Optimization
Case study using Matlab optimization toolbox

October 22, 2018

1 Algorithms

1.1 Gradient and conjugate gradient methods

Write in Matlab M-files the codes corresponding to the gradient method with optimal step size and the conjugate gradient method. Check the codes for the following cost functions:

1. $f(x) = 2x_1^2 + x_2^2 - 2x_1x_2 - x_1 - x_2$
2. $f(x_1, x_2) = (x_1 - 1)^2 + 5(x_1^2 - x_2)^2$

Test the methods for different initial conditions. Analyze and compare the convergence rate of the two algorithms.

1.2 Quasi-Newton method

Write in a Matlab M-file the code corresponding to the quasi-Newton algorithm using the BFGS method for estimating the matrix S_k. Check the code for the following cost functions:

1. $f(x) = 2x_1^2 + x_2^2 - 2x_1x_2 - x_1 - x_2$
2. $f(x_1, x_2) = (x_1 - 1)^2 + 5(x_1^2 - x_2)^2$

Test the code for different initial conditions and compare the results with the algorithms developed in 1.1.

2 Case study for dimensioning

2.1 Design of a can

We want to design a can that holds at least 1.5 liters of liquid, as well as to meet other design requirements. The cans will be produced in the billions, so it is desirable to minimize their manufacturing costs. Since cost can be directly related to the surface area of the sheet metal used, it is reasonable to minimize the amount of sheet metal required. The dimension of the can is defined by the height H and the diameter D (H and D will be the design variables).

1. Write the expression of the cost function and solve the problem when no constraints occur.

2. Study the impact of the following restrictions on the size of the can: the diameter should be no more than 10 cm whereas the height should be no more than 18 cm and no less than 8 cm.

2.2 Tubular column design

The purpose is to design a minimum-mass tubular column (cf. Figure 1). The length of the column is L, the outer radius R_o and the inner radius R_i. The tubular column (cantilever column) is fixed at the base and free at the top. The column must support a load P without buckling or over-stressing. The critical load is given by:

$$P_C = \frac{\pi^2 EI}{4L^2} \tag{1}$$

where E is the Young's modulus and I the inertia of the cross section. The material stress σ for the column is defined as P/A, where A is the cross-sectional area of the column. The material allowable stress under the axial load is σ_a, and the material mass density is ρ (mass per volume unit). The design variables used for the optimization problem are R_o and R_i.

1. Write the expressions of the cost function and the constraints.

2. Solve the problem using the Matlab optimization toolbox for the following numerical values: $P = 14\text{kN}$, $L = 10\text{m}$, $\sigma_a = 0.5\text{MPa}$, $\rho = 7850\text{kg/m}^3$, $E = 210\text{GPa}$, $R_o \leq 40\text{cm}$, $R_i \leq 40\text{cm}$. Analyze the impact of P and σ_a on the dimensioning.

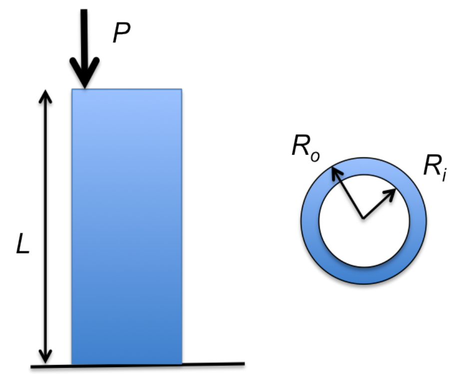

Figure 1: Tubular column

2.3 Spring design(Optional)

The objective aims at minimizing the weight of a tension/compression spring ([1]) (see Figure 2), subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and design variables. There are three design variables: the wire diameter x_1 (in mm), the mean coil diameter x_2 (in cm), and the number of active coils x_3. The mathematical formalisation leads to the following equations:

- Cost function:
$$f(x) = (x_3 + 2)x_2x_1^2$$

- Constraints:
$$g_1(x) = 1 - \frac{x_2^3 x_3}{7178x_1^4} \leq 0$$
$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566x_2x_1^3 - x_1^4} + \frac{1}{5108x_1^2} - 1 \leq 0$$
$$g_3(x) = 1 - \frac{140x_1}{x_2^2x_3} \leq 0$$
$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$
$$0.05 \leq x_1 \leq 2.0$$
$$0.25 \leq x_2 \leq 1.3$$
$$2.0 \leq x_3 \leq 15.0$$

[1] Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer, L. Cagnina and S. Esquivel, 2008

Solve the problem using the Matlab optimization toolbox.

