# Optimal trajectory planning for nonlinear systems: robust and constrained solution

P. Boscariol*, A. Gasparetto

*Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica*
*University of Udine, Italy*
*Via delle Scienze 208, 33100 Udine*
* *e-mail address: paolo.boscariol@uniud.it*

**Abstract**

This paper presents a solution to the problem of generating constrained robust trajectory planning for nonlinear mechatronic systems. By using an indirect variational solution method, the necessary optimality conditions deriving from the Pontryagin's minimum principle are imposed, and lead to a differential Two-Point Boundary Value Problem (TPBVP); numerical solution of the latter is accomplished by means of collocation techniques. The robustness to parametric mismatches is obtained trough the use of sensitivity functions, while a hard constraint on actuator effort is obtained using a smoothing technique. Numerical results shows that the robustness can be greatly improved, and that the inclusion of constraints on actuator effort does not affect it.

## 1 Introduction

The targets of robot design are almost always: high speed, precise operations, safety, and high payload over weight ratio. For this reason an increasing number of robots are built with lightweight structures, presenting therefore some sort of structural flexibility [1, 2]. Moreover many rigid link robots, like the ones commonly used in industrial applications, present some joint flexibility [3]. The operation of such manipulators requires the use of accurate and clever techniques both in the design of closed-loop controllers, both in the design of smooth trajectories. The first approach is aimed at ensuring that the robot follows a pre-defined trajectory, while the latter concerns the definition of the trajectory. The whole problem is even more evident and hard to solve when the robot presents some structural flexibility, since the presence of oscillations even after the completion of the task severely reduces the operability of the robot [4, 5, 6]. A large number of works have been developed to find solutions to the aforementioned problem by generating smooth trajectories, as reported in the works [7, 8, 9]. A main distinction can be made among trajectory planning algorithms separating model-free and model-based approaches. The main advantage of the first approach is that the resulting trajectory can be adapted to several robots and, moreover, the knowledge of the dynamics of the system is not needed. This is a very useful feature, since this kind of expertises is not often found in industry. On the other hand, model-based approach requires the knowledge of the dynamics of the robot for which the trajectory is planned, but generally they prove to be more accurate and usually lead to higher levels of performance. Since most model-based approaches are based on optimal control theory, they provide a limited robustness to model-plant mismatches, as emphasized in [10]. This means that a trajectory that is optimal in the nominal case, is far from the optimal solution if applied to a perturbed plant. This concept of robust performance is widespread in control applications [11], while, to the best of Authors' knowledge, is quite unexplored as far as trajectory planning algorithms are concerned.

Model-free approaches are often based on geometrical approaches [12], since the trajectory is defined as a sequence of polynomial functions [13], splines [9] or NURBS [14]. Other possible ways



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

to generate a trajectory is to use filters [15].

Model-based approach have been studied in a large number of works, with applications to basically every kind of robot. The solution of the planning problem has been investigated for mobile robots in papers such as [16]. Flexible joint robot have been considered in [17, 18]. Also the design of trajectories for Flexible-Link Manipulators (FLM) have been studied quite extensively [19, 20, 21]. Approaches based on the definition and the solution of Two-Point Boundary Value Problems (TPBVP) have been developed, among others, in [20, 22]. In these works a point-to-point trajectory is computed by solving a constrained optimization problem, by imposing that the trajectory must connect the two boundary points while respecting the robot dynamics.

Also in this work a solution to the problem of computation of constrained point-to-point trajectories is analyzed, but with a particular focus on the robustness of the solution to parametric uncertainties. The topic of robustness have been extensively studied in the area of closed-loop control. Typical references from an extensive literature include, among others [23, 24, 25], but to the best of authors' knowledge, there are very few works that specifically focus on robust trajectory planning algorithms.

One example is [26], in which robustness is achieved by introducing in the fitness function a term of Gaussian cumulative noise. The work by Shin [27] focuses on the definition of robot trajectories by taking into account the uncertainties brought by payload variations trough the change of bounds on joint torques. Other interesting approaches to robust trajectory planning are currently available as solutions to the problem of robust optimization for dynamic systems: an extensive overview of this problem is available in [28]. The aim of this paper is to propose a method for planning a trajectory which is based on two-point boundary value problems and on the concept of desentization. Sensitivity function have been used in [29, 30, 31] to improve the robustness of closed-loop optimal controllers. The design of such controllers is done analytically, since a solution of this kind can be found is the plant taken into consideration is linear. It must be highlighted that the method proposed here applies to nonlinear plants, therefore it greatly enhances the field of application of the method presented in [29, 30, 31].

The inclusion of constraints on actuator action also plays an important role in most real-world applications, in which the actuator's capabilities must be exploited up to their full potential and without violating the actuator's safe operating area. Among the techniques presented in the papers [29, 30, 31],the only one that allows to include constraints is [31], but the method used to limit the control action in that paper cannot be directly applied to the class of problems considered in this work.

Moreover the feedforward techniques used in this paper do not require any additional sensor. Thus, they are more economical then closed-loop strategies, for the control of robotic manipulators performing repetitive tasks. Also their off-line nature allows overcoming many difficulties, such as highly non-linear dynamics and system or actuation constraints. The outcome of the approach presented in this paper is a position profile for the system to be operated that can be used by most industrial PLC-based controller.

The capabilities of this novel approach are shown by means of a benchmark problem, i.e. a robotic joint system with nonlinear elasticity, as a representative of the class of single d.o.f. elastic systems with a nonlinear behavior. Future extensions of this work will include the application of the technique to actual robotic systems, to provide an experimental validation to the method proposed here. Some possible application could include for example and hoverhead crane, as in the paper [32], or a serial robot with flexible couplings, as in [3]. In the first case the nonlinearity is present because of the pendulum-like dynamics, while in the second case the nonlinearities are due to nonlinear elasticity of coupling, robot kinematics, gravity and friction. The application of the proposed technique would allow to show also experimentally that the proposed method can improve the operability of the robot when parametric mismatches between the actual robot dynamics and the theoretical model are present.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

# 2 Theoretical background

The target here is to develop an optimal trajectory for a mechatronic system. We consider here point-to-point trajectory optimization problems, in which only the initial and final end-effector positions are given, and the manipulator is free to move between them. Therefore both the path and the trajectory are subject to optimization, and they are selected with the aim of minimizing a cost functional. Such cost may depend on execution time, actuator effort, jerks (or torque rates), or a combination of these variables. First of all, let us define the optimization problem that we want to solve. Given a dynamic system, that might be linear or nonlinear, described by a differential equation in the form:

$$\dot{\mathbf{x}} = \Omega(\mathbf{x}, t, \mathbf{u}) \tag{1}$$

in which $\mathbf{x}$ is the state variable of the system, and $\mathbf{u}$ is the control vector. If we choose a cost function:

$$J = f(\mathbf{x}, t, \mathbf{u}) \tag{2}$$

the following optimization problem can be stated:

$$\begin{cases} \min J(\mathbf{x}(t), t, \mathbf{u}) = \min \int\limits_{t_0}^{t_f} f(\mathbf{x}, t, \mathbf{u}) dt \\ s.to. \\ \mathbf{x}(t_0) = \alpha \\ \mathbf{x}(t_f) = \beta \\ \dot{\mathbf{x}}(t) = \Omega(\mathbf{x}(t), t, \mathbf{u}) \end{cases} \tag{3}$$

By solving this optimization problem, a trajectory for the state vector $\mathbf{x}$ is found so that the cost function $J$ is minimized. The trajectory is constrained to respect the dynamics of the system $\Omega(\mathbf{x}, t, u)$ and the value of $\mathbf{x}$ at the initial ($t = t_0$) and final ($t = t_f$) time. A solution of the optimization problem in equation (3) can be found using the calculus of variations and Pontryagin's Minimum Principle (PMP) [33].

