Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Backstepping Controller Retuning using ε Subdifferential Optimization

Bilal Lassami and Stéphane Font

Abstract— **This paper presents a general optimization approach in automatic's field. It discusses the design and retuning of controllers as well as the validation of the general manufacturer specifications, which will be expressed by a frequency and/or temporal templates in order to be solved by a general optimization problem. This is a very complex and particular problem. It will be shown that it involves nondifferentiable functions and criteria. The resolution of this kind of problems can be difficult or impossible via gradient and all classical descent algorithms. In this study, we propose an ε subdifferential algorithm, very efficient for nonsmooth optimization. This algorithm, mixing with an exact computation of gradient based on parametric sensitivity functions, appears to be well suited to problems with nonsmooth costs and constraints. As illustration, this method will be used to retune backstepping controller for a magnetic suspension system. Simulations and comparison results are given to demonstrate the effectiveness of the proposed approach.**

I. Introduction

Since the beginning of control theory, the design of optimal controllers taking into consideration complex specifications has been a fundamental challenge. A general trend has been to search for high performances (rapidity, precision, rejection or attenuation of perturbation signals…), while ensuring moderate control signals and good robustness properties. This need of high system performances, conjointly with the evolution of computation techniques and information processing, reinforces the necessity of optimization in automatic. Together with the growing of calculation power, the requirements evolved. The ability to take into account complex specifications feeds several subjects of research. Many studies related to the use of optimization for complex engineering requirements with low complexity algorithms are developed. These theories mainly focus on convex optimization methods that concern with synthesis techniques of controllers. Particularly, during the seventies, quadratic methods as LQ [1], [2] appeared and grew. At the end of the eighties and the beginning of the nineties, the methods known as robust control have been greatly developed. Concerning solvable cases, generally convex, we can quote the multi-criterion syntheses H_2 / H_∞ [3], the uncoupled H_∞ / H_∞ [4] and feasibility problem using Youla-Kucera parameterization [5]. In the general case (not algebraically solvable or nonconvex), there is no guarantee to obtain the absolute optimum. Nevertheless, optimization allows to retune controller parameters. Indeed, the control law design is seldom a direct operation. Generally multiple stages are necessary to obtain a satisfactory result. When the design conditions are modified, the most effective way is not necessarily to start again the work from the beginning. Thus, the need for a retuning may arise all along the development of the system each time the design model or specifications evolve.

In this article, the last situation is considered. Generic criteria properties will be shown as well as the way to consider the problem in order to get an efficient resolution. We will focus on methods, which can be developed out of the convex context. For complex specifications, the proposed approach allows an exact formulation of requirements to the detriment of an exact analysis of their feasibilities. The versatility of these generic optimization approaches is illustrated through a magnetic suspension system. The most important aim is to show the potential of our technique through an application for which retuning controller parameters by optimization is yet very difficult using a classical descent algorithm.

This paper is structured as follows. First, a formulation of generic specifications in control design problems is presented. The result is that all specifications are expressed by constraints in a global optimization problem. The resolution of such problem is difficult and requires specific algorithms. Subsequently, a descent algorithm based on the ε subdifferential notion is exposed, which is able to solve nonsmooth formulated problems. The last section is dedicated to present the application and its simulation results.

II. The Global Optimization Problem Formulation

Mathematical formulation of the different specifications will allow to express the retuning of controllers as a global optimization problem mixing temporal and/or frequency demands.

Let's consider the diagram given by figure 1. The tuned controller parameters are designed by the vector θ_1.

Authors are with Automatic Control Department, Ecole Supérieure d'Electricité (Supélec) Gif-sur-Yvette, F-91192F cedex France. (phone: +33169851370-86; fax: +33169851389; e-mail: Bilal.Lassami-Stephane.Font @supelec.fr).

0-7803-9568-9/05/$20.00 ©2005 IEEE

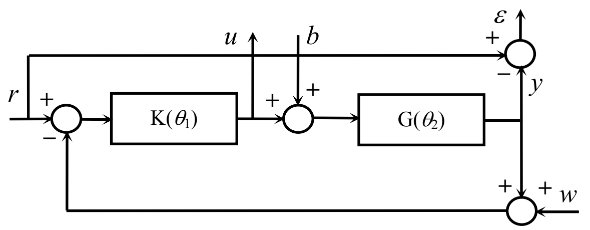

Fig. 1. Feedback system with tuned variables.

The model of the system to be controlled can also depend on adjustable parameters θ_2. The global parameter $\theta = \left(\theta_1^T, \theta_2^T\right)^T$ constitutes the vector of decision variables.

