Exercise 6 + 7

Engineering Optimization – Concepts and Applications

June 2, 2020

by

Pieter Bijl - 4492102
&
Toon Stolk - 4488458

Delft University of Technology

Exercise 6

Exercise 6.1

In order to visualise the optimization problem with the constraints a matlab plot was made, which can be seen in Figure 1:

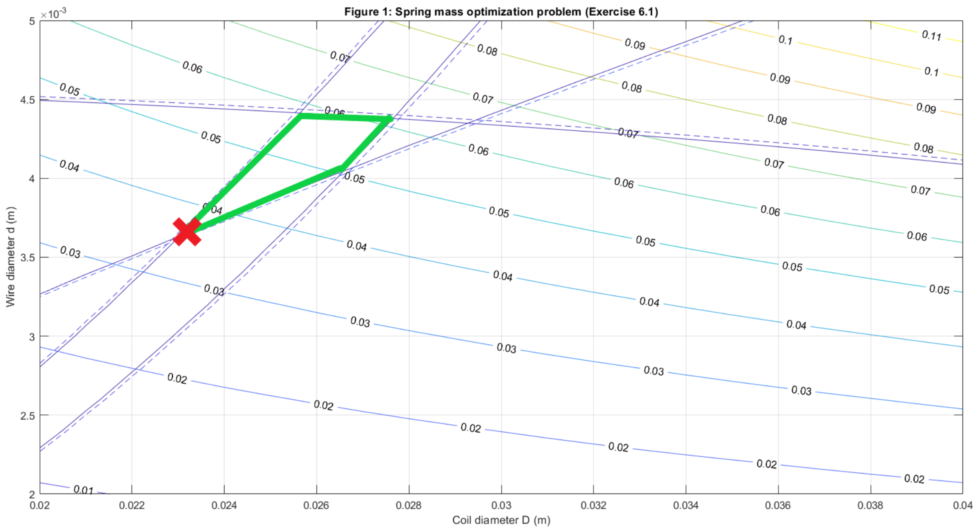

Figure 1: Overview of the objective function, the constraints, feasible domain and the expected optimum.

In this plot the objective function can be seen as contour lines where the values of the function have been included. The constraints have been visualised using pink lines where the dashed line signifies the side where the design is infeasible according to the constraint. The area surrounded by the green line is the feasible domain of the design and the red cross is the expected optimum lying around x = [0.0230 0.0037] and with a value between 0.03 and 0.04 for the objective function.

Exercise 6.2

The method of Sequential Quadratic Programming (SQP) was used in matlab by using the function fmincon(). This function computes the gradient of the functions using the forward finite difference method and then finds the optimum given an initial position and a function for the non-linear constraints.

The function springcon3 was modified to allow for an empty vector of equality constraints in order to perform the optimisation using the fmincon algorithm. As input value x0 = [0.034 0.0045] was used and this resulted in an optimum value for x of [0.0230 0.0036]. This matches closely with the result of the visual optimisation. The plotted result can be seen below:

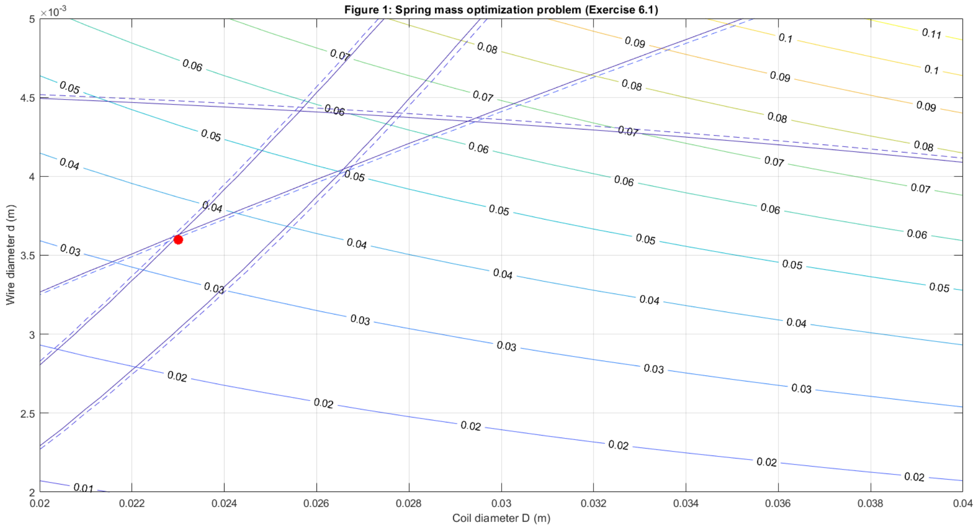

Figure 2: Optimal design point using the fmincon function (in red).

