Given a helical compression spring in a spring-mass-damper system, what are optimal springs?

Justin Krueger[1], Alistair Bentley[2], Tianyu Qiu[3], Saideep Nannapaneni[4], Jiahua Jiang[5], Tim Hodges[6]

Problem Presenters: Jordan Massad [7], Sean Webb [8], Faculty Mentors: Ilse Ipsen[9], Ralph Smith[10],

Abstract

1 Introduction

The use of mechanical switches in industry is a cornerstone of the modern world. Many mechanical switches are designed to use a spring. Depending on the use of the switch, the spring can be optimized for the function of the spring.

For example, consider an acceleration switch in figure 1.

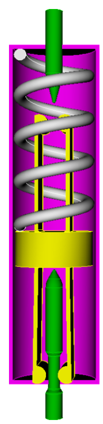


Figure 1: An example of an acceleration switch

Describe problem, approach, and summarize contribution and results. This switch is used in high acceleration testing. The use is to only record information at crucial times of the test. For this reason when a certain force is exerted on the switch the pins at both ends will connect. When this happens we have a circuit that will enable data collection. It is important that the switch does not open too soon or too late otherwise the data collection will be too large or too little.This is just one example of a switch that we must know the best spring to make the performance maximal. For a more in depth look at this example look at [1]

Given that a switch may be used in a myriad of ways, it is important that one can find an optimal spring, after the use of the spring is known. This leads to an optimization problem. Even more this leads to an optimization problem that requires flexibility to allow how a spring is determined to be optimal. In other words, the objective function and constraints must be interchangeable. In addition there are fabrication constraints that exist during manufacturing to consider.

[1]Mathematics, Virginia Tech University
[2]Mathematics, Clemson University
[3]Mathematics, University of Delaware
[4]Civil & Environmental Engineering,Vanderbilt University
[5]Mathematics, University of Massachusetts Dartmouth
[6]Mathematics, Colorado State University
[7]Sandia National Laboratory
[8]Sandia National Laboratory
[9]North Carolina State University
[10]North Carolina State University

Given any constraint or objective function it will depend on a set of variables. In optimization we only wish to concern ourselves with the state variables that are allowed to change during optimization. For this reason, we construct a spring object that will hold all attributes of a spring and we can pull the relevant variables when it is determined what they are.

In addition to this spring object, to reduce the uncertainty of the problem we conduct a sensitivity analysis of the objective function subject to the constraints and variables given. Allowing the user to reduce the state variables if it is determined that for a set of variables given there exist insensitive state variables.

A main contribution of this work is the framework to add constraints and objectives that may be unknown at the point of this paper. In addition a model of stress relaxation and uni-axial creep has been incorporated. If the reader does not approve of our stress relaxation and creep models, it is easy for the reader to incorporate other models into possible constraints and objectives. Overall the contribution of this paper is a flexible framework for which multiple new attributes of springs can be analyzed. **ADD!**

review history and existing literature Designing an optimal spring is not a new problem. In 2010 at the SAMSI IMSM workshop a team considered the design of an acceleration switch with enabled uncertainty. In fact, the switch they considered was in figure 1. This approach lead to a paper [1]. This is not the only approach to uncertainty, in [2] probabilistic response surface methodology was implemented to investigate the complications of uncertainty in designing a spring. In addition to these types of approaches it is possible to narrow focus to a single parameter, in [3] they focus on the spring stiffness. In [4] they attempt to design an optimal spring, with the introduction of a sizing tool. All references listed are informed to some extent by [5].

Our approach is not limited to an acceleration switch, instead it is but a special case of our approach. We do not focus on a set of parameters or attempt to fix the uncertainty of the spring design problem initially. We attempt to allow any possible construction of a spring optimization possible.

This does not rule out if a construction is not physically or mathematically invalid. We will dispose of that case when we are unable to find a feasible point for optimization within our feasibility check. In summary, we need to allow any possible construction of a spring design problem, and adjust accordingly.

outline In this paper we will give a description of the springs of interest, helical compression springs. Next, the formulation of the problem, and the approach taken to the problem. Following will be a section on workflow with descriptions of each step in the workflow. Lastly, we will discuss a few case studies given the framework constructed and a summary with future work that can be worked on.