Any transfer functions of the open or closed loop system will show the following dependence:

$$H\left(K(\theta_1), G(\theta_2), j\omega\right) = H(\theta, j\omega) \qquad (1)$$

For a particular input signal (r or b), any output signals will be of the form:

$$s\left(G(\theta_2), K(\theta_1), r(t), t\right) = s(\theta, t) \qquad (2)$$

The classical criteria of the specifications sheet (time response, maximum overshoot, cut-off frequency …) are consequently also dependent on parameters θ. These will be noted, respectively, in temporal and frequency domains by the expressions

$$\alpha_t\left(s(\theta,t), t\right) = \alpha_t(\theta) \qquad (3)$$

$$\alpha_\omega\left(H(\theta, j\omega), \omega\right) = \alpha_\omega(\theta) \qquad (4)$$

Generally, the control problem will be translated using constraints on indicators α_t and α_ω or on trajectories [6]. These last requirements are formulated, respectively, in temporal and in frequency domains, using respectively one of the following inequalities:

$$\begin{cases} \forall t > 0, s_{\min}(t) \le s(\theta,t) \le s_{\max}(t) \\ F_t\left(\alpha_t(\theta)\right) \le 0 \end{cases} \qquad (5)$$

$$\begin{cases} \forall \omega > 0, H_{\min}(\omega) \le \left|H(\theta, j\omega)\right| \le H_{\max}(\omega) \\ F_\omega\left(\alpha_\omega(\theta)\right) \le 0 \end{cases} \qquad (6)$$

For example, a request of a phase margin greater than 60° will be formulated as follows $F_\omega(\Delta\phi) = 60^\circ - \Delta\phi$.

It can be noticed that these various constraints can be equivalently formulated as criteria functions. For example, requirements that can be formulated using (5) or (6) can be equivalently formulated using one of the following criteria.

$$J_t = \max\left(\max_{t>0}\left(s(\theta,t) - s_{\max}(t)\right), \max_{t>0}\left(s_{\min}(t) - s(\theta,t)\right) \right) \qquad (7)$$

$$J_\omega = \max_{\omega>0}\left(\left|H(\theta, j\omega)\right| - H_{\max}(\omega)\right) \qquad (8)$$

A constraint will be verified if and only if the associated optimal criterion is negative.

Generally, the global optimization problem can be stated by mixing one or more criterion as equation (7) or (8) and one or more constraints like expressions (5) or (6).

For generic specifications, calculations of criteria and constraints are not easy. For examples; the calculations of gain and phase margin require, among other things, the resolution of polynomial equations; the time response calculations require the use of an unidirectional optimization. Furthermore, all temporal criteria require the integration of differential equations. In generic case, only estimation of trajectories can be obtained, that contains a finite number of estimated points. So the numerical estimation of sensitivity (or gradient) of such object is really very sensitive using indirect approach (usually finite difference approach). Moreover, constraints on trajectory lead to a semi-infinite problem: the underlying maximum function, as (7) or (8) must be calculated by optimization, gridding or other methods. To conclude, the main fact is that the criterion J can not be exactly computed and only an estimated value $\hat{J}$ is available. It will be noted: $\widehat{J} = J + b$ where b is the numerical noise of criterion calculation.

III. Resolution of the Global Optimization Problem

The proposed approach is based on the descent algorithm. It is a very efficient method when a descent direction can be well determined.

A. The Descent Algorithm

The structure of this class of algorithms allows to exploit fully the local descent information. It is composed of the following steps:

Step 1: Choose an initial parameters values $\theta = \theta_{init}$.
Step 2: Determine a descent direction vector $d(\theta)$.
Step 3: Find the step length λ according to the direction $d(\theta)$ such that $\lambda > 0$ and $J(\theta + \lambda\, d(\theta)) < J(\theta)$
Step 4: Updating the current point: $\theta \leftarrow \theta + \lambda\, d(\theta)$
Step 5: If stop criterion not verified go to step 2 else STOP.

The most famous method of this class of algorithms is the 'gradient method'. It is based on computation of the local gradient which presents the best local choice when it is defined [7]. The descent direction is then determined at each iteration k as follows.

$$\theta_{k+1} \leftarrow \theta_k + \lambda\, d_k \text{ with: } d_k = -\hat{g}_k \text{ and } g_k = \nabla J(\theta_k) = \left.\frac{\partial J}{\partial \theta}\right|_{\theta=\theta_k}$$

This particular algorithm presents many advantages, on the contrary to other information of structure or more elevated order (Hessian), it requires less expensive calculations than the superior order methods and it is robust with respect to estimation errors on the gradient; it is sufficient to verify $\hat{g}_k \cdot g_k > 0$ to get the convergence of the algorithm. However, the possibility of convergence, in a reasonable time, depends on the gradient calculations; it must be reliable (with a good precision) and easily tractable (low complexity of calculations).

