Advanced Robotics, 2013
Vol. 27, No. 1, 33–46, http://dx.doi.org/10.1080/01691864.2013.751162

Taylor & Francis
Taylor & Francis Group

FULL PAPER

Optimal design of nonlinear springs in robot mechanism: simultaneous design of trajectory and spring force profiles

Nicolas Schmit* and Masafumi Okada

Department of Mechanical Sciences and Engineering, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan

(*Received 3 April 2012; accepted 8 May 2012*)

In this paper, we aim at minimizing the actuator torques of robots working on production lines by adding to the mechanism dynamic equilibrators based on nonlinear springs, that work in parallel with the joints. We propose a method to simultaneously optimize the trajectory of the robot and the force profiles of the nonlinear springs to minimize the actuator torques. First, we express the trajectory and force profiles of the springs as a Hermite interpolation of a finite number of nodes, and then we show that the cost function of the optimization problem is a quadratic function of the springs design parameters. We derive a closed-form solution of the optimal spring parameters as a function of the trajectory parameters. As a consequence, the initial optimization problem is reduced to a trajectory optimization problem, solved with a sequential quadratic programming algorithm. We explain how the cost function can be modified to tune the nonlinearity of the springs and impose constraints on the stiffness. We show an example of optimal design of a three-degrees-of-freedom serial manipulator. Finally, we show that the nonlinear springs calculated for this manipulator can be technically realized by a noncircular cable spool mechanism.

Keywords: nonlinear spring; nonlinear stiffness; optimal design; robot; manipulator

1. Introduction

Robotic devices used for industrial processes are generally designed with stiff links and joints to ensure an accurate positioning of the end-effector. In such robots, the actuators have to counteract not only the gravity forces but also the inertial forces of the links which magnitudes increase when the operating speed increases. The actuators are used to change the mechanical energy of the system by providing positive or negative work. However, if the actuators are not back-drivable (like when used with gears with high reduction ratios), negative work is achieved by dissipating the mechanical energy, and the amount of energy to dissipate increases with the operating speed. This results in a poor energy efficiency, especially in high-speed motions.

In a robot undergoing a periodic motion, the introduction of springs in the mechanism can lead to a significant reduction of the energy consumption [1–3]. The elastic elements act as storage for energy which can be fed by the kinetic energy or the gravitational potential energy of the system. This energy then can be returned back to the system, the springs behaving as passive actuators. A common application of springs in machines or robots is the static compensation of gravity forces (spring equilibrators) [4–8]. Springs can also be designed to compensate for dynamic loads in periodic motions such as legged locomotion [2,9–11]. Compensation of dynamic loads using springs, though, work efficiently only if the frequency of the motion is close to the resonance frequency of the system [12]. Therefore, it is necessary to adapt the stiffness of the springs depending on the frequency of the motion. A simple solution is to use programmable springs [13]. A programmable spring consists in a series elastic actuator (SEA) [14] and a control loop programmed to make the actuator behave like a spring with the desired stiffness. The drawback of programmable springs is that the actuator of the SEA continuously consumes energy except in the case of a harmonic motion in which frequency matches the resonance frequency of the SEA's spring. Another solution is to use variable stiffness actuators (VSAs). VSAs are compliant actuators using two motors to control independently the force and the stiffness. A VSA can be realized by the antagonistic action of two SEAs [15–23] or use an actuator to provide the force and the other one to change the stiffness [1,24–30]. Similar to the case of constant springs, the stiffness of the VSA must be tuned so that the resonance frequency of the mechanism matches the frequency of the motion. This means that if the motion does not have a constant frequency, the

*Corresponding author. Email: schmit.n.aa@m.titech.ac.jp

stiffness must be tuned during the motion. However, in this case, most VSAs will become inefficient from an energetic point of view. This is due to the mechanism used to regulate the stiffness. In most of these actuators, compliance is controlled through adjusting the pretension of elastic elements. In other words, the stiffness is changed by adding or subtracting potential energy to springs. Since the motors are working against the restoring forces of the springs, changing the stiffness of the mechanism demands considerably high energy [30], except if the VSA is designed with the motors working perpendicularly to the force of the springs [1].

In this paper, we propose a different design strategy based on nonlinear springs placed in parallel with the actuators. Since the springs are nonlinear, the stiffness is a function of the displacement of the spring, thus a function of the joint coordinates. Since the joint coordinates change when the robot moves, the stiffness becomes a time-varying function. Consequently, the nonlinear springs behave like VSAs but do not consume energy to change the stiffness. In order to minimize the energy consumption, we want to determine the optimal spring force profiles and the optimal trajectory. On the one hand, if we know the spring force profiles, we can determine the optimal trajectory using the theory of optimal control. One the other hand, if we know the trajectory, we can determine the optimal spring force profiles using inverse dynamics. Optimizing *both* the spring force profiles and the trajectory is a tricky problem because the optimal spring force profiles depend on the trajectory and vice versa. In [2,10,12,31–34], several methods were proposed to simultaneously optimize the trajectory and the stiffness of the springs, but they are limited to linear springs. In [35], a method was developed to simultaneously optimize the trajectory and the time-varying stiffness of VSAs. However, this method is not suitable to solve our problem since in [35], the stiffness is an arbitrary function of time whereas in the case of nonlinear springs, the stiffness is a function of the joint coordinates.

We propose an optimal design method to simultaneously design the trajectory of a robot and the torque profiles of nonlinear springs located at each joint and acting in parallel with the actuators. The optimization process aims at minimizing the actuator torque[1]. First, we discretize the trajectory and the nonlinear torque profiles of the springs with a finite number of parameters (thereafter referred as 'trajectory parameters' and 'spring parameters,' respectively). Then, we show that the cost function of the optimization problem, defined as the integral of the actuator torques, is a quadratic function of the spring parameters. We express the optimal spring parameters as a function of the trajectory parameters, and rewrite the optimization problem so that it depends only on the trajectory parameters. We explain how the cost function can be modified to tune the nonlinearity of the springs and impose local conditions on the stiffness. We use an sequential quadratic programming (SQP) algorithm to optimize the trajectory. We show an example of optimal design of a three-DOF serial manipulator. Finally, we show that the optimal nonlinear springs calculated for this manipulator can be realized by a noncircular cable spool mechanism [36].

Since the nonlinear spring equilibrators are optimized for a given trajectory, this method is suitable for robots working on production lines and achieving repetitive tasks such as picking or assembling.

2. Problem statement

2.1. Equations of motion and design constraints

We consider a robot with n degrees of freedom (DOFs). Each joint is driven by an actuator and a nonlinear spring which act in parallel (the joint torque is the sum of the actuator torque and spring torque). Assuming that the mass–inertia parameters of the nonlinear springs are negligible compared with those of the system elements, the equations of motion of the robot can be written as:

$$A(q)\ddot{q} + B(q,\dot{q}) = u(t) + w(q) \tag{1}$$

where, t is the time, $q = (q_1(t), q_2(t), \ldots, q_n(t))^T$ is the column vector of joint variables, $u = (u_1(t), u_2(t), \ldots, u_n(t))^T$ is the column vector of actuator torques, $w = (w_1(q_1), w_2(q_2), \ldots, w_n(q_n))^T$ is the column vector of nonlinear spring torques, A(q) is the mass matrix, and $B(q,\dot{q})$ is a matrix which contains the Coriolis, centrifuge, and gravity terms. Note that the torque of each nonlinear spring w_i depends only on the joint variable q_i of the joint where the spring is located. All w_i are continuously differentiable functions.

Different constraints are imposed on the trajectory, actuator torques, and spring torques. These constraints have the form:

$$\mathcal{G}(q(0), \dot{q}(0), q(T), \dot{q}(T)) = 0 \tag{2}$$

$$\{q(t), \dot{q}(t)\} \in \mathcal{Q}, t \in [0, T] \tag{3}$$

$$w(q) \in W, q \in [q_{min}, q_{max}] \tag{4}$$

$$u(t) \in U, t \in [0, T] \tag{5}$$

$$\mathcal{F}_i(q(t), \dot{q}(t), t) \leq 0 \tag{6}$$

where, $\mathcal{G}$ are the boundary conditions, $\mathcal{Q}$ is a given domain of the phase space of the system under consideration (joint angular range and maximum velocity) and W

and U are the set of permissible nonlinear spring torques and actuator torques, respectively. $\mathcal{F}_i$ are constraints imposed to the trajectory depending on the robot task. For example, if the robot has to stop to the coordinates $\bar{q}$ from t_1 to t_2, we use the constraints:

$$\mathcal{F}_1 = (q_1(t) - \bar{q}_1)^2 \ (H(t-t_1) - H(t-t_2)) \tag{7}$$

$$\cdots = \cdots$$

$$\mathcal{F}_n = (q_n(t) - \bar{q}_n)^2 \ (H(t-t_1) - H(t-t_2)) \tag{8}$$

$$\mathcal{F}_{n+1} = (\dot{q}_1(t))^2 \ (H(t-t_1) - H(t-t_2)) \tag{9}$$

$$\cdots = \cdots$$

$$\mathcal{F}_{2n} = (\dot{q}_n(t))^2 \ (H(t-t_1) - H(t-t_2)) \tag{10}$$

where, H is the Heaviside step function.

