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

Introducing robustness in iterative control

S. Bittanti M.C. Campi S. Garatti

Abstract— **Iterative control has been widely studied in recent years as an efficient methodology for the design of highly-performing controllers of complex plants. The idea behind iterative design is that, when the plant is exceedingly complex, the design of the controller in one shot is hazardous. Instead, one can perform a sequence of closed-loop identification and controller design steps, aiming at progressively learning how to increase the control performance through experience.**

In this paper, we introduce a new iterative control scheme which explicitly accounts for the presence of uncertainty in the plant description at each step (iterative *robust* control). Our contention is that introducing robustness in iterative schemes permits to quickly improve performance through steps, while preserving the robust stability of the closed-loop system.

I. Introduction

A. Why iterative control?

Consider the problem of designing a highly-performing controller C for an uncertain (possibly nonlinear) plant P (see Figure 1). We suppose that the control performance is evaluated by means of a given control cost $J(C,P) \geq 0$ such that the lower J the better the performance. The final objective is to find a controller C that guarantees a given performance level: $J(C,P) \leq k$.

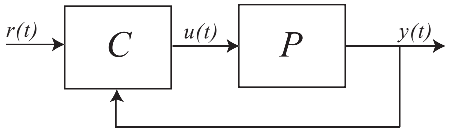

Fig. 1. Closed-loop system.

The main feature of the control problem under consideration is that the plant dynamics is assumed to be *unknown*, a situation which occurs in many practical engineering problems. A typical way to deal with uncertainty in the plant dynamics is to resort to identification methods to obtain a model $\widehat{P}$ of the plant. Then, the controller is designed based on $\widehat{P}$. This way of proceeding, however, calls for some care. Indeed, it is well known, [9], [13], that identifying the plant dynamics in one-shot may often result in a model which is unsuitable for controller design purposes. The reason is that it is a-priori difficult to select a suitable model class which achieves a sensible compromise between plant complexity and the limitations posed by finiteness of the data set (bias vs. variance error trade-off). Moreover, when the plant input can be manipulated, designing an experiment able to reveal at best the plant characteristics that are relevant for controller design can be difficult to obtain.

A well recognized fact, [9], [13], is that not all the plant characteristics are important for closed-loop performance. Thus, the goal of the identification step is that of accurately identifying only those (usually few) plant dynamical features which are relevant to control design. Though the system-model mismatch may still remain large, highly-performing controllers can then be designed as the identified model turns out to be properly tuned towards control objectives (*identification for control*). Clearly, the problem here is how to perform the identification experiment so as to identify the "plant dynamical features which are relevant to control design".

When P is a simple linear system, control design basically requires only the knowledge of the plant frequency behavior over an interval of frequencies around the desired closed-loop crossover frequency. Such information can be retrieved by suitably exciting the system or by pre-filtering of the available data, [12].

When instead plant P is complex and/or presents nonlinearities, tuning the identification method towards the final control design objective can be difficult. These are the circumstances in which iterative procedures prove powerful, [9], [11], [13].

In general terms, an iterative control scheme goes as follows. Suppose that a controller C_{i-1} has been already designed, whose performance is however not good enough. The controller C_{i-1} is updated to C_i through the following steps:

1. data are collected in *closed-loop* with C_{i-1} in place, and a model $\widehat{P}_i$ is estimated;
2. a new controller C_i is designed based on $\widehat{P}_i$;
3. C_i is connected to the plant and the performance is checked: if $J(C_i,P) \leq k$ then the procedure is halted; otherwise $i = i+1$ and the procedure is repeated from step 1.

The controller validation at step 3 is performed by means of experimental data collected while the real plant is operated with C_i.

As it appears, iterative control consists in a sequence of intertwined closed-loop identification and control design steps. The goal is to arrive to the identification of the plant dynamical features relevant to control design through small adjustments. The effectiveness of this idea can be better understood through a metaphor, as explained in the

Paper supported by the MIUR National Research Project "Innovative Techniques for Identification and Adaptive Control of Industrial Systems", and partially supported by CNR-IEIIT

S. Bittanti and S. Garatti are with the Dipartimento di Elettronica ed Informazione, Politecnico di Milano, p.zza L. da Vinci 32, 20133 Milano, Italy. E-mail: {bittanti,sgaratti}@elet.polimi.it

M.C. Campi is with the Dipartimento di Elettronica per l'Automazione, Università di Brescia, via Branze 38, 25123 Brescia, Italy. E-mail: campi@ing.unibs.it

0-7803-9568-9/05/$20.00 ©2005 IEEE

following.