First of all, the Hamiltonian of the system must be defined as:

$$\mathcal{H} = f + \Lambda^T \Omega(\mathbf{x}(t), t, u) \tag{4}$$

in which $\Lambda = [\lambda_1, \ldots, \lambda_N]^T$ is the vector of Lagrangian multipliers, which has the same size of the state vector $\mathbf{x}$. The necessary conditions for finding a minimum of the problem in Eq. (3) are:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0 \tag{5}$$

$$\dot{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \Lambda} \tag{6}$$

$$\dot{\Lambda} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \tag{7}$$

The above conditions can be put in a single system that makes the computation straightforward. By defining $\mathbf{u}^*$ the solution of equation (5), $\mathcal{H}^*(\mathbf{x}, t)$ is the Hamiltonian in which $\mathbf{u}$ has been substituted with $\mathbf{u}^*$. $\mathcal{H}^*$ is called the minimizing Hamiltonian. A new system of ordinary differential equation can be defined as:

$$\dot{\mathbf{y}}^* = \begin{bmatrix} \dfrac{\partial \mathcal{H}^*}{\partial \Lambda} \\ -\dfrac{\partial \mathcal{H}^*}{\partial \mathbf{x}} \end{bmatrix} \tag{8}$$



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

The new state vector $\mathbf{y}$ is obtained by augmenting the original state vector $\mathbf{x}$ with the vector of Lagrangian multipliers: $\mathbf{y} = [\mathbf{x}, \Lambda]^T$. Among the infinite possible trajectories of the dynamic system in equation (8), we are interested in finding the one that obeys to the boundary conditions $\mathbf{y}(t_0) = \alpha$ and $\mathbf{y}(t_f) = \beta$.

A solution to this problem, that is basically a TPBVP (Two-Point Boundary Value Problem), could theoretically be found in closed form. In many cases, however, it is solved numerically, given the difficulty of finding an analytic solution. Collocation method [22, 34] and shooting method [35] are often used for this task.

# 3 Formulation of the robust trajectory planning algorithm

The solution to the problem presented in the previous section works very well when the dynamic model used for planning the trajectory can reproduce faithfully the actual dynamics of the real system. This does not happen in all situations, given the difficulty of describing a complex plant with a reasonably simple model. Moreover, sometimes it is not even possible to describe the dynamics of the plant with just a single model. A common situation is when a robot is driving a payload that changes, as in a pick & place operation. As the mass carried by the robotic manipulator changes, also its dynamic model is altered. Quite often it happens also that nonlinearities are neglected during the modeling phase: in this case the trajectory planning algorithm and the control loop are required to compensate for the model-plant mismatches.

A possible solution to the problem of robustness of trajectory planning algorithms stated above relies in the use of the desensitization technique. Let us take into consideration a function $g(x, t, \mu)$ which is continuous in $(x, t, \mu)$ and has continuous first partial derivatives with respect to $x$ and the parameter $\mu$ for all values of $(x, t, \mu)$. Moreover, $g$ should have unique solution in the time interval $[t_0, t_f]$. The differential state equation is:

$$\dot{x} = g(x, t, \mu) \quad \text{with} \quad x(t_0) = x_0 \tag{9}$$

It is known that for all $\mu$ close to the nominal value $\mu_0$, equation (9) has a unique solution $g(t, \mu)$ that is close to the nominal solution $g(t, \mu_0)$. Since $g(x, t, \mu)$ has continuous derivatives with respect to $x$ and $\mu$, it is implied that the solution $x(t, \mu)$ is differentiable with respect to $\mu$ near $\mu_0$. The partial derivative of $x(t, \mu)$, i.e. $x_\mu$ is:

$$x_\mu(t, \mu) = \int_{t_0}^{t} \left[ \frac{\partial g}{\partial x}(s, x(s, \mu) x_\mu(s, \mu)) + \frac{\partial g}{\partial \mu}(s, x(s, \mu), \mu) \right] ds \tag{10}$$

and $\partial x_0 / \partial \mu = 0$, since $x_0$ is independent of $\mu$. By taking the derivative with respect to $t$, one obtains:

$$\frac{\partial x_\mu(t, \mu)}{\partial t} = \left. \frac{\partial g(x, t, \mu)}{\partial x} \right|_{x = x(t, \mu)} x_\mu(t, \mu) + \left. \frac{\partial g(x, t, \mu)}{\partial \mu} \right|_{x = x(t, \mu)}, \quad \text{with} \quad x_\mu(t_0, \mu) = 0 \tag{11}$$

For $\mu = \mu_0$ the right-hand side of Eq. (11) depends only on the nominal solution $x(t, \mu)$. $S(t) = x_\mu(t, \mu_0)$ is the unique solution of the differential equation:

$$\dot{S}(t) = \left. \frac{\partial g(x, t, \mu_0)}{\partial x} \right|_{x = x(t, \mu_0)} S(t, \mu_0) + \left. \frac{\partial g(x, t, \mu_0)}{\partial \mu} \right|_{x = x(t, \mu_0)}, \quad \text{with} \quad S(t_0) = 0 \tag{12}$$

Equation (12) is the sensitivity equation, while its solution, i.e. $S(t)$, is the sensitivity function. It is used in this context since it allows to estimate the effect of the variation of the parameter $\mu$ on the solution of the differential state equation (9). By means of the sensitivity functions corresponding to Eq. (1), the solution of the optimization problem in Eq. (3), can be formulated in order to include the variation of the parameter $\mu \in [\mu_{min}, \mu_{max}]$:



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

$$S(\mathbf{x}, t_0, \mu) = 0, \quad \text{and} \quad S(\mathbf{x}, t_f, \mu) = 0 \tag{13}$$

By doing this, the robustness of the optimal solution $x^*(\mathbf{x}, t, u, \mu)$ with respect to the variation of $\mu$ from the nominal value $\mu_0$ can be increased. This technique is usually called desensitization. The effectiveness of desensitization has been shown numerically [29, 30, 31] for the design of closed-loop control systems. However this approach has found its use only taking into account linear models [29, 30, 31]. The aim of this paper is to show that it can be used also for nonlinear models, and that it effectively improves the robustness to parametric uncertainties in model-based trajectory planning algorithms.