2.2. Cost function

We define the cost function of the optimization problem as the time integral of the actuator torques:

$$\mathcal{C} = \frac{1}{2} \int_0^T ||u(t)||^2 \mathrm{d}t \tag{11}$$

where, $t=0$ and $t=T$ are the start and finish time of the motion. We define $u_0(t)$ as the required actuator torque vector to achieve a given trajectory $q(t)$ when the joints are driven only by the actuators (no springs). $u_0(t)$ is calculated by inverse dynamics as:

$$u_0(t) = A(q(t))\ddot{q}(t) + B(q(t), \dot{q}(t)) \tag{12}$$

When the joints are driven by both the actuators and nonlinear springs, the required actuator torque vector $u(t)$ to achieve a trajectory $q(t)$ is calculated as:

$$u(t) = u_0(t) - w(q(t)) \tag{13}$$

Substituting (13) in (11), we rewrite the cost function as:

$$\mathcal{C} = \frac{1}{2} \int_0^T ||u_0(t)||^2 \mathrm{d}t - \int_0^T w^T(q(t)) u_0(t) \ \mathrm{d}t + \frac{1}{2} \int_0^T ||w(q(t))||^2 \mathrm{d}t \tag{14}$$

The optimization problem consists in finding the trajectory $q(t)$ and torque profiles of the nonlinear springs $w(q)$ which minimize $\mathcal{C}$ while satisfying the boundary conditions (2) and constraints (3,4,5,6).

3. Parameterization of the trajectory and spring torque profiles

For each joint i, the range of the joint variable $[q_{i,\min}, q_{i,\max}]$ is divided into N equal subintervals of length Δq_i. These intervals define $N+1$ nodes:

$$q_{ij} = (j-1)\Delta q_i + q_{i,\min}, \quad j \in \{1..N+1\}, \quad \text{with} \quad q_{i,N+1} = q_{i,\max} \tag{15}$$

A third-order polynomial is used to represent the torque w_i of the ith nonlinear spring on each subinterval. The value of w_i and its derivative at the nodes $q_{i,j}$ are used as design parameters. For each subinterval, the unique third-order polynomial is defined that has values $(f_{i,j}, f_{i,j+1})$ and derivatives $(s_{i,j}, s_{i,j+1})$ at the end points of $[q_{i,j}, q_{i,j+1})$. This definition allows the spring torque at any point $q_i = q_{i,j} + \rho_i \Delta q_i$ in $[q_{i,j}, q_{i,j+1})$ to be written as:

$$w_i(q_i \in [q_{i,j}, q_{i,j+1})) = \alpha_i f_{i,j+1} + \beta_i f_{i,j} + (\gamma_i s_{i,j+1} + \delta_i s_{i,j})\Delta q_i \tag{16}$$

where, α_i, β_i, γ_i, δ_i are calculated as:

$$\alpha_i = \rho_i^2(3 - 2\rho_i) \tag{17}$$

$$\beta_i = 2\rho_i^3 - 3\rho_i^2 + 1 \tag{18}$$

$$\gamma_i = \rho_i^2(\rho_i - 1) \tag{19}$$

$$\delta_i = \rho_i(\rho_i - 1)^2 \tag{20}$$

$$\rho_i(q_i \in [q_{i,\min}, q_{i,\max})) = \frac{q_i - q_{i,\min}}{\Delta q_i} \bmod 1, \quad \rho_i(q_{i,\max}) = 1 \tag{21}$$

This scheme, called Hermite interpolation, automatically gives continuity of the torque and its derivative at the nodes. An example of torque profile is shown in Figure 1.

From (16), the spring torque w_i at any point of $[q_{i,\min}, q_{i,\max}]$ is calculated as:

$$w_i(q_i) = \sum_{j=1}^{N} \left[(\alpha_i f_{ij+1} + \beta_i f_{ij} + (\gamma_i s_{ij+1} + \delta_i s_{ij})\Delta q_i) \ \chi_{ij}(q_i) \right] \tag{22}$$

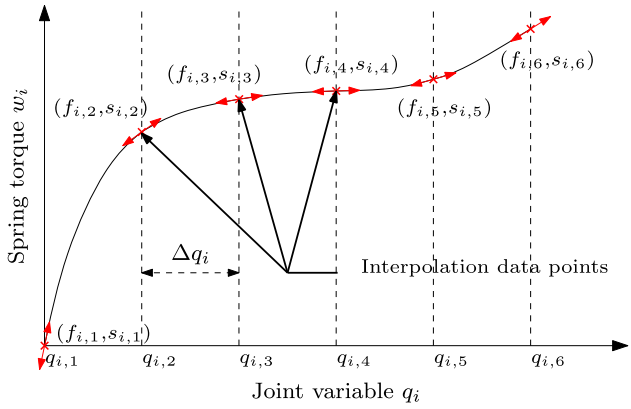


Figure 1. Interpolation of a spring torque profile. The values of the torque $f_{i,j}$ and its derivative $s_{i,uj}$ at each node are used as design parameters. On each subinterval, the torque is expressed by a third-order polynomial uniquely defined by its values and derivatives at the endpoints of the subinterval.

$$\chi_{ij}\,(q_i) = 1 \text{ if } q_i \in [q_{ij}, q_{ij+1}]), \quad 0 \text{ if } q_i \notin [q_{ij}, q_{ij+1}]) \tag{23}$$

$$\chi_{i,\,N}\,(q_{i,\max}) = 1$$

where, the functions $\chi_{i,j}$ are the indicator functions of the subintervals $[q_{i,j}, q_{i,j+1})$. We gather all the design parameters $(f_{i,j}, s_{i,j}\Delta q_i)$ in a vector y_i, and rewrite (22) as[2]:

$$w_i(q_i) = \psi_i^T(q_i) y_i \tag{24}$$

$$y_i = [f_{i,1}, \quad (s_{i,1}\Delta q_i), \quad \ldots, \quad f_{i,j}, \quad (s_{i,j}\Delta q_i), \quad \ldots, \quad f_{i,N+1}, \quad (s_{i,N+1}\Delta q_i)]^T \tag{25}$$

$$\psi_i(q_i) = [\beta_i\chi_{i,1},\ \delta_i\chi_{i,1},\ \ldots,\ (\alpha_i\chi_{i,j-1} + \beta_i\chi_{i,j}),\ (\gamma_i\chi_{i,j-1} + \delta_i\chi_{i,j}),\ \ldots,\ \alpha_i\chi_{i,N},\ \gamma_i\chi_{i,N}]^T \tag{26}$$

Finally, the vector of all spring torques is expressed as:

$$w(q) = \Psi^T(q) Y \tag{27}$$

$$Y = [y_1^T, \ldots, y_n^T]^T \tag{28}$$

$$\Psi(q) = \text{blockdiag}(\psi_1(q_1), \ldots, \psi_n(q_n)) \tag{29}$$

where the operator blockdiag stands for 'block diagonal matrix',$\Psi(q)$ is a matrix function of the joint variables, and Y is a vector which contains the interpolation data points of all torque profiles of the nonlinear springs. The elements of Y are thereafter referred as 'spring parameters'.