B. Gradient Calculation Using Parametric Sensitivities

Two kinds of methods can be distinguished for the computation of the gradient. The first one is the indirect method using finite differences techniques. This method is very useful but leads to unreliable result when the noise is relatively strong. For instance, the analysis of variance gives:

$$\sigma\left[\frac{\Delta\hat{J}}{\Delta\theta}\right]=\sigma\left[\frac{1}{h}\left[J(\theta+h\cdot u)+b_1-J(\theta)-b_2\right]\right]\underset{h\to 0}{\approx}\frac{2\sigma(b)}{h} \tag{9}$$

where u denotes a unit vector and b_i the noise of gradient estimation assumed that $b_i \to N(0,\sigma(b))$.

The second method is the direct calculus. It is a precise method but it requires an important first step of calculus before running optimisation when it is applied to semi-infinite problems [8]. For generic problems, the estimation of gradient is strongly noised, so in order to ameliorate the quality of the local information, the direct calculus will be considered here. The gradient can be computed using the sensitivity function approach. The principle of this method will now be exhibited in temporal domain through a NLI model. Let's consider the system:

$$\begin{cases}\dfrac{dx(t)}{dt}=f[x(t,a)]\\ y(t,a)=h[x(t),a]\end{cases} \tag{10}$$

f,h can also depend on the command $u(t)$ and the time t. a denotes a vector of any parameter of the equations, relation with θ will be explained in the sequel.

The parametric sensitivities of the output signal y with respect to arguments a are defined by the following expression for $i \in \{\, 1,...,\dim(a) \,\}$ [9], [10]:

$$S_{y/a_i}(t,a)=\frac{\partial}{\partial a_i}y(t,a) \tag{11}$$

This definition applied to the NLI model gives

$$S_{y/a_i}(t,a)=\frac{\partial h[x(t),a]}{\partial x^T}S_{x/a_i}(t,a)+\frac{\partial h[x(t),a]}{\partial a_i} \tag{12}$$

The derivation with respect to parameters a gives:

$$\frac{d\left[S_{x/a_i}(t,a)\right]}{dt}=\frac{\partial f[x(t),a]}{\partial x^T}S_{x/a_i}(t,a)+\frac{\partial f[x(t),a]}{\partial a_i} \tag{13}$$

The computation of the parametric sensitivities of the output signal y with respect to coefficients a requires $\dim(a)+1$ simulations (the resolution of the system of equations defined by 10 and 13).

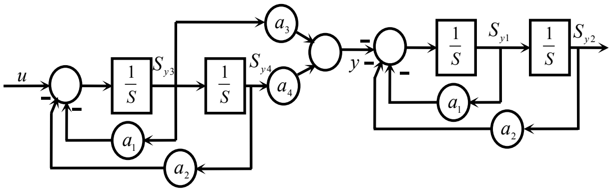

Fig. 2. An example of the sensitivity block diagram (*n,m*)=(2,1); it permits an efficient calculation of the differential equations of sensitivity functions.

This indicates that the direct method is not more complicated anymore that the one by finite differences.

If coefficients a depend on parameters θ, the parametric sensitivities of the output signal y with respect to parameters θ are then given by:

$$S_{y/\theta_k}=\sum_{i=1}^{n+m+1}\frac{\partial a_i}{\partial\theta_k}S_{y/a_i} \tag{14}$$

The sensitivity approach can be easily developed in the linear case where it presents a lot of simplifications. In fact, all transfer functions of the control loop can be expressed by a differential equation:

$$\frac{d^n}{dt^n}y+\sum_{i=0}^{n-1}a_{n-i}\frac{d^i}{dt^i}y=\sum_{j=0}^{m}a_{n+m+1-j}\frac{d^j}{dt^j}u \tag{15}$$

The partial derivatives of the differential equation (15) with respect to coefficients a give $n+m+1$ differential equations. For $k=1,2,...,n$, it becomes:

$$\frac{d^n}{dt^n}S_{y/a_k}+\sum_{i=0}^{n-1}a_{n-i}\frac{d^i}{dt^i}S_{y/a_k}=-\frac{d^{n-k}}{dt^{n-k}}y \tag{16}$$

For $k=n+1,...,n+m+1$,

$$\frac{d^n}{dt^n}S_{y/a_k}+\sum_{i=0}^{n-1}a_{n-i}\frac{d^i}{dt^i}S_{y/a_k}=\frac{d^{n+m+1-k}}{dt^{n+m+1-k}}u \tag{17}$$

Let's note that the resolution of the system of $n+m+2$ equations, defined by equations (15), (16) and (17), only requires the use of an $2\times n$ order model. The evaluation requires a numerical integration for which the error is small enough and above all is not amplified as in the case of the finite differences approach. The figure 2 shows the reduced block diagram of these equations for $(n,m)=(2,1)$.

As an illustrative example, the following criterion can be considered.

$$J(\theta)=\sum_{i=1}^{t_f}w_i\left|y(t_i,\theta)-Y_{\max}(t_i)\right|_+ \tag{18}$$

with $|f|_+=\max(f,0)$.