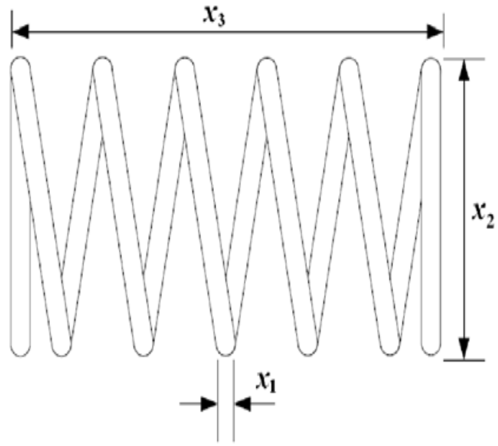

Figure 2: Spring design

3 Geometric shape optimization

We consider the optimization of the shape of a Linear Deformable Object (LDO) in 2D (see Figure 3) [2]. We denote $\{O, x_0, y_0\}$ the reference frame and $L = 1$m the length of the LDO. Let s, with $0 \leq s \leq L$, be the distance of a current point on the LDO along its central axis and let $\{p(s), \xi(s), \eta(s)\}$ be the frame at the current point characterized by the distance s. Therefore, the orientation of the vectors $\{\xi(s), \eta(s)\}$ with respect to the reference frame is defined by the angle $\theta(s)$. The shape results from the constraints applied at the endpoints of the LDO: at $s = 0$, we have $p(s = 0) = O$ and $\theta(s = 0) = \theta_0$, and at $s = L$, $p(s = L) = p_1$ and $\theta(s = L) = \theta_1$.

The function $\theta(s)$ is expressed as a linear combination of the following basis

[2] Modeling of Linear Objects Considering Bend, Twist, and Extensional Deformations, H. Wakamatsu, S. Hirai, and K. Iwata, IEEE lnternational Conference on Robotics and Automation, 1995

functions :

$$\phi_1(s) = 1, \quad \phi_2(s) = s, \tag{2}$$

$$\phi_3(s) = \sin\frac{2\pi s}{L}, \quad \phi_4(s) = \cos\frac{2\pi s}{L} \tag{3}$$

$$\phi_5(s) = \sin\frac{4\pi s}{L}, \quad \phi_6(s) = \cos\frac{4\pi s}{L} \tag{4}$$

such that:

$$\theta(s) = \sum_{i=1}^{6} a_i \, \phi_i(s) \tag{5}$$

where $a_i \in \mathbb{R}, i = 1, \cdots, 6$ are the shape variables of the LDO. Then, the position of a point on the LDO can be obtained by the following expression:

$$p(s) = \int_0^s \begin{pmatrix} \cos\theta(s) \\ \sin\theta(s) \end{pmatrix} ds \tag{6}$$

If we consider that the effects of gravity can be neglected, the potential energy of the LDO is equivalent to the torsional energy expressed as follows:

$$U(a) = \frac{1}{2}\int_0^L R_f \left(\frac{d\theta}{ds}\right)^2 ds \tag{7}$$

where $a = \begin{pmatrix} a_1 & \cdots & a_6 \end{pmatrix}^T$ is the vector of shape variables, and R_f represents the flexural rigidity coefficient of the LDO.

The optimization problem to solve is expressed as

$$\begin{gathered} \underset{a}{\text{minimize}}\ U(a) \\ \text{s. t.} \\ p(s=0) = O, \theta(s=0) = \theta_0 \\ p(s=L) = p_1, \theta(s=L) = \theta_1 \end{gathered} \tag{8}$$

where θ_0, p_1, θ_1 are user-defined constraints.

1. Write a Matlab function that takes as input the vector a and computes the energy U. Write a Matlab function that takes as input the vector a and computes the values of the constraints.

2. Solve the optimization problem and draw in 2D the shape of the LDO for different constraints θ_0, p_1, θ_1. Analyze the effects of the initial condition relating to a.