The application of sensitivity function can be used to extend the optimization problem in (3) to the following formulation:

$$\begin{cases} \min J(\mathbf{x}(t), t, \mathbf{u}) = \min \int\limits_{t_0}^{t_f} f(\mathbf{x}, \mathbf{s}, t, u) dt \\ s.to. \\ \mathbf{x}(t_0) = \alpha \\ \mathbf{x}(t_f) = \beta \\ S(t_0) = 0 \\ S(t_f) = 0 \\ \dot{\mathbf{x}}(t) = \Omega(\mathbf{s}(t), t, u) \\ \dot{S}(t) = \dfrac{\partial \Omega}{\partial \mu}(\mathbf{x}(t), t, u) \end{cases} \tag{14}$$

The main difference between the problem in Eq. (3) and Eq. (14) is that in the latter the system state is constrained also to the dynamics of the sensitivity functions $\dot{S}(t)$. Additionally, also the initial and final value of the sensitivity function are forced to zero: $S(t_0) = 0$ and $S(t_f) = 0$. These constraints on sensitivity functions are responsible for the robust behavior of the system.

It is worthwhile to point out that the dynamic equation of the sensitivity functions that appear in Eq. (14) can be also computed automatically using a Computer Algebra System,such as Sage [36] or Axiom [37], thus making the formulation of the TPBVP completely automatic once $\Omega$, $\alpha$, $\beta$, $t_f$ and $\mu$ are chosen. The same numerical method used for solving the problem in Eq. (3) can be used here. In the next section, also the problem of including hard constraints will be dealt with.

# 4 Numerical results: constrained trajectory for the nominal system

The test case under consideration here is an elastic joint with a nonlinear spring characteristic, as in figure 1. No damping elements are introduced. The nonlinear characteristic of the spring is described by the elastic force $F(\Delta q)$:

$$F(\Delta q) = k(\Delta q + \Delta q^3) \tag{15}$$

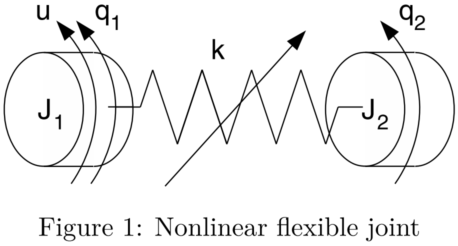


Figure 1: Nonlinear flexible joint



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

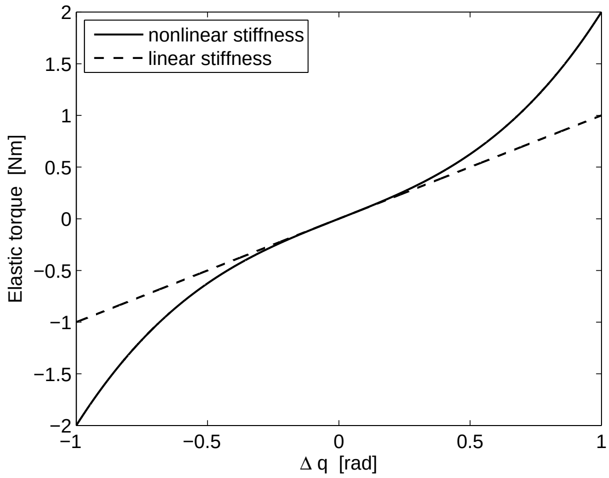


Figure 2: Nonlinear spring elastic characteristic, $k = 1$

being $\Delta q$ the relative displacement of the two inertias $J_1$ and $J_2$. The angular position of the two rotating masses are, respectively, $q_1$ and $q_2$. The magnitude of the elastic force is shown in figure 2 for $k = 1$. This description of the elastic force is commonly used to represent the effects of nonlinearities introduced by flexible coupling between robot links. For example, this model is used in [38, 3, 39, 40] and [41] to represent the elasticity of gear boxes in industrial robots. In the work [42] also the validation of this model with reference to an ABB IRB6600 industrial robot is provided. This model can also be used to describe the dynamics of a multi-link flexible robot using the approach proposed by Spong in the work [43]. The choice of this benchmark problem has been made to provide a clear explanation of all the steps needed to implement the proposed procedure, and also to make the results easy to understand. Anyway, is should be made clear that the method explained in this paper can be applied to more complex applications, also involving dynamic systems represented by a larger state vectors. While the computation of the augmented state vector can be made for any state vector size, the limit on the dimensions of the problem is dominated by the difficulty of the numerical solution of the optimization problem. Therefore the limits of numerical methods used for the solution of boundary value problems also applies here [34, 44].

If $u$ is the external torque applied to the mass $J_1$, the dynamics of the system is described by the two second-order differential equations:

$$\begin{aligned} J_1\ddot{q}_1 &= k(q_2 - q_1) + k(q_2 - q_1)^3 + u \\ J_2\ddot{q}_2 &= -k(q_2 - q_1) - k(q_2 - q_1)^3 \end{aligned} \tag{16}$$

Let us consider, for sake of simplicity, the case in which $J_1 = J_2 = 1$; this has not necessarily to be case, but this assumption allows to reduce the complexity of the equations. Also the mass speeds will be indicated with $\omega_1$ or $\dot{q}_1$, and with $\omega_2$ or $\dot{q}_2$. With this choice the ODE system in first order form that will be used to compute the optimal trajectory is:

$$\dot{\mathbf{x}} = \begin{bmatrix} k(q_2 - q_1) + k(q_2 - q_1)^3 + u \\ -k(q_2 - q_1) - k(q_2 - q_1)^3 \\ \omega_1 \\ \omega_2 \end{bmatrix} \tag{17}$$

The application of Pontryagin's minimum principle requires for the Hamiltionian to be differentiable in time with continuous derivatives in $\mathbf{x}$ and $\Lambda$, therefore hard constraints in the form of a saturation function cannot be included directly. If the aim is to impose constraints on the value of the control action $u$, a smoothing technique can be used. In particular the method introduced by



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

Avvakumov et al. in [45] can be used here to impose a constraint on the amplitude of the control signal $u$. The saturation function:

$$sat(s,\gamma) = \begin{cases} s, & |s| < \gamma \\ \gamma sign(s), & |s| > \gamma \end{cases} \tag{18}$$

can be approximated by:

$$SAT(s,\gamma,\nu) = \frac{\gamma}{2}\left(\sqrt{\nu + \left(\frac{s}{\gamma}+1\right)^2} - \sqrt{\nu + \left(\frac{s}{\gamma}-1\right)^2}\right) \tag{19}$$

with $\nu$ some small positive number.

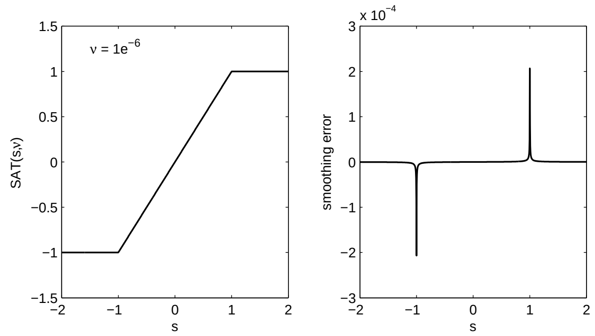


Figure 3: Smooth saturation function (left) evaluated for $\nu = 1e^{-6}$ and $\gamma = 1$, approximation error (right)