The gradient of this criterion is given almost everywhere for $k\in\{1,...,n+m+1\}$ by the following expression:

$$\frac{\partial J}{\partial\theta_k}=\sum_{i=1}^{t_f}\frac{w_i}{2}\cdot S_{y/\theta_k}\cdot\left(1+\frac{\left|y(t_i,\theta)-Y_{\max}(t_i)\right|}{y(t_i,\theta)-Y_{\max}(t_i)}\right) \tag{19}$$

C. *Nonsmooth Criteria*

An attentive observation of the several criteria formulated on the retuning controller problems shows that the classical specifications are often formulated by nonsmooth criteria and constraints. For example, criteria which contain any 'max' function are nonsmooth when the maximum is reached simultaneously in several points; see (18) and (19). However, they present the particularity to be nearly differentiable; the space of nonsmooth points has measure zero [11], [12]. Nevertheless, the subspace of nondifferentiable points is usually reached during the optimization process because these points are often local minima in almost all directions. At these locations, the determination of a descent direction is not possible with the gradient. A generalized approach using subgradient and subdifferential have to be used [11].

1) Subgradient: Let's consider X a nonempty convex set and f a convex function not necessarily differentiable, $f : X \in \Re^n \rightarrow Y \in \Re$ (in a local meaning if necessary). The subgradient ξ at $\bar{x}$ is defined by the property: $f(x) \geq f(\bar{x}) + \xi^T(x - \bar{x})$ globally (or locally) verified on the convex set.

2) Subdifferential: The set of subgradients of a function f at $\bar{x}$ is the subdifferential of $f(x)$ at $\bar{x}$. It is denoted by $\partial f(\bar{x})$. If the function f is differentiable at $\bar{x}$ then the subdifferential $\partial f(\bar{x})$ is reduced to the gradient $\nabla f(\bar{x})$.
The subdifferential is not such a good choice for a descent method because it is typically very hard to determine and one subgradient does not necessarily represent an efficient ascent direction.

In order to construct a rigorous procedure, the Clarke subgradient is used because it presents an interesting approximation quality [13].
The proposed algorithm may be applied to any function $f : \Re^n \rightarrow \Re$ that is continuous and differentiable almost everywhere on $\Re^n$, in practice, it seems to put up with stronger discontinuities as long as the function is continuous almost everywhere. Formally, it is assumed that f is locally Lipschitz continuous and continuously differentiable on an open dense subset D of $\Re^n$ and that there is a point $\tilde{x} \in \Re^n$ for which the set $L = \{x / f(x) \leq f(\tilde{x})\}$ is compact. The local Lipschitz hypothesis allows us to approximate the Clarke subdifferential [13] as explained in the sequel.

3) Clarke subdifferential approximation: For each $\varepsilon > 0$, one defines the multifunction $G_\varepsilon : \Re^n \rightarrow \Re^n$ by:

$$G_\varepsilon(x) = Co(\nabla f(x + \varepsilon B) \cap D) \qquad (20)$$

where $B = \{x \ / \ \|x\| \leq 1\}$ is the closed unit ball and $Co(\ )$ is the closed convex hull.

Figure 3 illustrates an example of a Clarke ε subdifferential approximation.

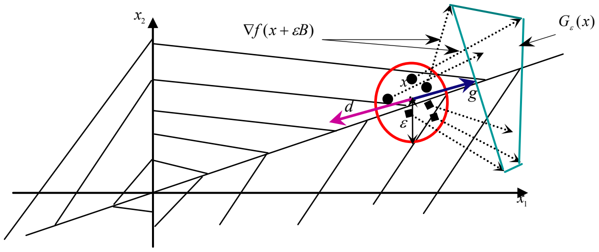

Fig. 3. Approximation of the Clarke ε subdifferential.

The sets $G_\varepsilon(x)$ can be used to give the following representation of the Clarke subdifferential of the function f at a given point x:

$$\bar{\partial} f(x) = \bigcap_{\varepsilon > 0} G_\varepsilon(x) \qquad (21)$$

The ideas used in the algorithm are based on the ε subdifferential introduced by Goldstein [14].

