary creep) is characterized by a nearly constant creep rate (minimum creep rate). In the third stage (tertiary creep), the creep strain rates increases rapidly and leads to rupture. The first stage is usually reversible with time after unloading, while the second/third ones are not. Since the primary creep occurs in a short duration and the tertiary one leads quickly to rupture, the secondary creep is under most serious consideration in many engineering designs.

Another common test to determine the material's susceptibility to high temperature is called stress relaxation. During the test the load is continuously decreased in such a way that the initial strain remains constant. The mechanism is more or less the same as that of creep.

6.2 Creep rate law of material

The starting point is the assumption that the creep rate may be described as a product of three separate functions of stress, temperature and time

$$\epsilon_{cr}(\sigma, T, t) = f_{\sigma}(\sigma)f_T(T)f_t(t)$$

The widely used functions of stress $f_{\sigma}(\sigma)$ are [12]:

$a\sigma^n$	Norton, 1929, Bailey, 1929
$b \left(\exp \frac{\sigma}{\sigma_0} - 1 \right)$	Soderberg, 1936
$a \sinh \frac{\sigma}{\sigma_0}$	Prandtl, 1928, Nadai, 1938, McVetty, 1943
$a_1\sigma^{n_1} + a_2\sigma^{n_2}$	Johnson et al., 1963
$a \sinh \left(\frac{\sigma}{\sigma_0} \right)^n$	Garofalo, 1965

where $a, b, a_1, a_2, \sigma_0, n_1, n_2$ are material constant. The dependence on the temperature $f_T(T)$ is usually expressed by the Arrhenius law

$$f_T(T) = \exp \left(\frac{-Q}{RT} \right)$$

where Q and R denote the activation energy and the Boltzmann's constant.

The time dependence part $f_t(t)$ is assumed to be [13].

t	secondary creep
bt^m	Bailey
$(1 + bt^{1/3}) \exp kt$	Andrade
$\sum_j a_j t^{m_j}$	Graham and Walles

For simplicity, we are going to only use the Norton-Bailey law and ignore any dependence on the temperature since the temperature is constant.

6.3 Stress Relaxation of Helical Springs

The following section is based on the paper [14]. Due to conservation of the total shear strain, the sum of the creep strain ϵ_{cr} 's rate of change and that of elastic shear strain ϵ_{el} is zero:

$$\dot{\epsilon}_{cr} + \dot{\epsilon}_{el} = 0$$

where the dot means differentiation in time. The elastic shear strain is related to the shear stress by shear modulus G : $\epsilon_{el} = \sigma/G$ and taking the derivative with time on both sides, we arrive at

$$\dot{\epsilon}_{el} = \dot{\sigma}/G$$

According to Norton-Bailey law(also known as time hardening law),

$$\dot{\epsilon}_{cr}(t) = c\sigma^{n+1}t^{k-1} \quad (16)$$

where c is the shear strain rate, n, k are temperature dependent material constants.

Substituting the above two equations to the conservation law,

$$\dot{\sigma}(t)/G + c\sigma(t)^{n+1}t^{k-1} = 0 \quad (17)$$

The initial condition is

$$\sigma(0) = G\theta r$$

where θ is the initial twist angle per unit length, r the radius of wire.

$$\sigma = \left((G\theta r)^{-n} + \frac{c}{k} G n t^k \right)^{-\frac{1}{n}}$$

The torque can be written as

$$M(t) = 2\pi \int_0^{d_w} r^2 \sigma(r, t) dr$$

$$M(0) = \frac{1}{2} G \pi \theta d_w^4$$

where d_w is the wire diameter.

We denote the spring force $P(t)$. Since the spring load is linearly related the torque $P(t) \propto M(t)$, given the constant deflection s ,

$$\frac{P(t)}{P(0)} = \frac{M(t)}{M(0)} = \frac{4}{G\theta d_w^4} \int_0^{d_w} r^2 \left((G\theta r)^{-n} + \frac{c}{k} Gnt^k \right)^{-\frac{1}{n}} dr$$

where

$$\theta = \frac{2s}{\pi N_a ((d_i + d_o)/2)^2}$$

The integral has no endpoint singularity and can be integrated to high order precision using the Gauss quadrature formula. The closer this quantity is to 1, the better the spring quality is. We notice a seemingly simpler hypergeometric function has been presented in the paper [15] which we find to yield unrealistic values due to divergence issues.