The Hamiltonian of the system, evaluated for the minimum effort solution, i.e. for $f = \frac{1}{2}u^2$ is:

$$\mathcal{H} = \lambda_3\omega_1 - \lambda_1(k(q_1-q_2) - u + k(q_1-q_2)^3) + \lambda_4 * \omega_2 + \lambda_2(k(q_1-q_2) + k(q_1-q_2)^3) + \frac{u^2}{2} \tag{20}$$

and therefore:

$$\frac{\partial \mathcal{H}}{\partial u} = u + \lambda_1 \tag{21}$$

From Eq. (21) the optimal control action is $u^* = -\lambda_1$, therefore the ODE system computed after the evaluation of the three Pontryagin's necessary conditions in equations (5,6,7) is:

$$\dot{\mathbf{y}}^* = \begin{bmatrix} -\lambda_1 - k(q_1-q_2) - k(q_1-q_2)^3 \\ k(q_1-q_2) + k(q_1-q_2)^3 \\ \omega_1 \\ \omega_2 \\ -\lambda_3 \\ -\lambda_4 \\ k(\lambda_1-\lambda_2)(3{q_1}^2 - 6q_1q_2 + 3{q_2}^2 + 1) \\ -k(\lambda_1-\lambda_2)(3{q_1}^2 - 6q_1q_2 + 3{q_2}^2 + 1) \end{bmatrix} \tag{22}$$

The problem above can also be modified in order to include a hard constraint on the variable $\lambda_1$: the formulation for the constrained problem can be obtained from Eq. (22) by direct substitution of $\lambda_1$ with $\frac{\gamma}{2}\left(\sqrt{\nu + \left(\frac{\lambda_1}{\gamma}+1\right)^2} - \sqrt{\nu + \left(\frac{\lambda_1}{\gamma}-1\right)^2}\right)$.

In this way the value of $\lambda_1$, i.e. of the torque applied to the firt mass, can be limited in



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

the range $u \in [-\gamma, \gamma]$. This constraint can be used to take into account the limitations of the actuation system. Therefore it can be seen that the use of a numerical approximation of the saturation function can be used to impose hard constraints on physical quantities of the system. However, $u$ is not the only variable whose value can be constrained: the same technique can be used to limit the excursion of any of the variables that belong to the augmented state vector $\mathbf{y}$.

In the following the results of the computation of the trajectory planning algorithm are shown. The boundary conditions are set in order to bring the two masses from the initial position $q_1(t = 0) = 0.1$ rad and $q_2(t = 0) = 0.1$ rad to $q_1(t = t_f) = q_2(t = t_f) = 0$ rad with initial and final speed equal to zero. Therefore a rest-to-rest motion is planned. In this case the total execution time is chosen to be $t_f = 2$ s. The values of $u$ is limited in the range $[-2.5, 2.5]$ Nm for the constrained solution.

The control action $u$ is for the unconstrained and constrained case is shown in figure 4. The planned trajectory is shown in Fig. 5. In particular, it can be seen that the inclusion of constraints allows to precisely limit the amplitude of $u$: as imposed in the definition of the optimization problem, the torque provided by the motor never exceeds the prescribed value of 2.5 Nm. This result has been obtained with $\nu = 1 \times 10^{-9}$. It has been verified numerically that lowering the value of $\nu$ does not improve the quality of the solution. The actual trajectories for the unconstrained and constrained solutions are similar to each other, as it can be seen in Figure 5. In the constrained case the first mass achieves a slightly higher speed than in the nominal case, which is a direct effect of the torque limitation.

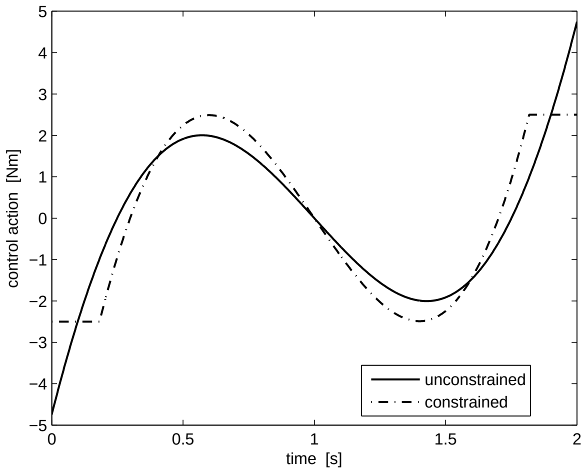


Figure 4: Control action $u$ for the unconstrained and constrained cases, nominal system



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

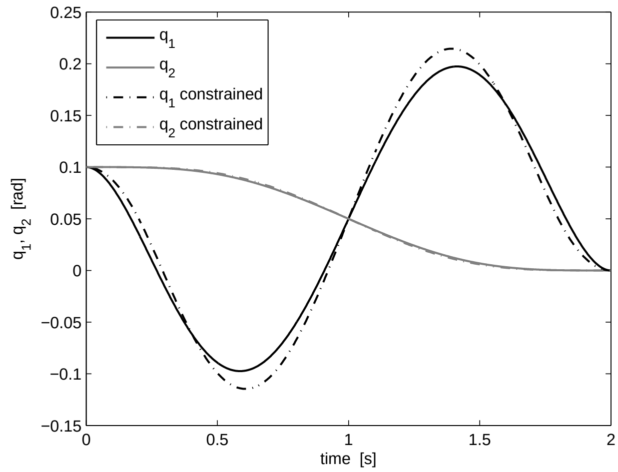


Figure 5: $q_1$ and $q_2$ for the unconstrained and constrained cases, nominal system

In order to verify that the computed trajectories are accurate, the control action $u(t)$ is fed directly to the dynamical model of Eq. (17) as in a feed-forward action. This test has the only purpose of evaluating the accuracy of the planned trajectory on the actual dynamic model, as this procedure allows to measure the residual energy of the system after the task completion: this measure will be later used to provide a numerical evaluation of the robustness of the planned trajectories. The resulting evolution of the mass position and speed in terms of $q_1$, $q_2$ and their time derivatives $\omega_1$ and $\omega_2$ are shown in figures 6 and 7, respectivelly:

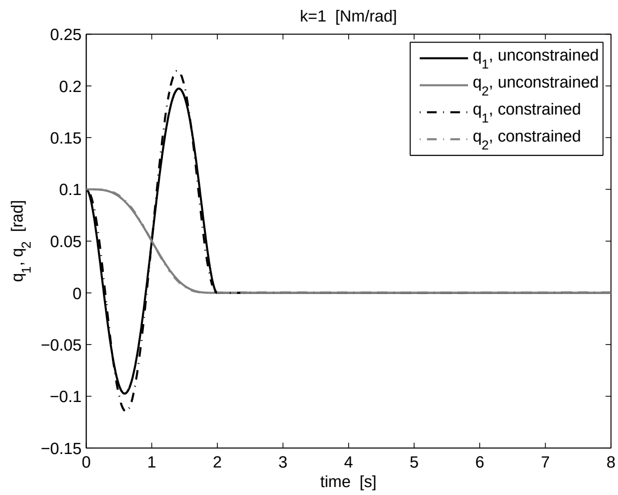


Figure 6: Mass positions $q_1$ and $q_2$, nominal trajectory, obtained with feedforward control action applied to the nominal system