The trajectory is parameterized in a similar way, the only difference being that the interpolation nodes are the same for all DOFs (because the time variable is the same for all DOFs). The total time interval $[0, T]$ is divided into N_t equal subintervals of length Δt. These intervals define $N_t + 1$ nodes:

$$t_k = (k-1)\Delta t, \quad k \in \{1 \ldots N_t + 1\}, \quad \text{with} \quad t_{N_t+1} = T \tag{30}$$

A third-order polynomial is used to represent the joint variable q_i on each subinterval. The value of the position $x_{i,k}$ and velocity $v_{i,k}$ at the nodes t_k are used as design parameters. Using a similar methodology as for the parameterization of the torque profiles of the nonlinear springs, we express the ith joint variable as:

$$q_i(t) = \varphi_i^T(t)\, x_i \tag{31}$$

$$x_i = [x_{i,1}, \quad (v_{i,1}\Delta t), \quad \ldots, \quad x_{i,k}, \quad (v_{i,k}\Delta t), \quad \ldots, \quad x_{i,N_t+1}, \quad (v_{i,N_t+1}\Delta t)]^T \tag{32}$$

$$\varphi_i(t) = [\beta\tau_1, \quad \delta\tau_1, \quad \ldots, \quad (\alpha\tau_{k-1} + \beta\tau_k), \quad (\gamma\tau_{k-1} + \delta\tau_k), \quad \ldots, \quad \alpha\tau_{N_t}, \quad \gamma\tau_{N_t}]^T \tag{33}$$

where the functions τ_k are the indicator functions of the subintervals $[t_k, t_{k+1})$. Finally, the vector of all joint variables is expressed as:

$$q(t) = \Phi^T(t) X \tag{34}$$

$$X = [x_1^T, \ldots, x_n^T]^T \tag{35}$$

$$\Phi(t) = \text{blockdiag}\,(\varphi_1(t), \ldots, \varphi_n(t)) \tag{36}$$

where $\Phi(t)$ is a time dependent matrix. X contains the interpolation data points of all DOFs of the trajectory. The elements of X are thereafter referred as 'trajectory parameters.'

Thereafter, the optimization problem consists in finding the trajectory parameters X and spring parameters Y which minimize $\mathcal{C}$ while satisfying (2)–(6).

4. Optimization of spring and trajectory parameters

4.1. Cost function at optimal spring design

Substituting (27) and (34) in (14), we rewrite the cost function as:

$$\mathcal{C}(X,Y) = \frac{1}{2}\int_0^T ||u_0(t)||^2 \mathrm{d}t - \int_0^T Y^T \Psi(\Phi^T(t)X) u_0(t) \mathrm{d}t + \frac{1}{2}\int_0^T Y^T \Psi(\Phi^T(t)X)\Psi^T(\Phi^T(t)X) Y \mathrm{d}t \quad (37)$$

Since Y is not time-dependent, we can rewrite $\mathcal{C}$ as:

$$\mathcal{C}(X,Y) = \mathcal{C}_0(X) - Z^T(X)Y + \frac{1}{2}Y^T K(X) Y \quad (38)$$

$$\mathcal{C}_0(X) = \frac{1}{2}\int_0^T ||u_0(t)||^2 \mathrm{d}t \quad (39)$$

$$Z(X) = \int_0^T \Psi(\Phi^T(t)X) u_0(t) \mathrm{d}t \quad (40)$$

$$K(X) = \int_0^T \Psi(\Phi^T(t)X)\Psi^T(\Phi^T(t)X) \mathrm{d}t \quad (41)$$

where K is a symmetric matrix. K, Z and $\mathcal{C}_0$ are functions of the trajectory parameters X but independent from the spring parameters Y.

Formally, the optimal parameters (X^*,Y^*) are obtained by minimizing $\mathcal{C}$ with respect to X and Y simultaneously:

$$\left.\frac{\partial \mathcal{C}(X,Y)}{\partial X}\right|_{X=X^*} = 0 \quad (42)$$

$$\left.\frac{\partial \mathcal{C}(X,Y)}{\partial Y}\right|_{Y=Y^*} = 0 \quad (43)$$

However, since (38) is a quadratic function of Y, we can solve (43) and express the optimal spring parameters as a function of the trajectory parameters[3].

$$Y^*(X) = K^{-1}(X)\, Z(X) \quad (44)$$

Substituting (44) in (38), we obtain the *cost function at optimal spring design* $\mathcal{C}^*$ as:

$$\mathcal{C}^*(X) = \mathcal{C}(X, Y^*(X)) = \mathcal{C}_0(X) - \frac{1}{2} Z^T(X) K^{-1}(X) Z(X) \quad (45)$$

Since the optimization of the nonlinear springs is 'embedded' in $\mathcal{C}^*$, this function depends only on the trajectory parameters X. The first term of $\mathcal{C}^*$ evaluates the trajectory when the robot is driven only by the actuators, and the second term evaluates the improvement due to the contribution of the nonlinear springs.

4.2. Adjusting the nonlinearity of the springs

Depending on the trajectory parameters, the optimal spring parameters may result in highly nonlinear torque profiles which would not be realizable technically. Thus, in order to control the nonlinearity of the springs, we add to (38) a term which weights the nonlinearity of the profiles. A geometrical way to measure the nonlinearity is to use the mean value of the square of the curvature κ of the torque profile:

$$\frac{1}{N\Delta q_i}\int_{q_{i,\min}}^{q_{i,\max}} \kappa^2 \, \mathrm{d}q_i \quad \text{with} \quad \kappa = \frac{\frac{\mathrm{d}^2 w}{\mathrm{d}q^2}}{\left(1 + \left(\frac{\mathrm{d}w}{\mathrm{d}q}\right)^2\right)^{\frac{3}{2}}} \quad (46)$$

Since $\kappa \leq |\frac{\mathrm{d}^2 w}{\mathrm{d}q^2}|$, we can simplify (46) by dropping the term $\frac{\mathrm{d}w}{\mathrm{d}q}$. This provides a more tractable expression to weight the curvature.

$$\frac{1}{N\Delta q_i}\int_{q_{i,\min}}^{q_{i,\max}} \left(\frac{\mathrm{d}^2 w}{\mathrm{d}q^2}\right)^2 \mathrm{d}q_i \quad (47)$$

Using (47) to weight the nonlinearity of the torque profiles, the cost function (38) becomes:

$$\mathcal{C}(X,Y) = \mathcal{C}_0(X) - Z^T(X)Y + \frac{1}{2}Y^T K(X) Y + \frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N\Delta q_i}\int_{q_{i,\min}}^{q_{i,\max}} \left(\frac{\mathrm{d}^2 w_i(q_i)}{\mathrm{d}q_i^2}\right)^2 \mathrm{d}q_i\right] \quad (48)$$

λ_i are weighting coefficients that the designer can modify to adjust the nonlinearity of each torque profile. The higher the values of λ_i, the more linear the springs. If all λ_i are set to zero, (48) is equivalent to (38).

Substituting, (24) in (48), we rewrite the new term as:

$$\frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N\Delta q_i}\int_{q_{i,\min}}^{q_{i,\max}} \left(\frac{\mathrm{d}^2 w_i(q_i)}{\mathrm{d}q_i^2}\right)^2 \mathrm{d}q_i\right] = \frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N\Delta q_i} y_i^T \int_{q_{i,\min}}^{q_{i,\max}} \frac{\mathrm{d}^2\psi_i \mathrm{d}^2\psi_i^T}{\mathrm{d}q_i^2 \, \mathrm{d}q_i^2} \, \mathrm{d}q_i \; y_i\right] \quad (49)$$

We decompose ψ_i as follows:

$$\psi_i = \sum_{j=1}^{N} \widetilde{\psi}_{i,j}\ \chi_{i,j} \tag{50}$$

$$\widetilde{\psi}_{i,j} = [\ \underbrace{0..0}_{2(j-1)\text{ zeros}}\ , \beta_i, \delta_i, \alpha_i, \gamma_i, \underbrace{0..0}_{2(N-j)\text{ zeros}}\ ]^T \tag{51}$$

where α_i, β_i, γ_i, δ_i, and $\chi_{i,j}$, were defined in (17)–(20), (23), respectively. Substituting (50) in (49), we obtain:

$$\frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N\Delta q_i} y_i^T \int_{q_{i,\min}}^{q_{i,\max}} \frac{\mathrm{d}^2\psi_i \mathrm{d}^2\psi_i^T}{\mathrm{d}q_i^2\ \mathrm{d}q_i^2}\,\mathrm{d}q_i\ y_i\right] = \frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N(\Delta q_i)^4} y_i^T \sum_{j=1}^{N}\left(\int_0^1 \frac{\mathrm{d}^2\widetilde{\psi}_{i,j}\mathrm{d}^2\widetilde{\psi}_{i,j}^T}{\mathrm{d}\rho_i^2\ \mathrm{d}\rho_i^2}\,\mathrm{d}\rho_i\right) y_i\right] \tag{52}$$

where ρ_i was defined in (21). Since α_i, β_i, γ_i, and δ_i are third-order polynomials of ρ_i, the integral term in (52) can be calculated by integration by parts as:

$$\int_0^1 \frac{\mathrm{d}^2\widetilde{\psi}_{i,j}\mathrm{d}^2\widetilde{\psi}_{i,j}^T}{\mathrm{d}\rho_i^2\ \mathrm{d}\rho_i^2}\,\mathrm{d}\rho_i = \left[\frac{\mathrm{d}\widetilde{\psi}_{i,j}\mathrm{d}^2\widetilde{\psi}_{i,j}^T}{\mathrm{d}\rho_i\ \mathrm{d}\rho_i^2}\right]_{\rho_i=0}^{\rho_i=1} - \left[\widetilde{\psi}_{i,j}\frac{\mathrm{d}^3\widetilde{\psi}_{i,j}^T}{\mathrm{d}\rho_i^3}\right]_{\rho_i=0}^{\rho_i=1} \tag{53}$$