The next step was to evaluate the output parameters [x, fval, exitflag, output, lambda]. x has already been given and discussed. The value for the objective function (fval) was equal to 0.0354, which also makes sense when looking at the figure. The exitflag of the algorithm was equal to 1, which means that the stepsize of the function was below the tolerated level and thus the algorithm would stop, having reached a minimum. Output shows the structure of the optimisation and values such as the number of iterations and the final step size. This algorithm converged in 19 iterations and with a final stepsize of 3.6708e-09. Lambda are the lagrange multipliers with the values:

$$\bar{\mu} = \begin{bmatrix} 4.7255e-07 \\ 0.0191 \\ 0.0191 \\ 0.0680 \\ 4.0933e-07 \end{bmatrix} \tag{1}$$

Showing that the multipliers are 0 for the first and fifth constraints and active for the other three. The values for the second and third are equal, since they are dependant on each other. It is important to notice that the values of the active constraints are positive and the values of the inactive constraints are practically zero. This means that the optimum complies with the KKH condition. The last interesting thing is to check the value of the constraints at the optimum:

Table 1: Overview of the values of the constraints at the point x = [0.0230 0.0036].

Constraint	g(1)	g(2)	g(3)	g(4)	g(5)
Value	-0.1693	-4.1860e-06	-4.1860e-06	-1.1762e-06	-0.1954

It can be seen that there are three active constraints and two inactive constraints. The last analysis step that was performed was choosing different starting points, these were all over the possible values for x and as long as the constraints on the values of x were met, the optimum would not change. Outside of the allowable values for x the initial points were not evaluated since the functions might not be valid anymore.

Exercise 7

Exercise 7.1

The feasible region changes due to the linearisation of the constraints. The difference between the movement of the linearised constraints is explained by the different points around which the constraints are linearised. Therefore, some linearised constraints are moving in the feasible domain, some are moving out of the feasible domain and some are even quite accurate. The linearised constraints are depicted in the figure below:

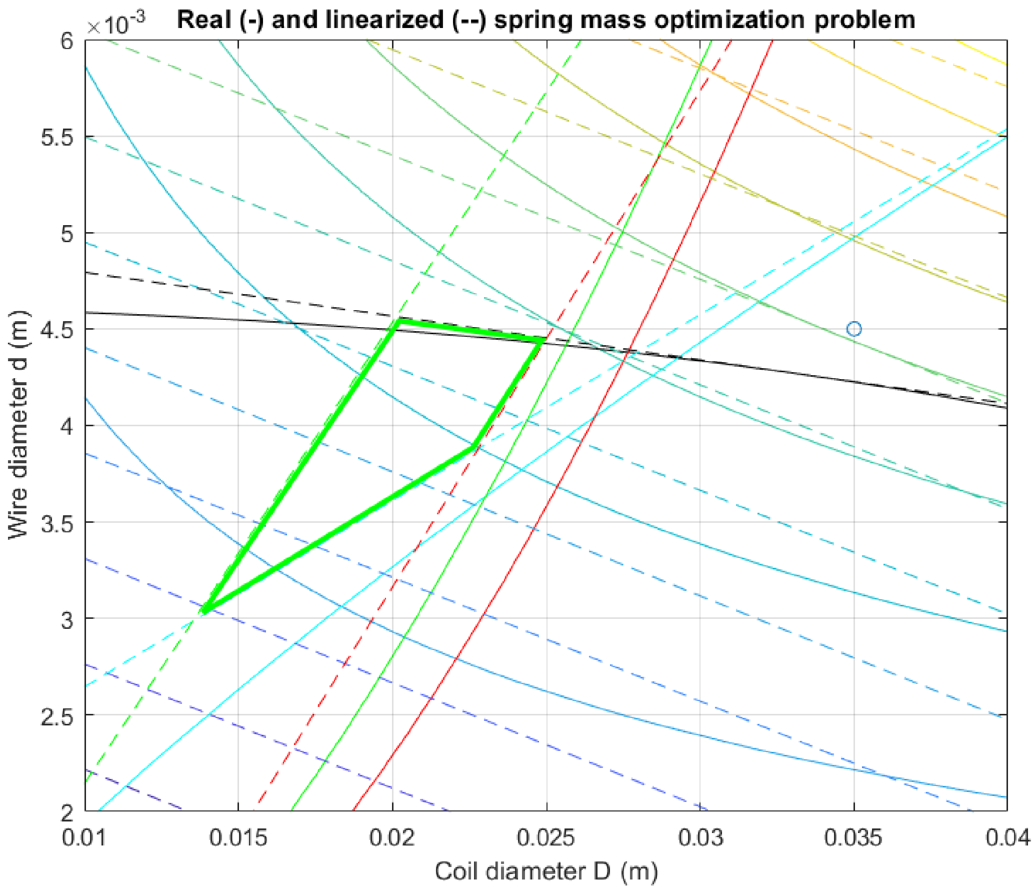

Figure 3: Linearised constraints, starting point

An algorithm that changes the point around which the constraints are linearised only, takes 5 cycles for the value to converge to an optimum. The converged value is: $[0.02318, 0.003659]$. The steps are visualised in the figures below.