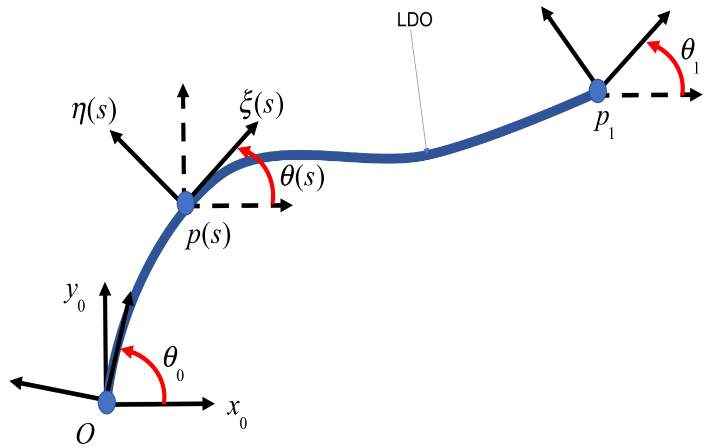

Figure 3: Linear Deformable Object

4 Robotics

4.1 Multi-objective optimization

The purpose is to control a multi arm robot as described on Figure 4. The control is defined by three different tasks that consist in reaching a desired position for the right and left hand, and a desired position for the Center of Mass (CoM). The design variables are the joint positions of the robot $q = (q_1, \cdots, q_5)^T$. The cost function for each task is expressed as the Euclidian distance from the current position (of the left and right hands, and of the CoM) to the desired position. This current position is obtained from the forward kinematics. The function *fminimax* will be used to solve the optimization problem.

1. Find the expressions of the forward kinematics model for the left and right hands, and for the CoM.

2. Write a Matlab function that computes the vector of the cost functions from the current value of q

3. Solve the control problem using Matlab optimization toolbox for different desired positions. Add to the problem the following joint limitations: $-\frac{\pi}{6} \leq q_1 \leq \frac{\pi}{6}$, $-\frac{\pi}{4} \leq q_2 \leq \frac{\pi}{4}$, $-\frac{5\pi}{6} \leq q_3 \leq \frac{5\pi}{6}$

For the numerical application, we take the following values:

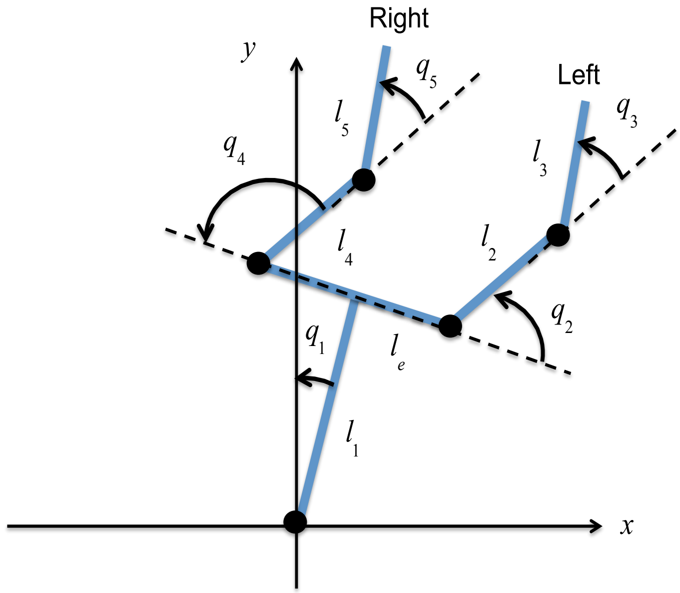

Figure 4: Multi arm robot

- $l_1 = 0.7$m, $l_e = 0.3$m, $l_2 = 0.4$m, $l_4 = l_2$, $l_3 = 0.3$m, $l_5 = l_3$.
- $m_1 = 20$kg, $m_2 = 6$kg, $m_4 = m_2$, $m_3 = 4$kg, $m_5 = m_3$
- $l_{1m} = 0.6$m, $l_{2m} = 0.2$m, $l_{4m} = l_{2m}$, $l_{3m} = 0.15$m, $l_{5m} = l_{3m}$

4.2 QP based inverse kinematics (Optional)

We aim at controlling a 3 degree-of-freedom planar robot (see Figure 5) using inverse kinematics defined by the equation $\dot{e} = J(q)\dot{q}$, where e is the tracking error defined by $e = P_d - P$, P_d being the desired position and $P = (x, y)^T$ the current position of the robot, and $J(q)$ is the Jacobian matrix. A simple solution consists in using the basic solution defined by $\dot{q} = J^+(q)\ \dot{e}$. However, this approach cannot be used when constraints appear. We suggest to use another approach based on quadratic programming. For this, we search for the minimization of the following cost function $f(\dot{q}) = \dot{q}^T W \dot{q} \geq 0$ subject to the constraints $\dot{e} = J(q)\dot{q}$ and $\dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max}$.