Calculating (53) from the equations of α_i, β_i, γ_i, δ_i, we obtain

$$\frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N(\Delta q_i)^4} y_i^T \sum_{j=1}^{N}\left(\int_0^1 \frac{\mathrm{d}^2\widetilde{\psi}_{i,j}\mathrm{d}^2\widetilde{\psi}_{i,j}^T}{\mathrm{d}\rho_i^2\ \mathrm{d}\rho_i^2}\,\mathrm{d}\rho_i\right) y_i\right] = \frac{1}{2}\sum_{i=1}^{n}\left[\frac{\lambda_i}{N(\Delta q_i)^4} y_i^T\ Q\ y_i\right] \tag{54}$$

where, the matrix Q is calculated as:

$$Q = \begin{bmatrix} Q_0 & Q_2 & & & \\ Q_2^T & Q_3 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & Q_3 & Q_2 \\ & & & Q_2^T & Q_1 \end{bmatrix} \tag{55}$$

$$Q_0 = \begin{bmatrix} 12 & 6 \\ 6 & 4 \end{bmatrix} \tag{56}$$

$$Q_1 = \begin{bmatrix} 12 & -6 \\ -6 & 4 \end{bmatrix} \tag{57}$$

$$Q_2 = \begin{bmatrix} -12 & 6 \\ -6 & 4 \end{bmatrix} \tag{58}$$

$$Q_3 = \begin{bmatrix} 24 & 0 \\ 0 & 8 \end{bmatrix} \tag{59}$$

Finally, (48) is equivalent to

$$\mathcal{C}(X,Y) = \mathcal{C}_0(X) - Z^T(X)Y + \frac{1}{2}Y^T K(X)Y + \frac{1}{2}Y^T G Y \tag{60}$$

where, the matrix G is defined as:

$$G = \mathrm{blockdiag}\left(\frac{\lambda_1}{N(\Delta q_1)^4}\,Q, \ldots, \frac{\lambda_n}{N(\Delta q_n)^4}\,Q\right) \tag{61}$$

We see that we can control the nonlinearity of the spring torque profiles by adding to the cost function a term $\frac{1}{2}Y^T G Y$, where the matrix G is symmetric. From (60), we derive the modified cost function at optimal spring design:

$$\mathcal{C}^*(X) = \mathcal{C}_0(X) - \frac{1}{2}Z^T(X)(K(X)+G)^{-1}Z(X) \tag{62}$$

If all weighting coefficients λ_i are set to zero, (62) is equivalent to (45). Note that since the matrix Q is constant, it can be calculated before the optimization process in order to reduce the total calculation cost.

4.3. *Adding constraints to impose positive stiffness*

So far, the only constraint that we have imposed to the load–displacement functions of the spring is to be C^1 piecewise third-order polynomial functions (this constraint is imposed implicitly by the choice of the interpolation method). Consequently, the load-displacement functions defined by the optimal spring parameters in (44) may exhibit negative stiffness. If the robot has to keep a static position for a long time, a design with negative stiffness may not be energetically optimal because the control loop will consume energy to artificially stabilize the system. Therefore, in this section, we propose a methodology to impose the stiffness to be positive at given joint coordinates (chosen by the designer).

We consider the ith DOF of the robot. For a given joint variable q_i, the stiffness of the spring is given by $-\frac{dw_i(q_i)}{dq_i}$ (since $w_i(q_i)$ is defined as the force applied by the spring to the joint, the restoring force of the spring is $-w_i(q_i)$). We note $\overline{q}_{i,j}$ as the joint variables where we

constrain the stiffness, and $n_{\overline{q}_i}$ as the number of constraints applied to the ith DOF. For each joint variable $\overline{q}_{i,j}$, we impose the spring stiffness to be at least equal to $d_{i,j}>0$. The set of constraints can be written as:

$$\forall j \in \{1 \dots n_{\overline{q}_i}\}, -\frac{\mathrm{d}w_i(q_i)}{\mathrm{d}q_i}\bigg|_{q_i=\overline{q}_{i,j}} \geq \mathrm{d}_{i,j} \tag{63}$$

From (24), $\frac{\mathrm{d}w_i(q_i)}{\mathrm{d}q_i}$ is calculated as:

$$\frac{\mathrm{d}w_i(q_i)}{\mathrm{d}q_i} = \psi_i'^T(q_i)y_i \tag{64}$$

where ψ_i' is calculated as:

$$\psi_i'(q_i) = \frac{1}{\Delta q_i}[\beta_i'\chi_{i,1},\ \delta_i'\chi_{i,1},\ \dots,\ (\alpha_i'\chi_{i,j-1} + \beta_i'\chi_{i,j}),\ (\gamma_i'\chi_{i,j-1} + \delta_i'\chi_{i,j}),\ \dots,\ \alpha_i'\chi_{i,N},\ \gamma_i'\chi_{i,N}]^T \tag{65}$$

$$\alpha_i' = 6\rho_i(1-\rho_i) \tag{66}$$

$$\beta_i' = 6\rho_i(\rho_i - 1) \tag{67}$$

$$\gamma_i' = 3\rho_i^2 - 2\rho_i \tag{68}$$

$$\delta_i' = 3\rho_i^2 - 4\rho_i + 1 \tag{69}$$

ρ_i is calculated using (21). We rewrite (63) as

$$\Theta_i^T y_i + \mathcal{D}_i \leq 0 \tag{70}$$

where

$$\Theta_i = [\psi_i'(\overline{q}_{i,1}), \dots, \psi_i'(\overline{q}_{i,n_{\overline{q}_i}})] \tag{71}$$

$$\mathcal{D}_i = [d_{i,1}, \dots, d_{i,n_{\overline{q}_i}}]^T \tag{72}$$

The notation ≤ 0 in (70) means that each element of the vector $\Theta_i^T y_i + \mathcal{D}_i$ must be negative. Gathering the constraint relationships of all DOFs, we rewrite (70) as

$$\Theta^T Y + \mathcal{D} \leq 0 \tag{73}$$

where,

$$\Theta = \mathrm{blockdiag}(\Theta_1, \dots, \Theta_n) \tag{74}$$

$$D = [\mathcal{D}_1^T, ..., \mathcal{D}_n^T]^T \tag{75}$$

The cost function $\mathcal{C}$ defined in (38) is continuously differentiable in Y (because it is a quadratic function) and the inequality constraints (73) are linear in Y. Consequently, if Y^* is the vector of optimal spring parameters (i.e. which minimizes $\mathcal{C}$) under (73), then there exists a vector of constants η called Karush-Kuhn-Tucker (KKT) multipliers such that

$$\nabla_Y[\mathcal{C}(X, Y^*) + \eta^T(\Theta^T Y^* + \mathcal{D})] = 0 \Leftrightarrow KY^* - Z + \Theta\eta = 0 \tag{76}$$

$$\Theta^T Y^* + \mathcal{D} \leq 0 \tag{77}$$

$$\eta \geq 0 \tag{78}$$

$$\eta^T(\Theta^T Y^* + \mathcal{D}) = 0 \tag{79}$$

(76) is the stationary condition, (77) is the primary feasibility condition, (78) is the dual feasibility condition, and (79) is the complementary slackness condition. The KKT approach generalizes the method of Lagrange multipliers, which allows only equality constraints. See [37] for the derivation of the KKT conditions. One of the main issue in solving optimization problems with inequality constraints is to determine which constraints are active and which constraints are not. In this paper, we proceed the following way: we consider all the possible combinations of active/inactive constraints, and solve each optimization problem successively. For each problem, we verify that the solution (Y^*, η) satisfies all the KKT feasibility conditions. If several solutions verify all the feasibility conditions, we choose the solution with the Y^* which minimizes $\mathcal{C}$.