4*) Clarke ε subdifferential:* For each $\varepsilon > 0$, the Clarke ε subdifferential is given by

$$\bar{\partial}_\varepsilon f(x) = Co\left(\bar{\partial} f(x + \varepsilon B)\right) \qquad (22)$$

For: $0 < \varepsilon_1 < \varepsilon_2$ the following embedded inclusions are verified $\bar{\partial}_{\varepsilon_1} f(x) \subset G_{\varepsilon_2}(x) \subset \bar{\partial}_{\varepsilon_2} f(x)$.
Therefore, the Clarke ε subdifferential can be approximated by $G_\varepsilon(x)$, this is due to the hypothesis of almost everywhere differentiability of f, $G_\varepsilon(x)$ can be estimated by a finite spatial sampling:

$$G_\varepsilon(x) \approx Co\left(\bigcup_{i=1}^{m} (\nabla f(x + \varepsilon b_i) \cap D)\right) \quad \text{with } b_i \in B \qquad (23)$$

5*) Clarke ε stationary point:* A point x is Clarke ε stationary if $0 \in \bar{\partial}_\varepsilon f(x)$. In order to measure the proximity to Clarke ε stationarity, Burke [12] introduces the following scalar value:

$$\rho_\varepsilon(x) = dist(0 / G_\varepsilon(x)) \qquad (24)$$

IV. The ε Subdifferential Algorithm

An algorithm has been developed for the criteria which are differentiable almost everywhere. The constraint problems can be handled by an exact penalty, nondifferentiable weights can be also used. This algorithm is slightly different from the one developed on [12]. The basis differences are: addition of tests for managing numerical degenerated cases and the introduction of an isotropic point sampling of the hyperball for a better estimation of the ε subdifferential [15].

A. *Additional Notations:*

ε_k Sampling radius at the k^{th} iteration.
υ_k Optimality tolerance at the k^{th} iteration.
β Armijo parameter.

γ Backtracking reduction factor.
δ Optimality tolerance reduction factor.
μ Sampling radius reduction factor.

B. *Algorithm:*

Step 0: (Initialization)
Let $x^0 \in L \cap D$, $\gamma \in]0,1[$, $\beta \in]0,1[$, $\varepsilon_0 > 0$, $\upsilon_0 \geq 0$, $\mu \in]0,1]$, $\delta \in]0,1]$, $k = 0$ and $m \in \{n+1, n+2, n+3, ...\}$.

Step 1: (Approximation of the Clarke ε subdifferential)
Let $u^{k1}, u^{k2}, ..., u^{km}$ be sampled independently and uniformly from B, and set: $x^{k0} = x^k$ and $x^{kj} = x^k + \varepsilon_k u^{kj}$ for $j = 1, ..., m$.

If one of samples points $(j = 1, ..., m)$ verifies $x^{kj} \notin D$ **then go to** Step 1.
Else $G_k = Co\{\nabla f(x^{k0}), \nabla f(x^{k1}), ..., \nabla f(x^{km})\}$.
$\nabla f(x^{kj})$ is calculated by sensitivity method (for semi-infinite problems)

Step 2: (Compute a descent direction d^k)
Let $g^k \in G_k$ solution of the positive quadratic problem $g_k = \underset{g \in G_k}{\arg\min}\ dist(0 / G_k)$
If $\upsilon_k = \|g^k\| = 0$, **Then** Stop (ε stationarity).

Else
→ **If** $\|g^k\| \leq \upsilon_k$, **then** set $t_k = 0$, $\upsilon_{k+1} = \delta\upsilon_k$, $\varepsilon_{k+1} = \mu\varepsilon_k$ and **go to** Step 4.
→ **Else** $\upsilon_{k+1} = \upsilon_k$, $\varepsilon_{k+1} = \varepsilon_k$, $d^k = -g^k / \|g^k\|$

Step 3: (Compute a step length t_k)
$t_k = \max_{s \in \{0,1,2,...\}} \gamma^s \ / \ f(x^k + \gamma^s d^k) < f(x^k) - \beta\gamma^s \|g^k\|$

Step 4: (Update)
If $x^k + t_k d^k \in D$ **then** $x^{k0} = x^k + t_k d^k$, $k = k+1$ and **go to** Step 1.
Else, let $\hat{x}^k \in \hat{x}^k + \varepsilon_k B$ satisfying $\hat{x}^k + t_k d^k \in D$ and $f(\hat{x}^k + \gamma^s d^k) < f(x^k) - \beta\gamma^s \|g^k\|$, and **then** $x^{k0} = \hat{x}^k + t_k d^k$, $k = k+1$. **Go to** step 1.

V. Application to a Magnetic Suspension System

Let's consider the magnetic suspension device shown in figure 4. It consists of an iron pendulum in a vertical magnetic field created by an electromagnet. The related closed loop block diagram is depicted in figure 5, where z is the measured position of the pendulum compared with the sensor position center in an absolute reference frame and i is the output current signal of the actuator.
In a first time, the dynamic between the input voltage u of the actuator and its output current i is not considered. With a state vector $x = [z, \dot{z}, i]^T$, the model of the system is:

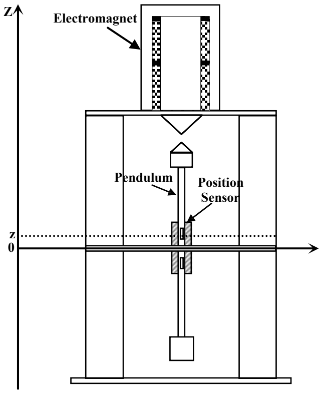

Fig. 4. Magnetic suspension device.