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

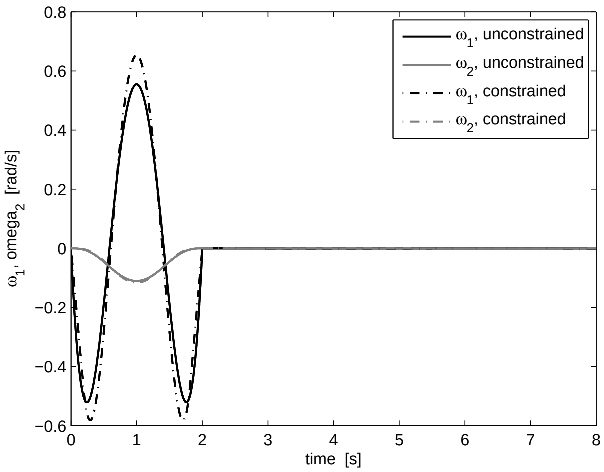


Figure 7: Mass speeds $\omega_1$ and $\omega_2$, nominal trajectory, obtained with feedforward control action applied to the nominal plant

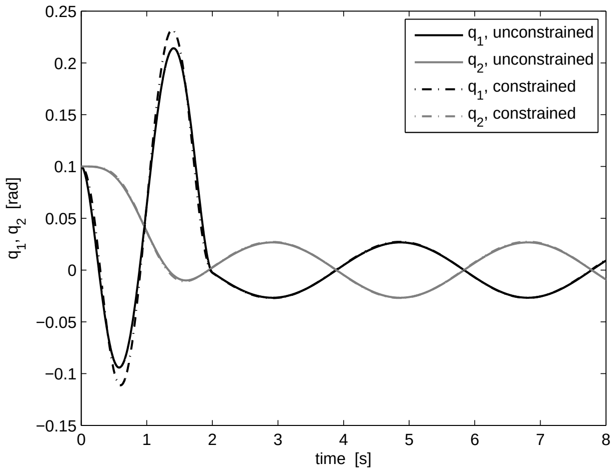


Figure 8: Mass position $q_1$ and $q_2$ for the unconstrained and constrained cases with elastic constant $k = 1.3\ N/m$, nominal trajectory

I can be seen in Fig. 6 and 7 that the planned trajectories are very accurate, since they allow to reach the right boundary condition with precision and null residual speed or vibration. This happens only for the nominal system, i.e. for a system in which the elastic constant $k$ is equal to 1 Nm/rad. If the value of $k$ is different from the nominal value, and in particular if it is equal to 1.3 Nm/rad, the result of applying the feedforward torque profile is shown in fig. 6 : after the completion of the task, i.e. after 2 seconds, the right boundary condition is not met, and persistent oscillations are induced on the system. The analysis of the plot shows that a 30% mismatch between the estimated and the actual stiffness of the joint can lead to a high level of residual vibrations. The peak-to-peak value of residual vibration is in this case equal to 0.056 rad, i.e. more than the double of the prescribed mass displacement at the end of the trajectory.

In order to quantify the actual level of robustness of the outcome of the trajectory planning algorithm, a simple measurement is introduced. The actual distance between the conditions reached by the system and the right boundary condition is evaluated in terms of residual energy



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

at final time $t_f$. The residual energy of the system $E(t_f)$ can be evaluated as the sum of kinetic energy $T$ and elastic energy $U$:

$$E = T + U \tag{23}$$

in which:

$$T = \frac{1}{2}\left(J_1{\omega_1}^2 + J_2{\omega_2}^2\right); \qquad U = \int_0^{\Delta q^*} F\Delta q d\Delta q = \int_0^{\Delta q^*} (k\Delta q + k^3\Delta q) d\Delta q = k\left(\frac{1}{2}\Delta q^{*2} + \frac{1}{4}\Delta q^{*4}\right)$$

Figure 9 shows how the value of of the elastic constant $k$ affects the value of residual energy after task completion. It can be seen in figure 9 that the residual energy is equal to zero only for $k = 1\ N/m$, and that the residual energy quickly grows with increasing and decreasing values of $k$. This applies with very similar trends for both the constrained and the unconstrained solutions. The following part of the paper will show how the use of sensitivity functions allows to improve the robustness of the computed trajectories with respect to variations of the elastic constant $k$.

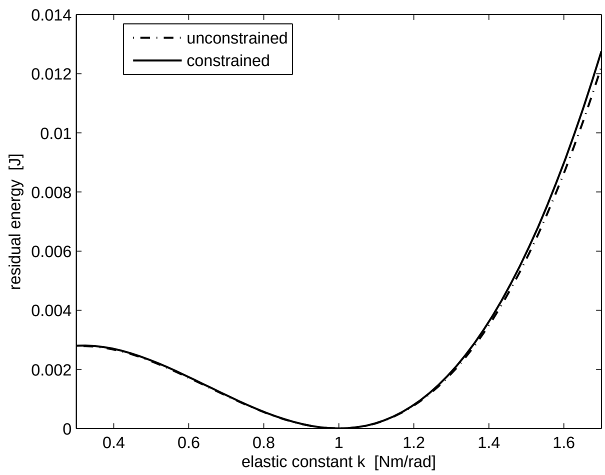


Figure 9: Nominal trajectory: residual energy for $k \in [0.3, 1.7]$, constrained and unconstrained solutions

# 5 Numerical results: computation of robust constrained trajectory

As explained in section 3, the robustness of the algorithm can be improved by imposing an additional set of constraints on the sensitivity functions, as in Eq. (14). Therefore the system in Eq. (17) can be augmented by including the sensitivity function of the state vector $\mathbf{x}$ with respect to the elastic constant $k$:



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

$$
\begin{aligned}
&\frac{d}{dt}\left(\frac{\partial \omega_1}{\partial k}\right) = (q_2-q_1)+(q_2-q_1)^3+k\left(\frac{\partial q_2}{\partial k}-\frac{\partial q_1}{\partial k}\right)^3+3k\left(\frac{\partial q_2}{\partial k}-\frac{\partial q_1}{\partial k}\right)(q_2-q_1)^2\\
&\frac{d}{dt}\left(\frac{\omega_2}{\partial k}\right) = -(q_2-q_1)-(q_2-q_1)^3-k\left(\frac{\partial q_2}{\partial k}-\frac{\partial q_1}{\partial k}\right)^3-3k\left(\frac{\partial q_2}{\partial k}-\frac{\partial q_1}{\partial k}\right)(q_2-q_1)^2\\
&\frac{d}{dt}\left(\frac{\partial q_1}{\partial k}\right) = \frac{\partial \omega_1}{\partial k}\\
&\frac{d}{dt}\left(\frac{q_2}{\partial k}\right) = \frac{\partial \omega_2}{\partial k}
\end{aligned}
\tag{24}
$$

This choice of parametric uncertainty can be practically useful in all the cases in which the elastic constant of the flexible joint cannot be estimated with sufficient accuracy, or in the cases in which variations of the elastic constant are not described by the dynamic model used for trajectory planning. The TPBVP must be therefore formulated considering as the plant dynamics the ODE system in Eq. (17) augmented with the four differential equations in Eq. (24).