For a given combination of active/inactive constraints, the optimization problem is solved as follows: we note $(\Theta_{\mathrm{act}}, \mathcal{D}_{\mathrm{act}})$ and $(\Theta_{\mathrm{inact}}, \mathcal{D}_{\mathrm{inact}})$ as the couple $(\Theta, \mathcal{D})$ built by considering the active and inactive constraints, respectively. The vector of KKT multipliers related to the active constraints is η_{act}. Note that as a consequence of (79), $\eta_{\mathrm{inact}} = 0$. Rewriting the KKT conditions with these new variables, the vector of optimal spring parameters is solution of the problem:

$$KY^* - Z + \Theta_{\mathrm{act}}\eta_{\mathrm{act}} = 0 \tag{80}$$

$$\Theta_{\mathrm{act}}^T Y^* + \mathcal{D}_{\mathrm{act}} = 0 \tag{81}$$

$$\Theta_{\mathrm{inact}}^T Y^* + \mathcal{D}_{\mathrm{inact}} < 0 \tag{82}$$

$$\eta_{\mathrm{act}} > 0 \tag{83}$$

Assuming that $(\Theta_{act}^T K^{-1}\Theta_{\text{act}})$ is invertible, we solve the system (80), (81) with respect to Y^* and η. We obtain:

$$Y^* = K^{-1}(Z - \Theta_{\text{act}}\mathcal{K}^{-1}\mathcal{Z}) \tag{84}$$

$$\eta_{\text{act}} = \mathcal{K}^{-1}\mathcal{Z} \tag{85}$$

where,

$$\mathcal{K} = \Theta_{\text{act}}^T K^{-1}\Theta_{\text{act}} \tag{86}$$

$$\mathcal{Z} = \Theta_{\text{act}}^T K^{-1} Z + \mathcal{D}_{\text{act}} \tag{87}$$

Note that $\mathcal{K}$ is a symmetric matrix. If Y^* satisfies the conditions (82) and (83), we calculate the cost function at optimal spring design $\mathcal{C}^*(X) = \mathcal{C}(X, Y^*(X))$ with the Y^* calculated in (84):

$$\mathcal{C}^*(X) = \mathcal{C}_0(X) - \frac{1}{2} Z^T(X)K^{-1}(X)Z(X) + \frac{1}{2}\mathcal{Z}^T(X)\mathcal{K}^{-1}(X)\mathcal{Z}(X) \tag{88}$$

The cost function defined in (88) evaluates the trajectory parameters for the optimal springs design satisfying the constraints (63). Note that if we also want to adjust the nonlinearity of the springs, we just have to replace K by $(K + G)$ in all formulae, where the matrix G was defined in Section 4.2.

4.4. *Optimization of trajectory parameters*

In Section 4.1, we defined the *cost function at optimal spring design* $\mathcal{C}^*$ (45) which depends only on the trajectory parameters. The next step consists in finding the optimal trajectory parameters, which means finding the vector X which minimizes $\mathcal{C}^*$. However, since this function is nonlinear, it is not possible to derive a closed form expression of the optimal trajectory parameters, except for a few simple cases. Thus, we use a SQP algorithm to find an approximate solution of the optimal trajectory. SQP methods solve a sequence of optimization subproblems, each which optimizes a quadratic model of the objective subject to a linearization of the constraints. We used the SQP method implemented in Matlab to optimize the trajectory parameters. Details on SQP methods can be found in [38].

5. Example of optimal design

We consider the 3-DOF serial manipulator shown in Figure 2. ℓ_i is the length of link i, h_i is the distance from the origin of the coordinate frame attached to link i to the link's center of mass, θ_i is the angular displacement of coordinate frame i with respect to coordinate frame $(i-1)$ (coordinate frame 0 is the reference frame), and g is the acceleration of gravity. As design constraints, we impose the end effector to stop at prescribed positions during chosen time intervals. These constraints are summarized in Table 1. From time 0 to 1 s, the end effector must stop to coordinates (30, 10, 20), from time 5 to 6 s, the end effector must stop to coordinates (60, 80, 40), etc. Note that we imposed a same constraint at $t = 0$ s and $t = 20$ s in order to synthesize a periodic motion.

The initial and optimal trajectories are shown in Figure 3. The labels p_1–p_5 indicate the constrained parts of the trajectory. The initial and optimal paths are shown in Figure 4. The manipulator (in blue) is drawn at time $t = 8$ s. The label p_5 is not displayed because it would appear exactly above the label p_1.

In the left column of Figure 5, we show the restoring torques of the nonlinear springs. The vertical dashed lines show the location of the interpolation nodes of the torque profiles. An increasing restoring torque corresponds to positive stiffness while a decreasing restoring torque corresponds to negative stiffness. In the right column of Figure 5, we show the torques applied to each joint by their nonlinear spring, actuator, and the total torque. We can see on Figure 5(d) and (f) that a significant part of the total torque is provided by the nonlinear springs, thus reducing the average actuator torque. In Table 2, we show the average absolute actuator torques for a design with no springs, a design with optimal linear springs and a design with optimal nonlinear springs. The first row shows the value of the actuator torques for the initial trajectory and the second row shows the values of the actuator torques for the optimal trajectory. We precise

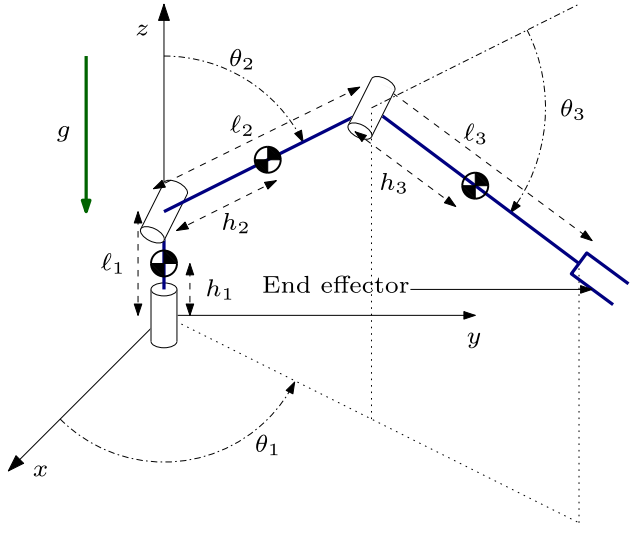


Figure 2. 3-DOF serial manipulator.

Table 1. Time constraints on the position of the end effector.

	t (s)	x (cm)	y (cm)	z (cm)
p_1	[0,1]	30	10	20
p_2	[5,6]	60	80	40
p_3	[10,11]	−50	−40	70
p_4	[15,16]	−50	100	120
p_5	20	30	10	20

that the optimal trajectories are different for the three designs: no springs/linear springs/nonlinear springs. The results for the design with linear springs were obtained by setting the coefficients λ_i (see Section 4.2) big enough so that the torque profiles of the springs are almost linear. From this table, we understand that the design with the best performances is obtained by a simultaneous optimization of the trajectory and the nonlinear spring torque profiles.

As explained in Section 4.2, we can control the nonlinearity of the springs by tuning the design parameters λ_i. In Figure 6, we show how the torque profile of the first nonlinear spring (Figure 5(a)) is modified when we change the parameter λ_1. The greater the value of λ_1, the more linear the torque profile. The coefficients λ_i should be tuned so that the torque profiles are smooth enough for the springs to be realizable technically. In Figure 7, we show a spring torque profile designed with local stiffness constraints. We imposed the stiffness of the torque profile shown in Figure 5(a) to be strictly positive (superior to 0.05 Nm/rad) at the positions p_1–p_4 (see Table 1). The red dash-dotted tangents show the active constraints and the blue dotted tangents show the inactive constraints. With this method, we ensure that the stiffness of the spring is positive where the manipulator has to keep a static position for a while.

6. Technical realization of the nonlinear springs

The spring synthesized in Section 5 are nonlinear and exhibit negative stiffness on a part of their displacement range. Since these springs do not correspond to any off-the-shelf spring, we propose to realize them with the mechanism presented in [36]. This mechanism consists in a linear spring connected to a cable wound around a noncircular spool whose shape is calculated so that the mechanism behaves as a nonlinear rotational spring with the prescribed torque profile. The spring mechanism is attached to each joint as shown in Figure 8.