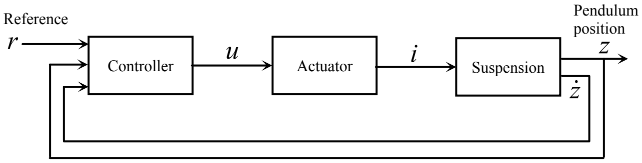

Fig. 5. Block diagram of the magnetic suspension system.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \dfrac{k}{m} \dfrac{x_3^2}{(c - x_1)^2} \\ x_3 = k_\upsilon u \\ y = x_1 \end{cases} \tag{25}$$

where m is the mass of the pendulum, k_υ is the amplification factor of the actuator, k is a positive constant and c denotes the nominal air gap. For a given desired constant position of the pendulum x_1^*, the equilibrium that we want to stabilize verifies:

$$x^* = \left[x_1^*, 0, \sqrt{mg/k}(c - x_1^*)\right]^T$$

Magnetic suspension system is an open loop unstable system, so then, in order to guarantee stable feedback, suitable control is needed. Backstepping method has been used. It allows to stabilize the system and to take in account strong nonlinear effect of the magnetic force.

A. *Backstepping Controller Design*

Backstepping approach consists in finding a strictly assignable Control Lyapunov Function (CLF), positive definite and radially unbounded that guarantees the global asymptotic stability of the system. Complete methodology can be found in [16] and [17].

Step 1: Consider the subsystem:

$$\dot{x}_1 = v_1(x_1) \tag{26}$$

In order to find the virtual control law $v_1(x_1)$ we introduce the CLF

$$V_1(x_1) = \frac{1}{2} e^2 \tag{27}$$

where e is the error signal $e = x_1 - x_1^*$.

Let's derivate (27), it comes:

$$\dot{V}_1(x_1) = e \cdot v_1(x_1) \tag{28}$$

By taking $v_1(x_1) = -\alpha_1 e$ with $\alpha_1 > 0$, we strictly assign the CLF $V_1(x_1)$.

Step 2: Now, we consider the whole magnetic suspension system:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \dfrac{k}{m} \dfrac{k_\upsilon^2 u^2}{(c - x_1)^2} \end{cases} \tag{29}$$

Let's consider the following CLF

$$V_2(x_1, x_2) = V_1(x_1) + \frac{1}{2}\alpha_3 (x_2 - v_1(x_1))^2, \quad \alpha_3 > 0 \tag{30}$$

As in step 1, we strictly assign the CLF $V_2(x_1, x_2)$ by setting

$$u^2 = \frac{m}{\alpha_3 k k_\upsilon^2}(c - x_1)^2 (\alpha_3 g - e(1 + \alpha_1\alpha_2) - x_2(\alpha_2 + \alpha_1\alpha_3)), \quad \alpha_2 > 0 \tag{31}$$

So as to obtain:

$$\dot{V}_2(x_1, x_2) = -\alpha_1 e^2 - \alpha_2 (x_2 + \alpha_1 e)^2 \tag{32}$$

This choice results in the closed loop system

$$\begin{cases} \dot{e} = x_2 \\ \dot{x}_2 = -\dfrac{1}{\alpha_3}((1 + \alpha_1\alpha_2)e + (\alpha_2 + \alpha_1\alpha_3)x_2) \end{cases} \tag{33}$$

A natural choice would have lead to the introduction of two parameters, classically the same as α_1 and α_2. Roughly speaking, two degrees of freedom seem sufficient for a second order system. A full study of this case ($\alpha_3 = 1$) shows that the corresponding variation of the damping coefficient ξ and the undamped frequency ω_n of the closed loop poles are restricted. The corresponding domain is given on figure 6. Our choice of CLF permits to reach the different dynamics of second order system (overdamped and underdamped) through the three parameters α_i. They permit to assign freely the poles of the closed loop system.

B. Simulation Results

The required performances are commonly expressed by the overshoot (< 20 %), the settling time (< 0.06 s), the tracking error accuracy and the limitation of the control ($|u| \leq 5\,\mathrm{V}$). These specifications are formulated using template forms and the general optimization problem is stated as follow:

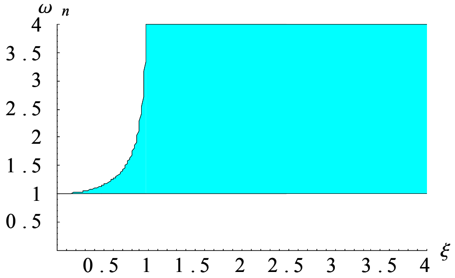

Fig. 6. The assignable domain of the closed loop linear system if $\alpha_3 = 1$.