The hamiltonian of the system, computed as in Eq. (4), is in this case:

$$
\begin{aligned}
\mathcal{H} = {} & \lambda_3\omega_1-\lambda_1\left(k(q_1-q_2)-u+k(q_1-q_2)^3\right)+\lambda_4\omega_2+\lambda_7\frac{\partial \omega_1}{\partial k}+\lambda_8\frac{\partial \omega_2}{\partial k}\\
&-\lambda_5\left(q_1-q_2+(q_1-q_2)^3+k\left(\frac{\partial q_1}{\partial k}-\frac{\partial q_2}{\partial k}\right)+3k(q_1-q_2)^2\left(\frac{\partial q_1}{\partial k}-\frac{\partial q_2}{\partial k}\right)\right)\\
&+\lambda_6\left(q_1-q_2+(q_1-q_2)^3+k\left(\frac{\partial q_1}{\partial k}-\frac{\partial q_2}{\partial k}\right)+3k(q_1-q_2)^2\left(\frac{\partial q_1}{\partial k}-\frac{\partial q_2}{\partial k}\right)\right)\\
&+\lambda_2\left(k(q_1-q_2)+k(q_1-q_2)^3\right)+\frac{u^2}{2}
\end{aligned}
\tag{25}
$$

and therefore the optimal solution for the control action $u$ is, again:

$$
u^* = -\lambda_1 \tag{26}
$$

The results above refer to the unconstrained solution, therefore the resulting trajectory will allow high values of torque when the execution time $t_f$ is low. The expressions of the minimizing Hamiltonian $\mathcal{H}^*$ and of the system of differential equation $\dot{\mathbf{y}}^* = f(\mathbf{x}, \Lambda)$ are here omitted. In order to include constraints on $u$, the same procedure already used for the non-robust solution must be applied here. The resulting system of differential equation is here omitted, since they are the result of the same substitution operated to Eq. (22) in the previous section for the non-robust solution. The boundary conditions for the TPBVP to be solved are:

$$
\begin{gathered}
q_1(t=0)=q_2(t=0)=0.1; \quad \omega_1(t=0)=\omega_2(t=0)=0;\\
\frac{\partial q_1}{\partial k}(t=0)=\frac{\partial q_2}{\partial k}(t=0)=0; \quad \frac{\partial \omega_1}{\partial k}(t=0)=\frac{\partial \omega_1}{\partial k}(t=0)=0;\\
q_1(t=t_f)=q_2(t=t_f)=0; \quad \omega_1(t=t_f)=\omega_2(t=t_f)=0;\\
\frac{\partial q_1}{\partial k}(t=t_f)=\frac{\partial q_2}{\partial k}(t=t_f)=0; \quad \frac{\partial \omega_1}{\partial k}(t=t_f)=\frac{\partial \omega_2}{\partial k}(t=t_f)=0;
\end{gathered}
$$

while the final time is $t_f = 3.75$ s. Final time has been increased to 3.75 s in order to produce a trajectory with the same peak torque as the nominal one. The procedure followed to formulate and solve the robust optimization problem, i.e. to plan a robust trajectory for the dynamic system of choice, is summarized as Alogritm 1.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

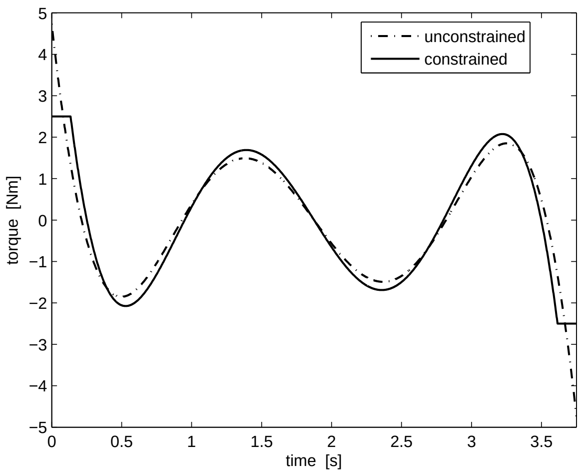


Figure 10: Robust trajectory for the nonlinear joint system: actuator torque, constrained and unconstrained solutions

---

**Algorithm 1** Robust trajectory computation

---

1. define ODE system in vectorized form: $\dot{\mathbf{x}} = \Omega(\mathbf{x}, t, u, \mu)$ with $\mathbf{x} \in \Re^n$
2. compute the partial derivatives with respect to the uncertain parameter $\mu$: $\dot{\mathbf{S}} = \dfrac{\partial\Omega(\mathbf{x}, t, u, \mu)}{\partial\mu}$
3. compute the augmented ODE system: $\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{S}} \end{bmatrix} = \begin{bmatrix} \Omega(\mathbf{x}, t, u, \mu) \\ \dfrac{\partial\Omega(\mathbf{x}, t, u, \mu)}{\partial\mu} \end{bmatrix}$
4. define the vector of Lagrangians: $\Lambda = [\lambda_1, \ldots, \lambda_{2n}]$
5. define the cost function: $f = f(\mathbf{x}, \mathbf{S}, t, u, \mu)$
5. write the Hamiltonian: $\mathcal{H} = f + \Lambda' \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{S}} \end{bmatrix}$
6. compute the optimal control action $u^*$ as the solution of: $\dfrac{\partial\mathcal{H}}{\partial u} = 0$
7. compute the minimizing Hamiltonian: $\mathcal{H}^* = \mathcal{H}|_{u \leftarrow u^*}$
8. compute the ODE: $\dot{\mathbf{y}} = \begin{bmatrix} \dfrac{\mathcal{H}^*}{\partial\lambda_i} \\ -\dfrac{\mathcal{H}^*}{\partial x_i} \end{bmatrix}$ with $i = 1, \ldots, n$
9. define $2n$ boundary conditions
10. solve the TPBVP with inputs defined at step 8 and 9

---



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

Figure 11 shows the solution of the TPBVP for the robust system, with and without constraints on $u$. In both cases, the two masses are quickly brought close to the rest position, but following a trajectory that is very different from the one shown in figure 5. It can also be seen in Fig. 11 that the maximum value of $u$ is effectively limited in the range $[-2.5, 2.5]$ Nm for the constrained solution, i.e. the same torque limit has been used for both the nominal and the robust solution. Again, in order to evaluate the actual robustness of the two trajectories, the plant is fed with the feedforward torques computed as the solution of the optimization problem and shown in figure 10. Figure 12 shows the evolution of the mass positions $q_1$ and $q_2$ when a nominal and a perturbed plant is fed with a feedforward force signal computed using the two strategies explained above. The speed of the two masses in the same operative conditions are shown in figure 13. For the nominal plant, both trajectories are accurate: it can be seen in Figure 12 and 13 that for both the unconstrained and constrained solution the masses are brought to the desired position after 5 seconds with null residual vibration. On the other hand, if the same test is performed on a perturbed plant, i.e. on a plant with the elastic coefficient $k$ increased by 30% over the nominal value, a noticeable residual vibration appears after task completion. The peak-to-peak value of the residual vibration is equal to 0.0161 rad, which is 3.4 times lower than the same value obtained under the same operative conditions but using the non-robust trajectory. Also in this case the inclusion of constraints does actually lead to an higher residual vibration.

The residual energy in the unconstrained solution for the nominal and the robust solutions are shown in of Figure 15: it can be seen here that the residual energy in the robust case is lower than the one for the nominal case for all values in the range $[0.3, 1.7]$ Nm/rad, therefore for all the cases taken into consideration the robust trajectory does actually provide an improvement in terms of residual energy.