1. Write the forward kinematic model of the robot and compute the Jacobian matrix.

2. Using the Matlab optimization toolbox (use the function *quadprog*), solve the control problem that consists in reaching a desired position for the end-effector from an initial position. The tracking error will be specified as solution of the equation $\dot{e} + Ke = 0, K > 0$. The dynamics of the robot will be neglected and the joint values will be obtained by integration of $\dot{q}$.

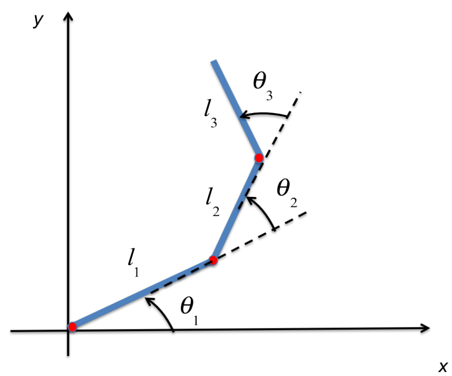

Figure 5: 3 dof robot

5 Topology optimization

The aim of this work is to analyze the implementation of an algorithm for topology optimization based on compliance minimization of statically loaded structures. For that, we will use the TOPOPT software (see http://www.topopt.dtu.dk). On the website, a Matlab code and an educational article [3] explaining how the code performs are available.

1. Download the Matlab code and the educational article.

2. Write a synthesis explaining both the principle of the topology optimization technique and how the optimization algorithm works. Analyze the impact of the parameters *volfrac*, *penal* and *rmin* on the simulation results.

3. Modify the Matlab code in order to introduce fixed holes in the design domain.

[3] O. Sigmund, *A 99 line topology optimization code written in Matlab*

6 Annexe

6.1 Forward kinematics

The robot is described on Figure 4. The position of the left end-effector in the plane (x, y) is defined by the following equations

$$x_L = -l_1 s_1 + l_e c_1 + l_2 c_{12} + l_3 c_{123} \tag{9}$$

$$y_L = l_1 c_1 + l_e s_1 + l_2 s_{12} + l_3 s_{123} \tag{10}$$

and the position of the right end-effector by

$$x_R = -l_1 s_1 - l_e c_1 - l_4 c_{14} - l_5 c_{145} \tag{11}$$

$$y_R = l_1 c_1 - l_e s_1 - l_4 s_{14} - l_5 s_{145} \tag{12}$$

where $c_j = \cos q_j$, $s_j = \sin q_j$, $c_{ij} = \cos(q_i + q_j)$, $s_{ij} = \sin(q_i + q_j)$.

6.2 Center of Mass

$m_1, \cdots, m_5$ are the mass of each segment, respectively. The lengths $l_{1m}, \cdots, l_{5m}$ define the position of the center of mass for each segment. The position of the center of mass is defined by

$$x_{COM} = -\frac{1}{M}\left(r_e c_1 + r_1 s_1 + r_2 c_{12} + r_3 c_{123} + r_4 c_{14} + r_5 c_{145}\right) \tag{13}$$

$$y_{COM} = \frac{1}{M}\left(r_1 c_1 - r_e s_1 - r_2 s_{12} - r_3 s_{123} - r_4 s_{14} - r_5 s_{145}\right) \tag{14}$$

where

$$\begin{aligned}
r_e &= l_e(m_4 + m_5 - m_3 - m_2) && (15)\\
r_1 &= l_1(m_2 + m_3 + m_4 + m_5) + l_{1m} m_1 && (16)\\
r_2 &= -l_2 m_3 - l_{2m} m_2 && (17)\\
r_3 &= -l_{3m} m_3 && (18)\\
r_4 &= l_4 m_5 + l_{4m} m_4 && (19)\\
r_5 &= l_{5m} m_5 && (20)\\
M &= m_1 + m_2 + m_3 + m_4 + m_5 && (21)
\end{aligned}$$