Since this mechanism can only handle positive torques, the nonlinear springs are realized by the antagonistic action of two cable-spool mechanisms. The first one realizes the torque profiles of Figures 9(a), 10(a), and 11(a)

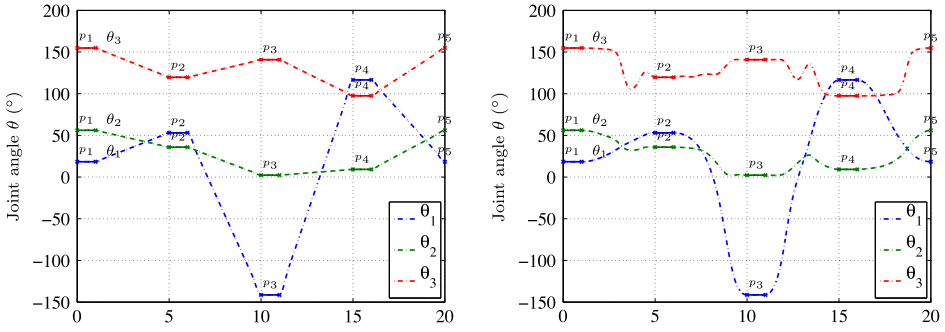


Figure 3. Trajectory of the manipulator (joint variables).

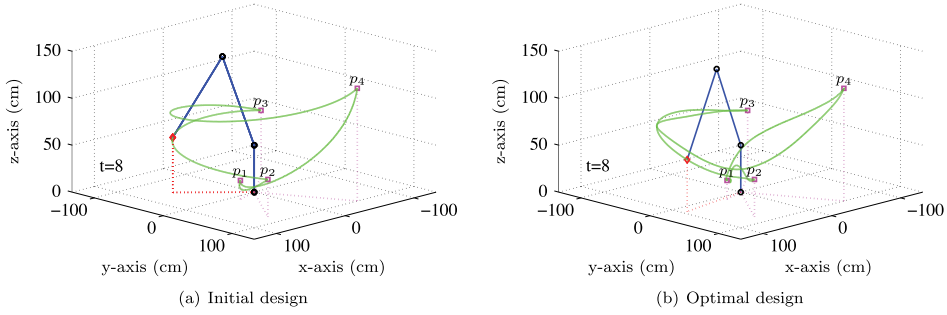


Figure 4. Path of end effector.

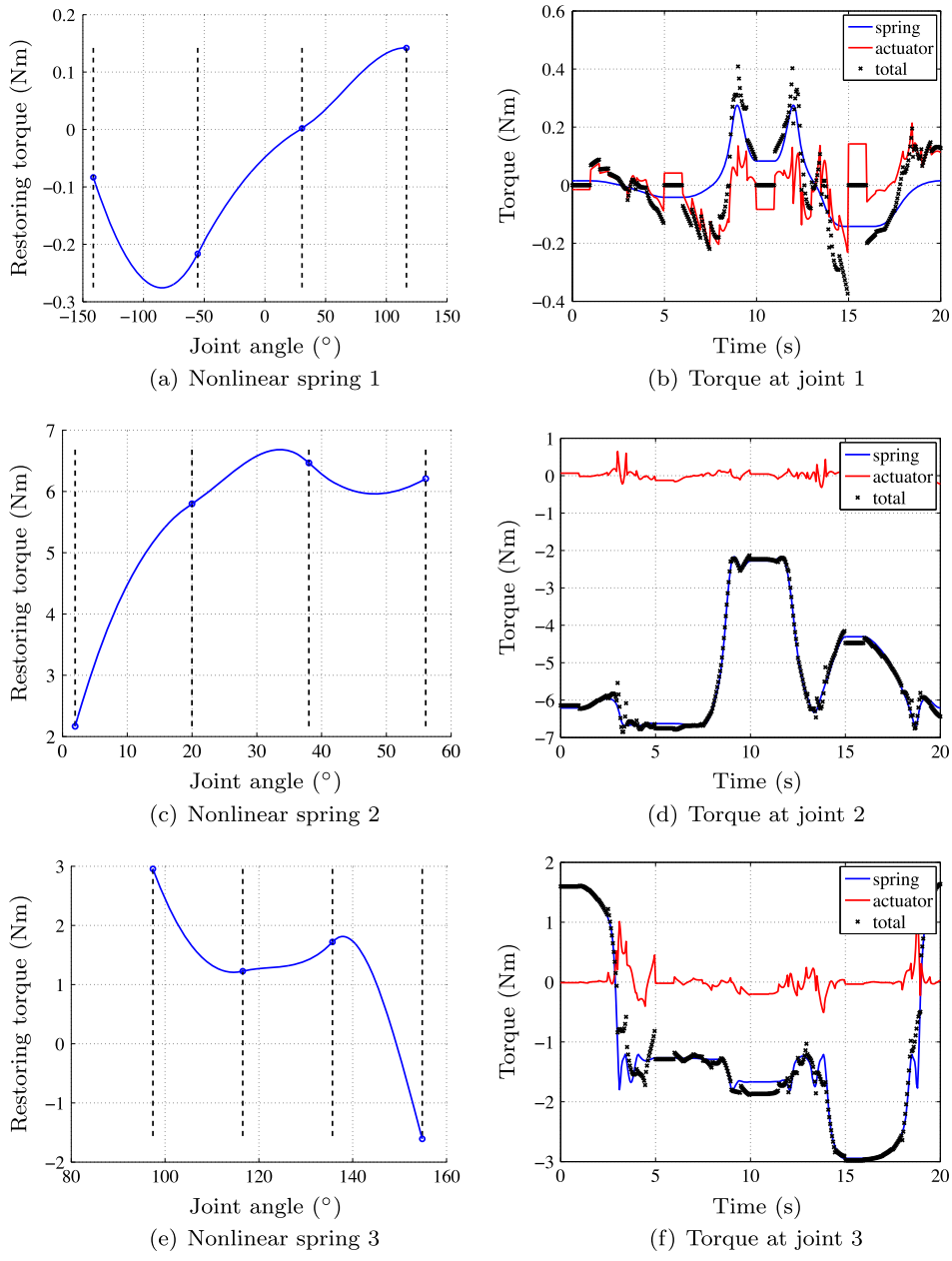


Figure 5. Spring torque profiles and joint torques.

Table 2. Comparison of average absolute actuator torque for each joint.

	No spring			Optimal linear spring			Optimal nonlinear spring		
$\frac{1}{T}\int_0^T \lvert u(t)\rvert \, \mathrm{d}t$ (Nm)	u_1	u_2	u_3	u_1	u_2	u_3	u_1	u_2	u_3
Initial trajectory	0.139	5.16	1.58	0.159	0.647	0.642	0.153	0.254	0.467
Optimal trajectory	0.0152	2.12	0.748	0.127	0.316	0.271	0.0907	0.102	0.109

calculated by shifting the torque profiles of Section 5 vertically so that the torque is strictly positive. A reduction ratio is introduced between the joint and the spool to adjust the rotation range of the spool. The second mechanism realizes a constant spring, so that the antagonistic action of the two mechanisms achieves the torque profiles of Section 5. The shape of the spool synthesizing the shifted torque profiles are shown in Figures 9(b), 10(b), and 11(b). See [36] for the design and realization of the constant springs.

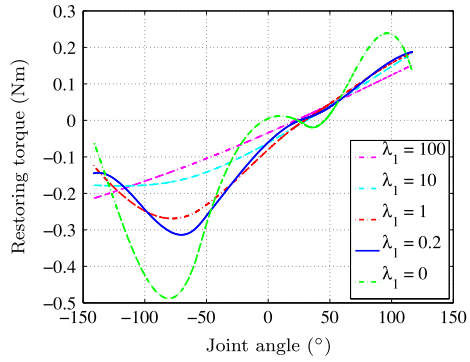


Figure 6. Torque profile of the first nonlinear spring (Figure 5(a)) for different values of λ_1. By changing the parameter λ_1, we can adjust the nonlinearity of the spring. The solid line, corresponding to $\lambda_1 = 0.2$, is the same torque profile as in Figure 5(a).

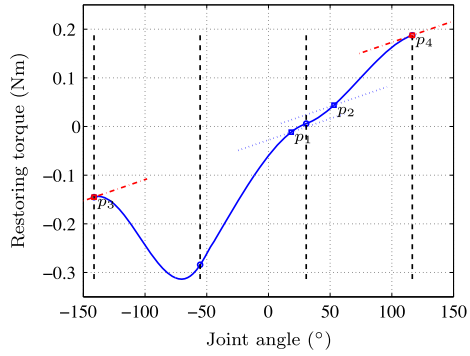


Figure 7. Imposing local constraints on the stiffness of the first nonlinear spring (Figure 5(a)). Red dash-dotted tangents show the active constraints and blue dotted tangents show the inactive constraints.