We can conclude that the robust approach to trajectory planning presented in this paper, while leading to larger operative speed and longer execution time under similar actuator efforts, does improve the response of the system in terms of residual vibration also for large variation of the uncertain parameter around its nominal value. It has been shown also that, for the cases taken into consideration, the inclusion of hard constraints on actuator effort does not affect the robustness properties of the robust and the non-robust trajectory.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

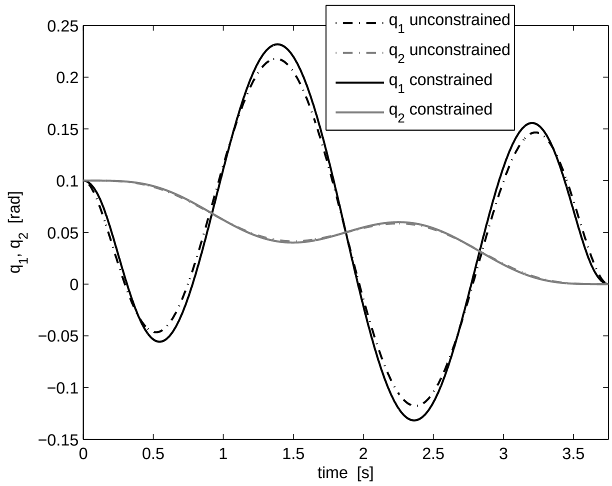


Figure 11: Robust trajectory for the nonlinear spring-mass system: $\beta_1$ and $\beta_2$, constrained and unconstrained solutions

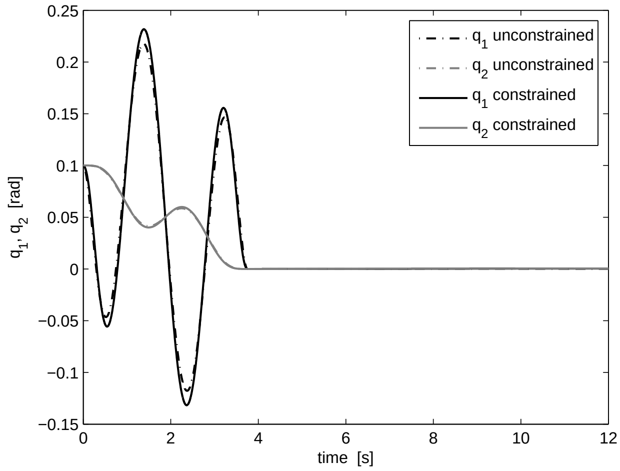


Figure 12: Robust trajectory for the nonlinear spring-mass system: mass positions $q_1$ and $q_2$, constrained and unconstrained solutions for $k = 1$ (left) and $k = 0.8$ $N/m$ (right)



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

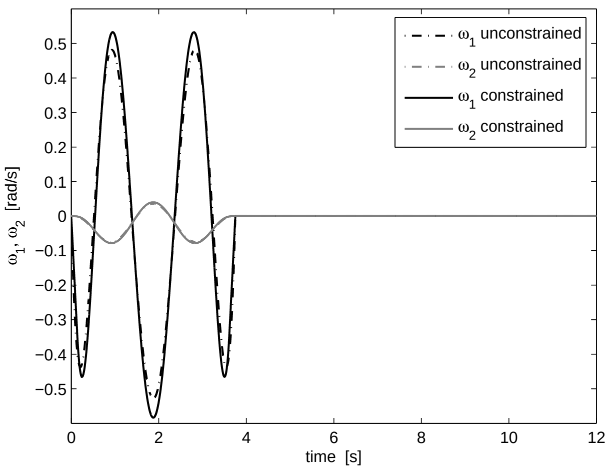


Figure 13: Robust trajectory for the nonlinear spring-mass system: mass speeds $\omega_1$ and $\omega_2$, constrained and unconstrained solutions

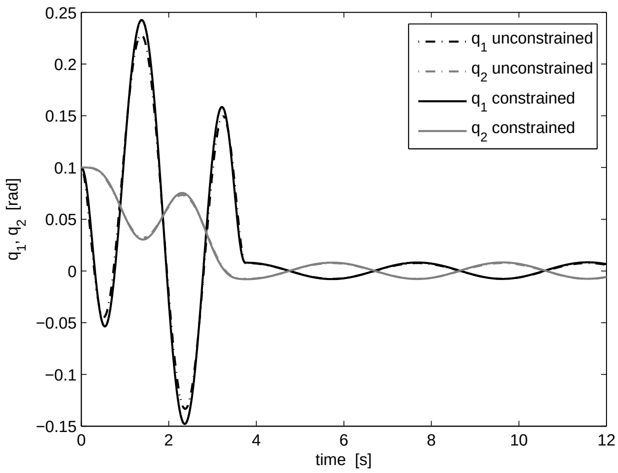


Figure 14: Robust trajectory for the nonlinear spring-mass system: mass positions $q_1$ and $q_2$, constrained and unconstrained solutions for $k = 1.3$ Nm/rad



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

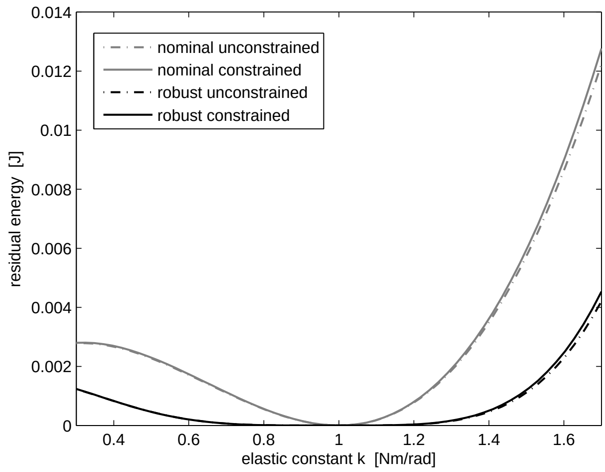


Figure 15: Residual energy: comparison between unconstrained nominal, unconstrained robust, constrained nominal and constrained robust trajectories

# 6 Conclusions

In this paper the problem of generating robust constrained model-based trajectories for nonlinear mechatronic systems is dealt with. The paper proposes a method to improve the robustness of the planned trajectory to parametric mismatches, which is based on the solution of a Two-Point Boundary Value Problem augmented with sensitivity functions. Constraints can be included by the using of an accurate smoothing technique. It must be highlighted that the proposed method applies also to nonlinear plants, unlike other works available in literature. The effectiveness of the approach is showed by a benchmark problem, i.e. an undamped elastic joint system with a nonlinear stiffness characteristic. The results indicates that the use of sensitivity function does lead to a sensible improvement of the response of the systems with parametric mismatches. The improvement in terms of residual energy after time completion can be achieved by the sensitivity function approach used in this works for a wide range of values of the uncertain parameter. Moreover the smoothing techniques used to introduce constraints allows to limit with precision the actuator effort. The inclusion of such constraint does not influence the robustness to the chosen parametric uncertainty.

# References


[1] B. Siciliano and O. Khatib, *Springer handbook of robotics.* Springer, 2008.

[2] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8, VDE, 2010.

[3] S. Moberg, *On modeling and control of flexible manipulators.* PhD thesis, Department of Electrical Engineering, Linköping University, Sweden, 2007.

[4] A. De Luca and G. Di Giovanni, "Rest-to-rest motion of a two-link robot with a flexible forearm," in *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on*, vol. 2, pp. 929–935, IEEE, 2001.