2 Helical Compression Springs

Helical springs are the typical spring that comes to most peoples mind, for illustrative purposes, see figure 2.

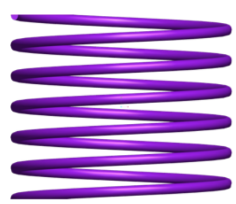


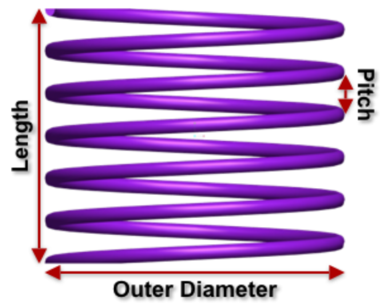
Figure 2: An example of a helical compression spring.

The above spring has many design parameters. These range from physical parameters such as wire diameter, to purely empirical attributes, for example spring rate. The empirical attributes have to be assumed in a range of values. This is because for extreme values these attributes exhibit behavior that is not under consideration. Below is a list of a spring's key design parameters and empirical attributes. We have added a few illustrations to illustrate some of the parameters.

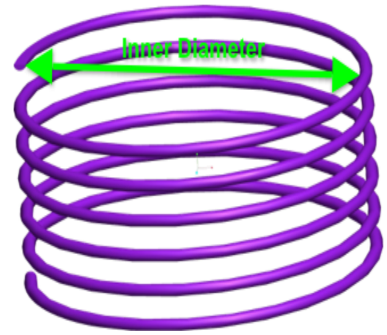
1. Spring's inner diameter d_i, illustrated in figure 3b.

2. Spring's outer diameter d_o illustrated in figure 3a.

3. Spring's wire diameter d_w.



(a) Pitch, outer diameter, and length of a spring.



(b) Inner diameter

Figure 3: A few illustrations of parameters for a helical compression spring.

4. Total number of spring coils N_t.
5. Active number of spring coils N_a, active coils are not touching any other coils, and is subject to the spring being closed or open.
6. Pitch p illustrated in figure 3a.
7. Spring's free length L_{free}, the spring's length without any force applied.
8. Spring's solid length L_{solid}, the spring's length when all coils are compressed together.
9. Spring's open length L_{open}, the spring's length at open position, open is beginning state.
10. Spring's open length L_{close}, spring length at close position, close is the ending state.
11. Spring's open length L_{hard}, the maximum a spring can compress for the application.
12. Spring's open force F_{open}, this is the force on the spring in open position.
13. Spring's shear modulus G, this is determined by the material of the spring.
14. Spring's youngs modulus E, this is determined by the material of the spring.
15. Spring's poisson ratio ν, this is determined by the material of the spring.
16. Ultimate torsional stress, UTS.

One can see that the above parameters are physical design parameters. Each of these are subject to design tolerances. Below is a set of empirical attributes that within a range of values exhibit typical behavior for a spring. We will assume that these attributes are within that range.

Spring Rate:

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

Spring Index:

$$C = \frac{d_i}{d_w} + 1. \tag{2}$$

Coil Binding Gap

$$g = \frac{L_{hard} - L_{solid}(d_w, N_a; ec)}{N_t - 1} \tag{3}$$

Max Shear Stress

$$\frac{G(L_{free} - L_{hard})}{4\pi N_a(ec)} \left[\frac{d_w(4d_i^2 + 9.46 d_i d_w + 3d_w^2)}{d_i(d_i + d_w)^3} \right] < UTS \tag{4}$$

Diametral Expansion

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

The reader may notice that

3 The Problem

Add dependency graph.

Design an algorithm that optimizes springs with interchangeable objectives and constraints. In addition, attempt to incorporate properties stress relaxation and creep into the available objectives and constraints.