6.4 Creep of Helical Springs

This section is based on the paper [15]. The starting point is still the Norton-Bailey law (16). There is naturally another way to write the shear strain rate:

$$\dot{\epsilon}_{cr} = \dot{\theta} r = \frac{8\dot{s}r}{\pi N_a (d_i + d_o)^2}$$

We obtain σ from substituting the above equation into Norton-Bailey law. Given the constant spring force P^0 ,

$$P^0 \frac{d_i + d_o}{4} = M(0) = 2\pi \int_0^{d_w} r^2 \sigma(r, t) dr = \frac{\pi}{4} \frac{n+1}{4+3n} \left(\frac{8d_w^{4+3n} \dot{s}}{t^{k-1} (d_i + d_o)^2 \pi N_a c (d_i + d_o)^2} \right)^{\frac{1}{n+1}}$$

It can be deduced that the spring length s follows

$$s(t) = s(0) + \left(\frac{(d_i + d_o) P^0}{\pi} \frac{4+3n}{n+1} \right)^{n+1} \frac{\pi (d_i + d_o)^2 N_a c}{8k d_w^{4+3n}} t^k$$

where $s(0)$ is the initial spring length. The less the difference of the spring length $s(t) - s(0)$, the better the spring quality is.

References

- [1] J. Massad, “Flexible optimization and uncertainty-enabled design of helical compression springs in non-linear spring-mass-damper systems.,” 7 2015.
- [2] M. R. Brake, J. Massad, B. Beheshti, J. Davis, R. Smith, K. Chowdhary, and S. Wang, “Uncertainty enable design of an acceleration switch,” *Proceedings of ASME 2011 International Mechanical Engineering Congress and Exposition*, 2011.
- [3] A. M. N. P. Sastry, B. K. D. Devi, K. H. Reddy, K. M. Reddy, and V. S. Kumar, “Reliability based design optimization of helical compression spring using probabilistic response surface methodology,” *International Conference On Advances in Engineering*, 2012.
- [4] H. Zhao, G. Chen, and J. zhe Zhou, “The robust optimization design for cylindrical helical compression spring,” *Advanced Materials Research*, vol. 433-440, 2012.
- [5] M. Paredes, M. Sartor, and A. Daidie, “Advanced assistance tool for optimal compression spring design,” *Engineering with Computers*, 2005.
- [6] “Newcomb spring corp.” http://www.newcombspring.com/article_spring_index.html. Accessed: 2015-07-20.
- [7] D. E. Finkel, “Direct optimization algorithm user guide,” 2003.
- [8] M. Bjorkman and K. Holmstrom, “Global optimization using the direct algorithm in matlab,” *Advanced Modeling and Optimization*, 1999.
- [9] Y. Saeys, I. Inza, and P. Larranaga, “A review of feature selection techniques in bioinformatics.,” *Bioinformatics*, 2007.
- [10] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, 1996.
- [11] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*. John Wiley and Sons Ltd., first ed., 2008.
- [12] K. Naumenko, “Modeling of high-temperature creep for structural analysis applications,” *Professorial thesis, Martin Luther University Halle-Wittenberg, Germany*, 2006.
- [13] C. Chandenduang, *Boundary element analysis of time-dependent material non-linearity*. PhD thesis, University of Nottingham, 2000.
- [14] M. Siegel and D. Athans, “Relaxation of compression springs at high temperatures,” *Journal of Fluids Engineering*, vol. 92, no. 3, pp. 627–632, 1970.
- [15] V. Kobelev, “Relaxation and creep in twist and flexure,” *Multidiscipline Modeling in Materials and Structures*, vol. 10, no. 3, pp. 304–327, 2014.