P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

[5] M. Benosman and G. Le Vey, "Control of flexible manipulators: A survey," *Robotica*, vol. 22, no. 05, pp. 533–545, 2004.

[6] G. Boschetti, D. Richiedei, and A. Trevisani, "Delayed reference control for multi-degree-of-freedom elastic systems: Theory and experimentation," *Control Engineering Practice*, 2011.

[7] A. A. Ata, "Optimal trajectory planning of manipulators: a review," *Journal of Engineering Science and Technology*, vol. 2, no. 1, pp. 32–54, 2007.

[8] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Trajectory planning in robotics," *Mathematics in Computer Science*, pp. 1–11, 2012.

[9] V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, "Experimental validation of minimum time-jerk algorithms for industrial robots," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 2, pp. 197–219, 2011.

[10] R. Caracciolo, D. Richiedei, and A. Trevisani, "Experimental validation of a model-based robust controller for multi-body mechanisms with flexible links," *Multibody System Dynamics*, vol. 20, no. 2, pp. 129–145, 2008.

[11] S. Islam and X. P. Liu, "Robust sliding mode control for robot manipulators," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 6, pp. 2444–2453, 2011.

[12] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer, 2008.

[13] M. Boryga and A. Graboś, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Mechanism and machine theory*, vol. 44, no. 7, pp. 1400–1419, 2009.

[14] C. Louembet, F. Cazaurang, and A. Zolghadri, "Motion planning for flat systems using positive b-splines: An lmi approach," *Automatica*, vol. 46, no. 8, pp. 1305–1309, 2010.

[15] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, 2005.

[16] D. Balkcom and M. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 199–217, 2002.

[17] O. Dahl, "Path constrained motion optimization for rigid and flexible joint robots," in *International Conference on Robotics and Automation*, vol. 2, pp. 223–229, IEEE, 1993.

[18] M. Korayem, H. R. Nohooji, and A. Nikoobin, "Optimal motion generating of nonholonomic manipulators with elastic revolute joints in generalized point-to-point task," *International journal of advanced design and manufacturing technology*, vol. 3, no. 2, pp. 1–9, 2010.

[19] A. Abe, "Trajectory planning for residual vibration suppression of a two-link rigid-flexible manipulator considering large deformation," *Mechanism and Machine Theory*, vol. 44, no. 9, pp. 1627–1639, 2009.

[20] M. H. Korayem, A. Nikoobin, and V. Azimirad, "Trajectory optimization of flexible link manipulators in point-to-point motion," *Robotica*, vol. 27, no. 6, p. 825, 2009.

[21] M. Korayem, M. Haghpanahi, H. Rahimi, and A. Nikoobin, "Finite element method and optimal control theory for path planning of elastic manipulators," in *New Advances in Intelligent Decision Technologies*, pp. 117–126, Springer, 2009.

[22] P. Boscariol and A. Gasparetto, "Model-based trajectory planning for flexible-link mechanisms with bounded jerk," *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 90–99, 2013.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

[23] J. Ackermann, A. Bartlett, D. Kaesbauer, W. Sienel, and R. Steinhauser, *Robust control.* Springer, 1993.

[24] M. Karkoub, G. Balas, K. Tamma, and M. Donath, "Robust control of flexible manipulators via $\mu$synthesis," *Control engineering practice*, vol. 8, no. 7, pp. 725–734, 2000.

[25] P. Tomei, "Robust adaptive friction compensation for tracking control of robot manipulators," *Automatic Control, IEEE Transactions on*, vol. 45, no. 11, pp. 2164–2169, 2000.

[26] D. Gallardo, O. Colomina, F. Flórez, and R. Rizo, "A genetic algorithm for robust motion planning," *Tasks and Methods in Applied Artificial Intelligence*, pp. 115–121, 1998.

[27] K. G. Shin and N. D. McKay, "Robust trajectory planning for robotic manipulators under payload uncertainties," *Automatic Control, IEEE Transactions on*, vol. 32, no. 12, pp. 1044–1054, 1987.

[28] B. Houska, *Robust optimization of dynamic systems.* PhD thesis, PhD thesis, Katholieke Universiteit Leuven, 2011.(ISBN: 978-94-6018-394-2), 2011.

[29] T. Singh, *Optimal reference shaping for dynamical systems: theory and applications.* CRC PressI Llc, 2010.

[30] T. A. Hindle and T. Singh, "Desensitized minimum power/jerk control profiles for rest-to-rest maneuvers," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 5, pp. 3064–3068, IEEE, 2000.

[31] R. Kased and T. Singh, "Rest-to-rest motion of an experimental flexible structure subject to friction: Linear programming approach," in *AIAA Guidance, Navigation and Control Conference, San Francisco, CA*, 2005.

[32] G. Boschetti, R. Caracciolo, D. Richiedei, and A. Trevisani, "Moving the suspended load of an overhead crane along a pre-specified path: A non-time based approach," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 256–264, 2014.

[33] L. Pontryagin and R. Gamkrelidze, *The mathematical theory of optimal processes*, vol. 4. CRC, 1986.

[34] L. Shampine, I. Gladwell, and S. Thompson, *Solving ODEs with MATLAB.* Cambridge University Press, 2003.

[35] R. W. Holsapple, "A modified simple shooting method for solving two-point boundary value problems," 2012.

[36] W. Stein *et al.*, "Sage: Open source mathematical software," 2008.

[37] R. D. Jenks, R. S. Sutor, and S. C. Morrison, *AXIOM: the scientific computation system*, vol. 6. Springer-Verlag New York, 1992.

[38] S. Moberg and J. Öhr, "Robust control of a flexible manipulator arm: A benchmark problem," in *Proceedings of the 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[39] P. Axelsson, "Evaluation of six different sensor fusion methods for an industrial robot using experimental data," in *10th International IFAC Symposium on Robot Control, Dubrovnik, Croatia*, 2012.

[40] P. Axelsson, M. Norrlöf, E. Wernholt, and F. Gustafsson, "Extended kalman filter applied to industrial manipulators," in *Proceedings of Reglermöte 2010, Lund, Sweden*, 2010.

[41] S. Moberg, J. Öhr, and S. Gunnarsson, "A benchmark problem for robust control of a multivariable nonlinear flexible manipulator," in *Proceedings of the 17th IFAC World Congress*, 2008.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239

[42] S. Moberg, E. Wernholt, S. Hanssen, and T. Brogårdh, "Modeling and parameter estimation of robot manipulators using extended flexible joint models," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 3, p. 031005, 2014.

[43] M. W. Spong, "Modeling and control of elastic joint robots," *Journal of dynamic systems, measurement, and control*, vol. 109, no. 4, pp. 310–318, 1987.

[44] L. F. Shampine, J. Kierzenka, and M. W. Reichelt, "Solving boundary value problems for ordinary differential equations in matlab with bvp4c," *Tutorial notes*, 2000.

[45] S. Avvakumov and Y. N. Kiselev, "Boundary value problem for ordinary differential equations with applications to optimal control," *Spectral and Evolution Problems 2000*, vol. 10, 2000.



P. Boscariol, A. Gasparetto (2016)
Optimal trajectory planning for nonlinear systems: robust and constrained solution. Robotica, 34, pp 1243-1259
doi:10.1017/S0263574714002239