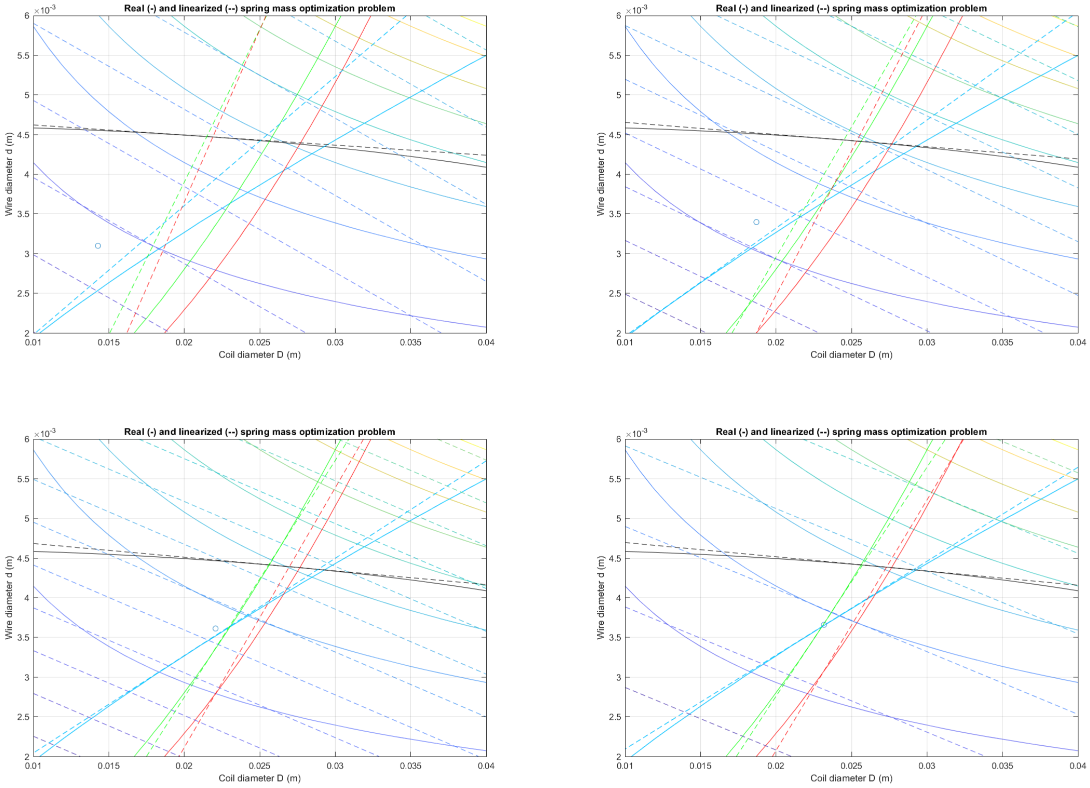

Figure 4: Cycles of the algorithm to find the optimal value, the steps are depicted with circles.

Exercise 7.2

Using sequential linear programming, the optimal value subjected to the linearised constraints can be found iteratively. This only takes 5 steps to reach a step difference of 10^-5, which was the selected value to stop the optimization. The steps are depicted in the figure below:

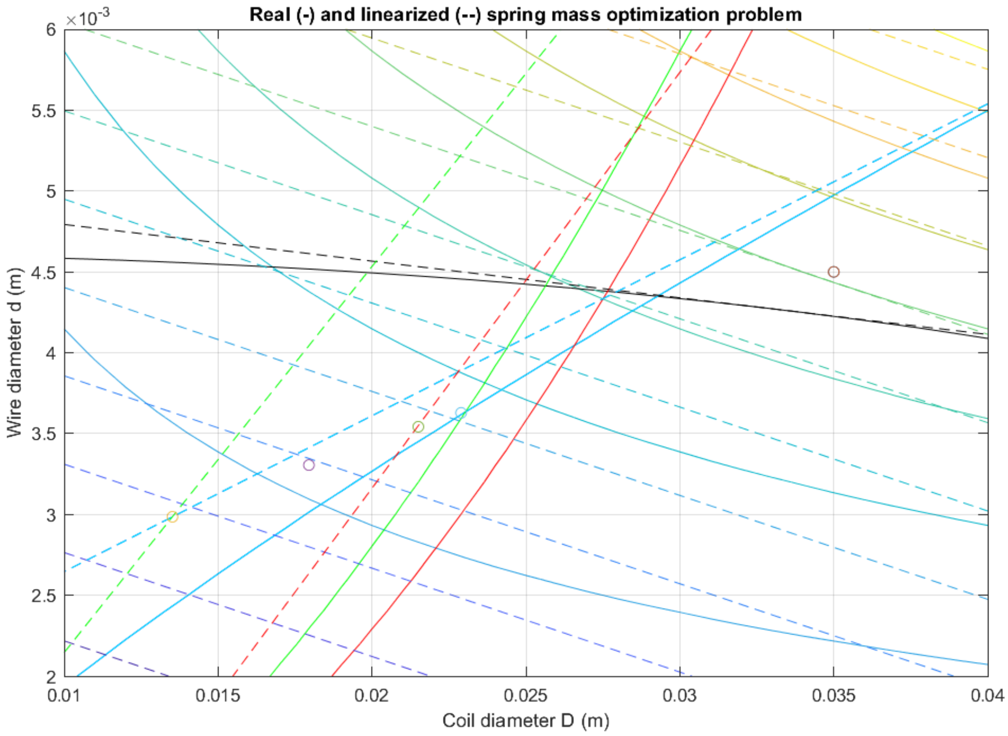

Figure 5: Iteratively found optimal value using sequential linear programming, the steps are depicted with circles and follow the feasible domains depicted in Figure 4.

Exercise 7.3

The steps in sequential linear programming can be constrained to allow for smoother transition to the optimum. For an initial guess of $x0 = [0.0300.0045]$ and a move limit of $ml = [0.0030.0003]$. The algorithm converges in 4 steps. The steps are depicted in the figure below.

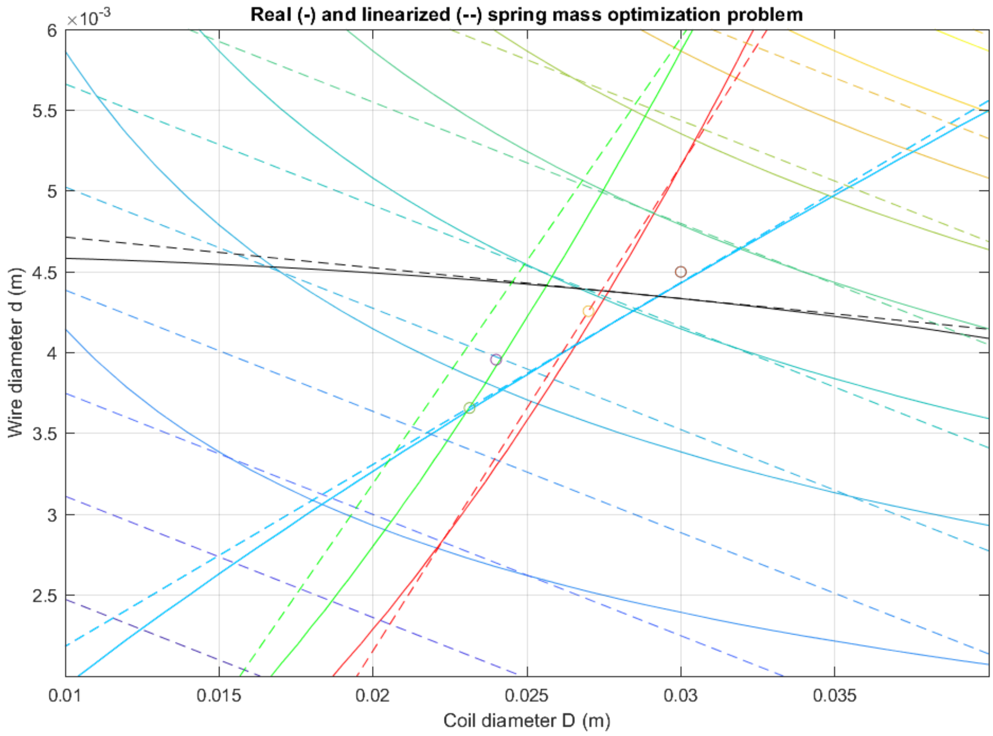

Figure 6: Iteratively found optimal value using sequential linear programming and a maximum step size of 0.003 and 0.0003 for the design variables respectively. The steps are depicted with circles and the feasible domain is based on the linearisation around point [0.030 0.0045].

For a starting point of $x0 = [0.0350.0045]$ with the same move limit as before, the algorithm does not find an optimum displaying an error message: `Linprog stopped because no point satisfies the constraints`. This can be remedied with a relaxation of the constraints, using a variable β. The variable is changed by the function linprog to relax the constraints to find an optimal value. If the value is within the constraints, the variable β will become 0. Using this method the starting point $x0 = [0.0350.0045]$ also converged to the optimal value in 7 steps, which is to be expected due to the 'distance' travelled with smaller step sizes compared to 7.1 and 7.2.