$$J(\theta) = \sum_{i=1}^{n_t}\left(\left|y(\theta,t_i) - Y_{\max}(t_i)\right|_+ + \left|Y_{\min}(t_i) - y(\theta,t_i)\right|_+\right) + \sum_{i=1}^{n_t}\left(\left|u(\theta,t_i) - U_{\max}(t_i)\right|_+ + \left|U_{\min}(t_i) - u(\theta,t_i)\right|_+\right) \tag{34}$$

Using the ε subdifferential algorithm, the last problem is solved. It's not really a RE-tuning situation but method is nevertheless very efficient for controllers design. Figure 7 exhibits the optimal solution z (upper), and the input voltage u (lower) for $x_1^* = 10^{-3} m$. It may be observed that simulation results are in concordance with the required performances. Note that these good performances of the controller are dependent on the full knowledge of parameters in the model (here: $m = 0.0844\ kg$, $k = 0.005$, $k_\upsilon = 0.1$, $c = 0.011\ m$ and $g = 9.81\ m/s^2$). The obtained optimal parameters are:

$\theta_{opy} = [\alpha_1^{opt}, \alpha_2^{opt}, \alpha_3^{opt}]^T = [128,\ 8 \cdot 10^{-5},\ 5 \cdot 10^{-5}]^T$.

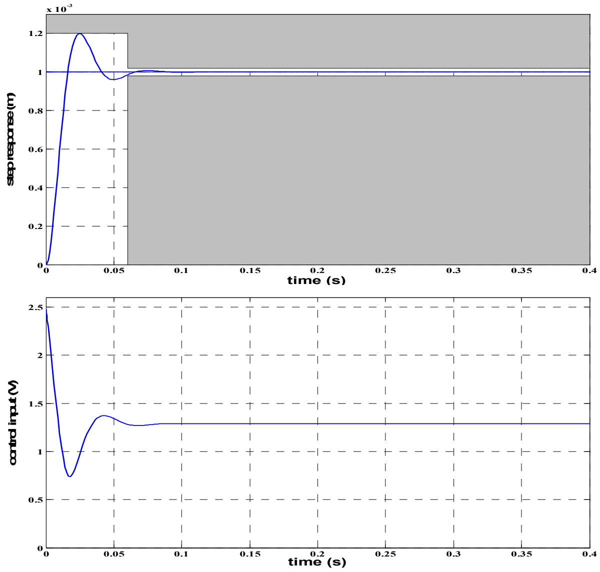

Fig. 7. Temporal responses of the system with the specifications $T_s \leq 6.10^{-2}(s)$, $D \leq 20\%$ and $|u(t)| \leq 5\ (V)$ ·

C. Controller Retuning

Dynamic of the actuator will now be taken into account. It is modelized by a first order model with a time constant τ and the same amplification factor k_υ. Then, the complete model of the magnetic suspension system is

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \dfrac{k}{m} \dfrac{x_3^2}{(c - x_1)^2} \\ \dot{x}_3 = (-x_3 + k_v u)/\tau \\ y = x_1 \end{cases} \tag{35}$$

The previous control law is used (equation 31). Results should be similar if the actuator is quick enough. Maintaining the same specifications, results are shown in figure 8 (dashed line).

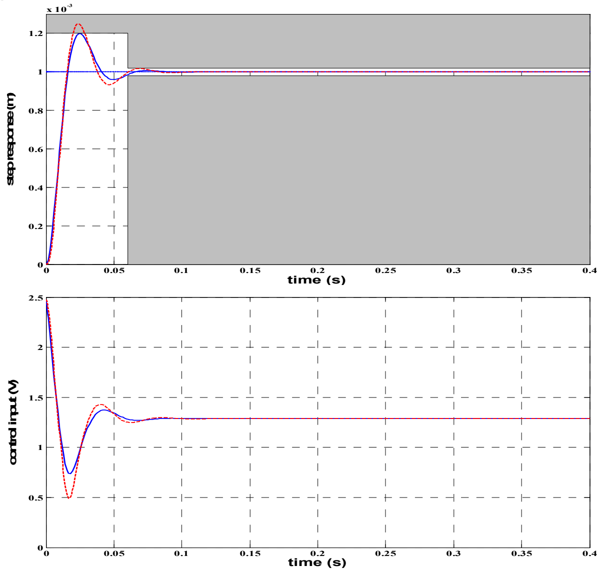

Fig. 8. Temporal responses comparison with and without actuator.

We remark that the specifications are no longer fulfilled; the templates of the step response are not verified. For this reason, the developed ε subdifferential algorithm is now used to retune the backstepping control law. The results of this operation are shown in figure 9.