A list of possible constraints/objectives given in minimization form is below

1. $d_i < d_i^{max}$
2. $d_i < d_o$
3. $d_i + 2d_w < d_o^{max}$
4. $d_{expand} - d_0^{max} < 0$
5. $\frac{G}{8N_a}\frac{d_w^4}{(d_i+d_w)^3} - k_{max} \leq 0$
6. $\frac{d_i}{d_w} + 1 - C_{max} < 0$
7. $(L_{free} - L_{open})\frac{G}{8N_a}\frac{d_w^4}{(d_i+d_w)^3} - F_{open} = 0$
8. $\frac{L_{hard}-L_{solid}}{N_t-1} + g_{min} \leq 0$
9. $\frac{L_{free}}{d_i+d_w} - \pi\sqrt{\frac{2(2\nu+1)}{\nu+2}}$
10. $-UTS + \frac{G(L_{free}-L_{hard})}{4\pi N_a(ec)}\left[\frac{d_w(4d_i^2+9.46d_i d_w+3d_w^2)}{d_i(d_i+d_w)^3}\right] < 0$

This list is not exhaustive as we want a user to be able to define new constraints and objectives. In addition to these we should be able to minimize any parameter of the spring. The incorporation of stress relaxation and creep have been left as computational experiments. If the reader wishes to read about these please check out section **Stress relaxation and creep section**.

In order to simplify the problem we have made some assumptions. First, we have assumed that constraints and objectives are all in terms of the number of total coils. This is to reduce the complexity of knowing if a constraint is dependent on the active number or total number of coils. These two numbers are dependent on the end conditions of the spring, whether it is open or closed at the end.

For this reason we also will consider all springs to be have a closed end condition. With these assumptions we simplify our constraints for pitch, p, solid length, L_{solid}, and the diametral expansion, d_{expand}.

JUSTIFY ASSUMPTIONS These assumptions have reduced the set of design parameters and have allowed us to simplify the problem. Without these assumptions we would need to deduce at run time what formulas we can use and this is a challenging problem from an implementation stand point. Also, we have a very complicated formula for diametral expansion if the end conditions are not closed. This presents issues of solving a cubic without the ability of discerning if the roots will be physically meaningful.

Describe data, how collected, their properties (removed noise, filled in missing data, applied normalizations).

4 The Approach

Present and justify approach for solving the problem Explain advantages of approach over existing ones Tell a story To be as flexible as possible we must be able to handle constraints and objectives that have an unknown number of variables a priori. To handle this uncertainty we instead of evaluating an objective or constraint with a set of values we evaluate with a "spring object" in the programming sense [6]. This spring has all the attributes of interest and the function merely picks what values it needs for evaluation.

This approach is also implemented in the design of a constraint and objective. An objective is an expression that we must be able to evaluate. A constraint is an expression that we must evaluate and check if this evaluationa violates the constraint condition. We use the idea of inheritance and let a constraint inherit the properties of being an objective with additional framework to compare to a condition. This allows the user to interchange constraints and objectives seamlessly.

Along with the flexible software design, we have implemented the ability to run feasibility and sensitivity checks before trying to optimize. This will allow the user to know additional information about the problem. First, from feasibility, the user will know quickly if the run is unable to find a feasible point to run optimization. Second, from sensitivity, the user can know which parameters will dramatically change the optimization versus which parameters are unnecessarily adding to the dimension of the optimization space.

5 Workflow

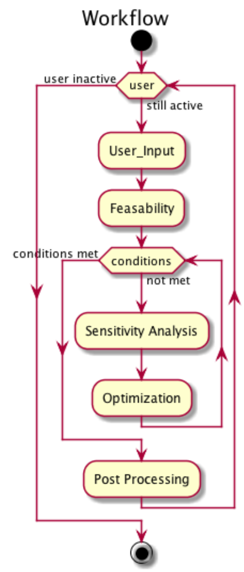


Figure 4: Illustration of the flow of our approach.

Above is an illustration of the workflow that is our approach. This section will go in depth into the feasibility, sensitivity analysis and optimization of our approach.