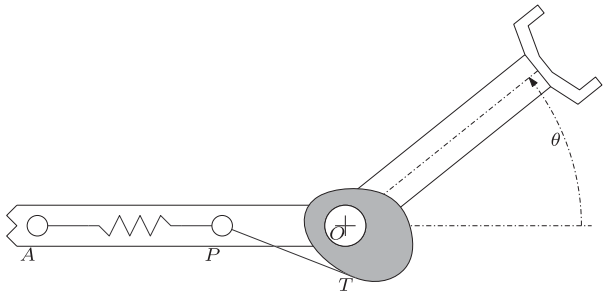


Figure 8. Nonlinear spring mechanism. The spool is rigidly attached to the upper link in O and the tip of the linear spring is attached to the lower link in A.

As explained in [36], the size of the spool mechanism can be scaled up or down by changing several design parameters such as the stiffness of the linear spring. Therefore, we can freely adjust the size of the spool mechanism to adapt it to the size of the robot. As an example, in Figure 6 of [36], we showed that a same nonlinear spring can be synthesized by several noncircular spools which maximal radius range from 93 mm down to 19 mm.

7. Discussion

An important point of the methodology proposed in this paper is the use of an Hermite interpolation to parameterize the torque profiles of the nonlinear springs. As a result, the cost function is a quadratic in the spring parameters, which makes it possible to express the optimal spring parameters as a function of the trajectory parameters. Furthermore, as mentioned in Section 3, this method automatically gives continuity of the torque and its derivative at the nodes.

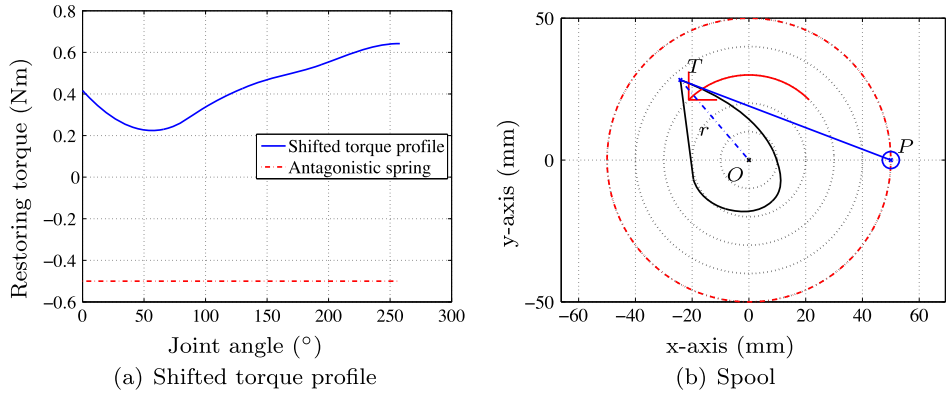
The continuity of the torque and its derivative, though, are not mandatory from a physical point of view since it is possible to build springs mechanisms which exhibit (discontinuous) piecewise C^1 functions. Theoretically, piecewise C^1 functions might lead to better performances since the set of C^1 functions is included in the set of piecewise C^1 functions. However, the technical realization of springs with discontinuous force profiles requires specific mechanisms, so the trade-off between performances and simplicity of realization of the springs should be considered. What is more, the heavier the mechanism used to realize a nonlinear spring, the more it is likely to impact the dynamics of the robot and possibly degrade the performances.

In Section 4.2, we showed that the springs can be designed more or less linear by adding to the cost function a term quadratic in the spring parameters. As shown in Table 2, nonlinear springs lead to better performances, but they have the drawback of being more complicated to realize than linear springs. There is a trade-off between performances and simplicity of realization of the springs.

The results of this research aim at improving the energy efficiency of robots working on production lines. Therefore, we consider a robot that always repeats a same trajectory to achieve a task such as picking or assembling. If we want to change the task of the robot, we first calculate the optimal trajectory and optimal nonlinear springs related to this new task, then replace the nonlinear spring mechanisms of each joint with the new ones. Since the spring mechanisms work in parallel with the actuators of the joints, they can be easily replaced without changing the structure of the robot itself.

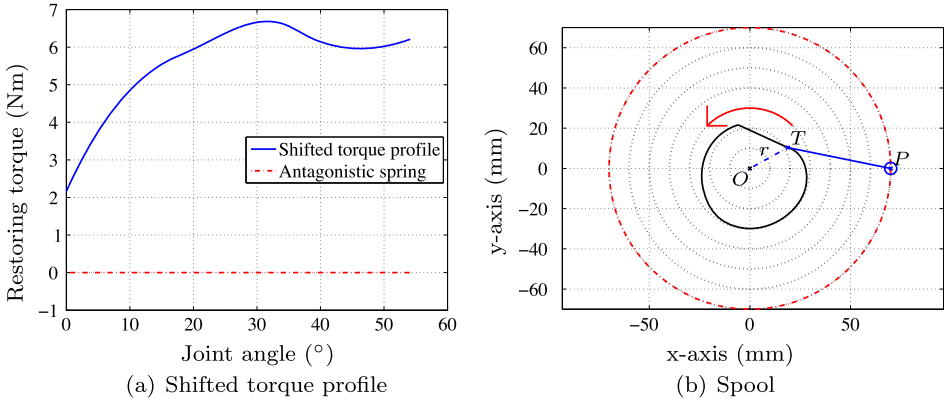
8. Conclusions

In this paper, we proposed a method to simultaneously design the trajectory of a robot and the torque profiles of nonlinear springs acting in parallel with the actuators in order to minimize the average actuator torque. We first expressed the trajectory and the torque profiles of the



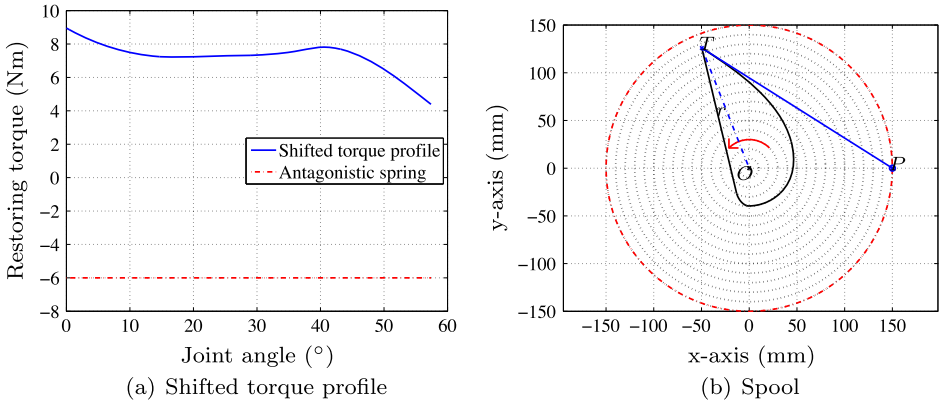
(a) Shifted torque profile (b) Spool

Figure 9. Realization of spring 1 with a noncircular spool mechanism.



(a) Shifted torque profile (b) Spool

Figure 10. Realization of spring 2 with a noncircular spool mechanism.



(a) Shifted torque profile (b) Spool

Figure 11. Realization of spring 3 with a noncircular spool mechanism.

springs using a third-order Hermite interpolation, and used the interpolation points as the design parameters. We showed that the cost function is a quadratic function of the spring parameters, and expressed the optimal spring parameters as a function of the trajectory parameters. We defined a *cost function at optimal spring design* which depends only on the trajectory parameters. We used an SQP algorithm to find an approximate solution of the optimal trajectory. We showed that the nonlinearity of the springs can be adjusted by adding to the cost function a weighting matrix G, and that it is possible to add local constraints on the stiffness of the springs. As an example, we proposed the optimal design of a three-DOF serial manipulator. Several constraints were set on the position of the manipulator for given time domains, while the algorithm was let free to optimize the unconstrained parts

of the trajectory. The results showed that a significant part of the overall joint torque was provided by the nonlinear springs resulting in a decrease in the average actuator torque required to drive the robot. We showed that the nonlinear springs calculated in this paper were technically realizable using a noncircular cable spool mechanism.

Acknowledgments

This research is supported by the Research on Macro/Micro Modeling of Human Behavior in the Swarm and its Control under the Core Research for Evolutional Science and Technology (CREST) Program (research area: Advanced Integrated Sensing Technologies), Japan Science and Technology Agency (JST).

Notes

1. In this paper, we consider rotational springs and actuators. However, the design methodology is the same for prismatic springs and actuators.
2. We use $s_{i,j}\Delta q_i$ instead of $s_{i,j}$ so that all elements of y_i have the same dimension.
3. K can become ill-conditioned if N is large (i.e. when trying to optimize spring torque profiles with many interpolation nodes). However, since we want to design springs which can be realized technically, we usually use a small N (smaller than 8) in order to obtain smooth torque profiles.