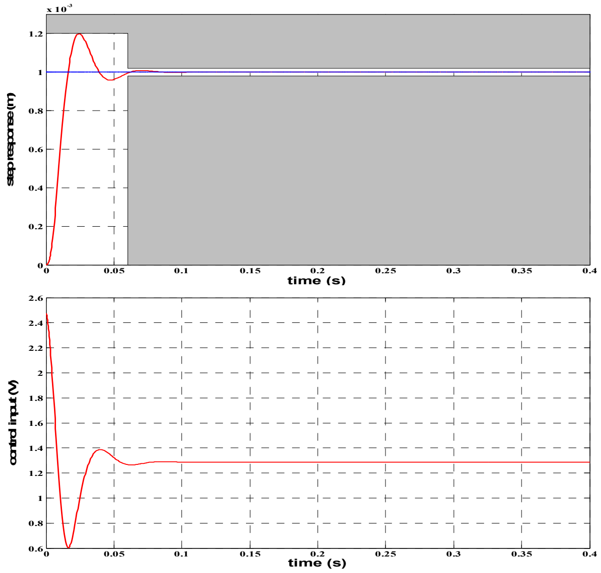

Fig. 9. Temporal responses of the new system with the specifications $T_s \le 6.10^{-2}(s)$, $D \le 20\%$ and $|u(t)| \le 5\ (V)$.

The new optimal parameters are given by $\theta_{opy} = \left[142, 7 \cdot 10^{-5}, 45 \cdot 10^{-6}\ \right]^T$ and the final results reach all specifications.

VI. Conclusion

To hold into account generic specifications for controller retuning, an exact formulation of the demands must be used. Corresponding criteria are generally nonsmooth, which often lead to difficult optimization problems. This study presents an adapted method: the Clarke subdifferential approximation and the parametric sensitivity proprieties of systems allow to formalize the approach and to define mathematical objects that can be numerically estimated with reliability. Concerning application, backstepping is not only used to stabilize the system but also to reach performances. An extra degree of freedom has been introduced for this purpose and can be easily managed via optimization approach. Naturally well suited for retuning, this approach appears to be also very efficient for a direct controller design. So, ε subdifferential optimization represents an effective tool in computer-aided design for controllers retuning.

References

[1] H. Kwakernaak and R. Sivan, *Linear Optimal Control System*. New York: Willey-Interscience, 1972.

[2] B.D.O. Anderson and J.B. Moore, *Linear Optimal Control.* Englewood Cliffs. New Jersey: Prentice-Hall, 1972.

[3] J.C. Doyle, K. Zhou and B. Bodenheimer, "Optimal control with mixed H_2 and H_∞ performance objective," in *Proc. of American Control Conference*, Pittsburgh, Pennsylvania, 1989, pp. 2065-2070.

[4] K. Zhou, J.C. Doyle and K. Glover, *Robust and Optimal Control*. New Jersey: Prentice-Hall, 1996.

[5] B.A Francis and J.C Doyle, "Linear control theory with an H_∞ optimality criterion," *SIAM Journal on Control and Optimization,* vol. 25, pp. 815-844.

[6] S. Boyd and C. Barrat, *Linear Controller Design: Limits of Performance.* Englewood Cliffs. New Jersey: Prentice-Hall, 1991.

[7] B.T. Polyak, *Introduction to Optimization.* Translations Series in Mathematics and Engineering, Optimization Software Inc, New York, 1987.

[8] E. Rosenwasser and R. Yusupov, *Sensitivity of Automatic Control Systems.* CRC Press Control Series, Boca Raton , 2000.

[9] R. Tomovic and M. Vukobratovic, *General Sensitivity Theory*. New York: Elsevier, 1972.

[10] E. Walter and L. Pronzato, *Identification de Modèles Paramétriques à partir de Données Expérimentales*. Masson, Paris, 1994.

[11] F.H. Clarke, *Optimization and Nonsmooth Analysis.* John Wiley, New York, 1983. Reprinted by SIAM, Philadelphia, 1990.

[12] J.V. Burke, A.S. Lewis and M.L. Overton, "A robust gradient sampling algorithm for nonsmooth, nonconvex optimization," *Revised version to appear In SIAM Journal on Optimization*, June 2004.

[13] J.V. Burke, A.S. Lewis and M.L. Overton, "Approximating subdifferentials by random sampling of gradients," *Mathematics of Operations Research*, pp. 567-584, 2002.

[14] A.A. Goldstein, "Optimization of Lipschitz continuous functions," *Mathematical Programming*, pp. 14-22, 1977.

[15] D.E. Knuth, *The Art of Programming* – vol. 2: *Seminumerical Algorithms*. Addison-Wesley Publishing Company, 1973.

[16] M. Krstic, I. Kanellakopoulos and P.V. Kokotovic, *Nonlinear and Adaptative Design.* New York: Prentice-Hall, 1995.

[17] L. Praly (2004) An Introduction to some Lyapunov Designs of Global Asymptotic Stabilizers. [Online]. Available: http://cas.ensmp.fr/Publications/Donald/Stabilisation.