5.1 Feasibility

5.2 Sensitivity Analysis

As the dimension of design variable space increases, the computational expense of the optimization procedure increases. To reduce the computational expense, it is often desirable to reduce the design variable space by removing the variables that have very little influence on the objective function. Thus, a dimension reduction strategy is required to reduce the design variable space. Dimension reduction approaches, in the literature have been divided into two categories - (1)filter approach, and (2)wrapper approach. In the filter approach, the input variables are ranked according to a ranking criterion and the most dominant variables can be selected by assuming a threshold influence value. In the wrapper approach, a subset of variables is selected from the list of all possible subsets of the input variables that best estimate the output variable. An optimization technique is used to obtain the best subset of input variables. In this work, the variance-based Global Sensitivity Analysis (GSA), a filter approach, is used for dimension reduction. Note that the input variables represent the design variables for the objective function.

Consider a objective function, G, with n design variables given by x_1, x_2, ... x_n, given by

$$Y = G(x_1, x_2, ...x_n)$$

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

$$S_i^I = \frac{V_{X_i}(E_{X_{-i}}(Y|X_i))}{V(Y)}$$

$$S_i^T = \frac{E_{X_{-i}}(V_{X_i}(Y|X_{-i}))}{V(Y)}$$

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 total-effects 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. Thus, dimension reduction is implemented for a faster design optimization.[7] [8] [9] [10]

5.3 Optimization

Given a constrained optimization problem, we usually have many ways to solve it, eg. Trust Region Reflective Algorithm, Direct optimization algorithm, Conjugate Gradient Algorithm, Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm,etc. For difficult global optimization problem with bound constraints, Direct optimization algorithm is one of the most simple and effective methods. It requires no knowledge of the Lipschitz constant or even the Lipschitz continuous of the objective function. That is, it can be implemented in optimization problem with more difficult objective function. Given that our objective function is subject to change, it is best for us to use a global optimization technique.

[11] [12] [13] [14]

6 Computational Experiments

6.1 Case 1:

Objectives: Spring rate and spring index.
Constraints: Relation on inner, wire, and outer diameter, coil binding gap, buckling slenderness, and maximum shear stress.
State Variables: d_i, d_w, and N_t

6.2 Case 2:

Objectives: 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.
State Variables: d_i, d_w, and N_t

6.3 Case 3:

Objectives: F_{open}
Constraints: Relation on inner, wire, and outer diameter, coil binding gap, buckling slenderness, and maximum shear stress.
State Variables: d_i, d_w, and N_t

6.4 Case 4:

Objectives: Stress Relaxation
Constraints: Relation on inner, wire, and outer diameter, diametral expansion, and coil binding gap.
State Variables: d_i, d_w, and N_t

7 Summary and Future Work

The ability to interchange constraints and objective functions with any number of design variables allows the user the utmost flexibility. With the addition of feasibility and sensitivity analysis it is possible for any configuration of objective function, constraints, and design variables to be analyzed for refinement. The quantification of stress relaxation and creep allow the user a chance to incorporate these properties into any configuration, especially those that have never been tested.

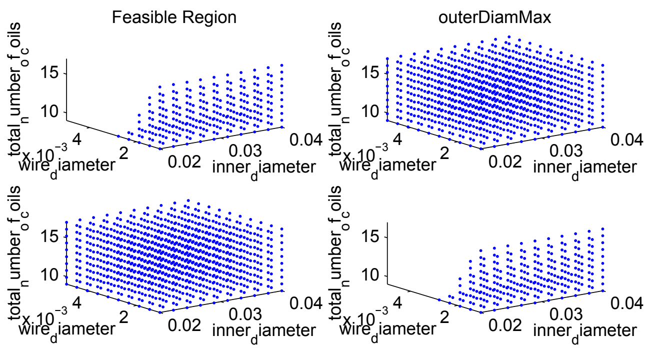


Figure 5: Example of feasible region.

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 ported in. 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.

Future work is bountiful for this approach. More analysis of the stress relaxation and creep could result in better performance. An in depth analysis of different models of stress relaxation and their performance in our model would be beneficial. The flexibility allows the inclusion of many different models of stress relaxation and creep to be added.