Notes on contributors

Nicolas Schmit was born in Reims, France, in 1985. He was graduated from the Ecole Polytechnique, Paris, and from the Institut Superieur de l'Aeronautique et de l'Espace (Supaero), Toulouse (France), in 2009. He is currently pursuing the PhD degree in mechanical engineering at Tokyo Institute of Technology, Tokyo. His dissertation research is on the optimal design of nonlinear stiffness of robotic mechanisms. He did his graduation internship at Thales Alenia Space's Research Department, Cannes (France), from April 2009 to September 2009. His research was on the robust control of telecom satellites subject to unstable fuel sloshing phenomena. He currently holds a fellowship from the Japanese Government (MONBUKAGAKUSHO: MEXT).

Masafumi Okada received his ME degree and PhD in Applied System Science from Kyoto University in 1994 and 1996, respectively. In 1997, he joined the Department of Mechano-Informatics in the University of Tokyo, and he is currently an associate professor in the Department of Mechanical Sciences and Engineering, Tokyo Institute of Technology. His interests include an attractor-based robot control, the mechanical design with nonlinear stiffness, and the amenity design for human environments.

References

[1] Jafari A, Tsagarakis NG, Vanderborght B, Caldwell DG. A novel actuator with adjustable stiffness (AwAS). In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; 2010; Taipei. Taiwan. p. 4201–4206.

[2] Schauss T, Scheint M, Sobotka M, Seiberl W, Buss M. Effects of compliant ankles on bipedal locomotion. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2009; Kobe, Japan. p. 2761–2766.

[3] Yamaguchi Y, Nishino D, Takanishi A. Realization of dynamic biped walking varying joint stiffness using antagonistic driven joints. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 3; 1998; Leuven, Belgium. p. 2022–2029.

[4] Walsh GJ, Streit DA, Gilmore BJ. Spatial spring equilibrator theory. Mech. Mach. Theory. 1991;26(2):155–170.

[5] Jo DY, Haug EJ, Beck RR. Optimization of force balancing mechanisms. Tech. Rep. A101421. Iowa City, IA 52242. USA: College of Engineering, The University of Iowa; 1982.

[6] Streit DA, Gilmore BJ. Perfect spring equilibrators for rotatable bodies. J. Mech. Transm. Autom. Des. 1989; 111:451–458.

[7] Ulrich N, Kumar V. Passive mechanical gravity compensation for robot manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 2; 1991; Sacramento, CA. p. 1536–1541.

[8] Endo G, Yamada H, Yajima A, Ogata M, Hirose S. A passive weight compensation mechanism with a non-circular pulley and a spring. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2010; Anchorage, Alaska. p. 3843–3848.

[9] McN R, Alexander. Three uses for springs in legged locomotion. Int. J. Robot. Res. 1990;9(2):53–61.

[10] Farrell KD, Chevallereau C, Westervelt ER. Energetic effects of adding springs at the passive ankles of a walking biped robot. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2007; Roma, Italy. p. 3591–3596.

[11] Ackerman J, Seipel J. Energetics of bio-inspired legged robot locomotion with elastically-suspended loads. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 11; 2011; San Francisco, CA. p. 203–208.

[12] Uemura M, Kawamura S. Resonance-based motion control method for mmulti-joint robot through combining stiffness adaptation and iterative learning control. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2009; Kobe, Japan. p. 1543–1548.

[13] Bigge B, Harvey IR. Programmable springs: developing actuators with programmable compliance for autonomous robots. Robot. Autonom. Syst. 2007;55(9):728–734.

[14] Pratt GA, Williamson MM. Series elastic actuators. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems. Vol. 1; 1995; Washington, DC. p. 399–406.

[15] Laurin-Kovitz KF, Colgate JE, Carnes SDR. Design of components for programmable passive impedance. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 2; 1991; Sacramento, CA. p. 1476–1481.

[16] Migliore SA, Brown EA, DeWeerth SP. Biologically inspired joint stiffness control. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2005; Barcelona, Spain. p. 4508–4513.

[17] English C, Russell D. Implementation of variable joint stiffness through antagonistic actuation using rolamite springs. Mech. Mach. Theory. 1999;34(1):27–40.

[18] Bicchi A, Tonietti G. Design, realization and control of soft robot arms for intrinsically safe interaction with humans. Tech. Rep. Centro Interdipartimentale di Ricerca 'E. Piaggio', Universita di Pisa, Italia; 2002.

[19] Koganezawa K, Watanabe Y, Shimizu N. Antagonistic muscle-like actuator and its application to multi-d.o.f. forearm prosthesis. Adv. Robot. 1997;12:771–789.

[20] Koganezawa K. Mechanical stiffness control for antagonistically driven joints. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; 2005; Edmonton, Alberta, Canada. p. 1544–1551.

[21] Tonietti G, Schiavi R, Bicchi A. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2005; Barcelona, Spain. p. 526–531.

[22] Huang TH, Kuan JY, Huang HP. Design of a new variable stiffness actuator and application for assistive exercise control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 11; 2011; San Francisco, CA. p. 372–377.

[23] Schiavi R, Grioli G, Sen S, Bicchi A. VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2008; Pasadena, CA. p. 2171–2176.

[24] Hurst JW, Chestnutt J, Rizzi A. An actuator with mechanically adjustable series compliance. Tech. Rep. CMU-RI-TR-04–24. Pittsburgh (PA): Robotics Institute, Carnegie Mellon University; 2004.

[25] Hurst JW, Chestnutt JE, Rizzi AA. An actuator with physically variable stiffness for highly dynamic legged locomotion. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 5; 2004; New Orleans, LA. p. 4662–4667.

[26] Wolf S, Hirzinger G. A new variable stiffness design: matching requirements of the next robot generation. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2008; Pasadena, CA. p. 1741–1746.

[27] Van Ham R, Vanderborght B, Van Damme M, Verrelst B, Lefeber D. MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: design and implementation in a biped robot. Robot. Autonom. Syst. 2007;55(10):761–768.

[28] Morita T, Sugano S. Design and development of a new robot joint using a mechanical impedance adjuster. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 3; 1995; Nagoya, Aichi, Japan. p. 2469–2475.

[29] Tsagarakis NG, Sardellitti I, Caldwell DG. A new varia stiffness actuator (CompAct-VSA): design and modelling. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 11; 2011; San Francisco, CA. p. 378–383.

[30] Jafari A, Tsagarakis NG, Caldwell DG. Exploiting natural dynamics for energy minimization using an Actuator with Adjustable Stiffness (AwAS). In: Proceedings of the IEEE International Conference on Robotics and Automation; 2011; Shanghai, China. p. 4632–4637.

[31] Mombaur KD, Longman RW, Bock HG, Schloder JP. Stable one-legged hopping without feedback and with a point foot. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 4; 2002; Washington, DC. p. 3978–3983.

[32] Mombaur KD, Longman RW, Bock HG, Schloder JP. Open-loop stable running. Robotica. 2005;23:21–33.

[33] Mombaur KD, Bock HG, Schloder JP, Longman RW. Open-loop stable solutions of periodic optimal control problems in robotics. J. Appl. Mech./Zeitschrift fr Angewandte Mathematik und Mechanik. 2005;85.7:499–515.

[34] Duindam V, Stramigioli S. Optimization of mass and stiffness distribution for efficient bipedal walking. In: Proceedings of the International Symposium on Nonlinear Theory and Its Applications. Electronic proceedings; 2005.

[35] Nakanishi J, Rawlik K, Vijayakumar S. Stiffness and temporal optimization in periodic movements: an optimal control approach. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 11; 2011 Sep; San Francisco, CA. p. 718–724.

[36] Schmit N, Okada M. Design and realization of a non-circular cable spool to synthesize a nonlinear rotational spring. J. Adv. Robotic. 2012;26:235–252.

[37] Kuhn HW, Tucker AW. Nonlinear programming. In: Proc. Second Berkeley Symp. on Math. Statist. and Prob. Univ. of Calif. Press; 1951; Berkeley, CA. p. 481–492.

[38] Barclay A, Gill PE, Rosen JB. SQP methods and their application to numerical optimal control. In: Werner H, Bittner SL, Kltzler R, editors. Variational calculus, optimal control, and applications. Vol. 124. International Series of Numerical Mathematics. Trassenheide, Germany: Birkhuser; 1998. p. 207–222.