Senstivity for optimization

Uncertainty Quantification incorporation into the model.

Analysis of stress relaxation and creep models.

Basics about uni-axial creep and stress relaxation The spring is heated to $0.3 - 0.5T_m$ (T_m is the melting temperature of the material) and loaded by a tensile force. The induced normal stress σ is much less than the yield limit of the material σ_y. The load and temperature are constant throughout the test. The strain ϵ^{cr} will slowly increase.

Conventionally, creep can be divded into 3 stages. In the first stage(primary/reduced/trasient) ,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 design in many engineering applications.

Stress relaxation is the phenomenon that the stress decreases when the strain is held constant in time. During the test the load is continuously decreased in such a way that the initial strain remains constant.

7.1 Creep rate law of material

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

$$\epsilon^{cr} = f_\sigma(\sigma) f_T(T) f_t(t)$$

The widely used functions of stress $f_\sigma(\sigma)$ are:

$a\sigma^n$	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 constants that could depend on time t. 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

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.

7.2 Stress relaxation of helical spring

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:

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

The elastic shear strain is related to the shear stress by shear modulus G:

$$\epsilon_{el} = \sigma/G$$

and therefore

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

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

$$\dot{\epsilon}_{cr}(t) = c\sigma^{n+1}t^{k-1} \tag{6}$$

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

Substituting the above two equations to the conservation law,

$$\dot{\sigma}(t)/G + c\sigma(t)^{n+1}t^{k-1} = 0 \tag{7}$$

The initial condition is

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

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

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

The torque can be written as

$$M(t) = 2\pi \int_0^{d_w} r^2 \sigma(r, t)\,\mathrm{d}r = {}_2F_1\left(\frac{4}{n}, \frac{1}{n}; \frac{4+n}{n}; \frac{c\theta^n G^{n+1} n t^k}{k} \frac{d_w^n}{2^n}\right) M(0)$$

where d_w is the wire diameter.

Since the spring load is linearly related the torque $P_z(t) \propto M(t)$, given the constant deflection s,

$$\frac{P_z(t)}{P_z(0)} = {}_2F_1\left(\frac{4}{n}, \frac{1}{n}; \frac{4+n}{n}; \frac{c\theta^n G^{n+1} n t^k}{k} \frac{d_w^n}{2^n}\right)$$

where

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

The closer this quantity is to 1, the better the spring quality is.

7.3 Creep of helical spring

The starting point is still the Norton-Bailey law (). 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 substiting the above equation into Norton-Bailey law. Given the constant spring force P_z^0,

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

It can be deduced that the spring length s follows

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

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] 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.

[2] 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 probabilisitic response surface methodology," *International Conference On Advances in Engineering*, 2012.

[3] H. Zhao, G. Chen, and J. zhe Zhou, "The robust optimization design for cylindrical helical compression spring," *Advanced Materials Research*, vol. 433-440, 2012.

[4] M. Paredes, M. Sartor, and A. Daidie, "Advanced assistance tool for optimal compression spring design," *Engineering with Computers*, 2005.

[5] A. M. Wahl, *Mechanical Springs*. Penton Publishing, first ed., 1944.

[6] P. Coad and J. Nicola, *Object Oriented Programming*. P T R Prentice Hall, first ed., 1993.

[7] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics.," *Bioinformatics*, 2007.

[8] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, 1996.

[9] B. Liang and S. Mahdevan, "Error and uncertainty quantification and sensitivity analysis in mechanics computational models," *International Journal for Uncertainty Quantification*, 2003.

[10] 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.

[11] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *Springer Science+Business Media*, 2012.

[12] M. Bjorkman and K. Holmstrom, "Global optimization using the direct algorithm in matlab," *Advanced Modeling and Optimization*, 1999.

[13] MATLAB, *version 8.3.0.532 (R2014a)*. Natick, Massachusetts: The MathWorks Inc., 2014.

[14] D. E. Finkel, "Direct optimization algorithm user guide," 2003.