Suppose a person is not too far from a cliff. It is dark and s/he can only light the scenery up by means of few matches he has in his pocket. The objective is to get as close as possible to the northernmost point on the edge of the cliff, without falling down (see Figure 2).

A basic decision to be taken is whether to use the matches altogether or else one after the other. By striking the matches altogether, the hope is to bring into light the whole route to the cliff edge (full plant identification). However, there would be the risk that the light would be too dim to reveal where the edge is located exactly. So, the matches would probably be wasted if used this way. A wiser strategy consists instead in lighting one match at a time so as to reveal the scene in the vicinity of the current position, and then moving a little step towards north in the visible area. As the person moves in the new position, s/he cannot proceed any further in a safe way. At this point, another match is used so that the next step can be performed safely, and so on until the northernmost point is reached.

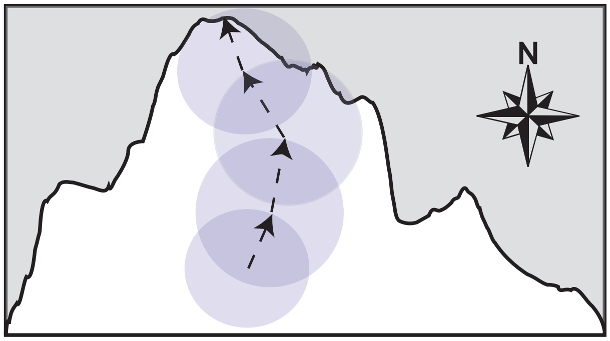

Fig. 2. An iterative procedure for the cliff problem.

Iterative control works similarly. The closed-loop setting at step 1 is introduced to avoid a full plant identification. Indeed, in closed-loop only some features of the plant are excited and estimated. Thus, during the first iterations – when poorly-performing controllers are connected to the plant – only certain plant dynamical features are unveiled. This information can be used to adjust in step 2 the current controller, so moving a little step in the direction of improving the control performance. Iterating this scheme, the desired control performance can be eventually reached.

One important aspect which is worth emphasizing in the metaphor is that the northernmost point was reached by enlightening not all the cliff but only a path connecting the initial position to the target. Therefore, the identification effort is spent towards achieving the final objective without learning "too much" of the system. Similarly, when improving step by step the control system performance, we need not to explore the whole plant dynamics in general.

B. The need for robustness in iterative control

Iterative control has been intensively studied in the last decade and there is a variety of iterative techniques with different and specific features, [9], [11], [13]. Yet, as shown in [1] and [2], a common feature is the need for cautious adjustments at each controller update step. This can be easily understood considering that, at each iteration i, only a partial description of the plant becomes available, and, consequently, the controller C_i will have to be designed on a conservative ground (*catious controller*).

In many iterative schemes the model $\widehat{P}_i$ at step 1 of the procedure is simply a nominal model of the plant with no concern for its reliability. Consequently, when the controller C_i is designed at step 2, one has no hints of its range of validity. This circumstance reflects into an over-conservative use of the model.

In e.g. the well known "windsurfer" approach, [11], the fact that the model reliability is unknown is taken care of by splitting step 2 into a number of sub-steps: using the model identified at step 1 and without updating it, the controller is progressively tuned so as to increase the control performance; at any sub-step, the designed controller is tested on the real plant to avoid reaching the closed-loop stability limit. When it is detected that a further performance improvement is likely to generate instability, the model $\widehat{P}_i$ is no longer deemed reliable and a new model is identified. This means that step 2 is halted and the procedure moves on to step 3.

Unfortunately, this way of proceeding has a drawback: each intermediate controller has to be tested on the real plant and this requires to access the plant many times for experiment. This results in a relatively long and expensive design procedure.

The drawbacks in the above approach can be alleviated by resorting to *robust* iterative control techniques, i.e. iterative schemes which explicitly account for the presence of uncertainty in the control design phase at step 2. This can be obtained by replacing steps 1 and 2 with the following ones:

$1'$. from the data collected in closed-loop:

   $1'.a$ estimate a nominal model;

   $1'.b$ estimate the model uncertainty;

$2'$. design the *best possible robust controller* C_i according to the existing level of uncertainty.

The idea behind points $1'$ and $2'$ above can be explained as follows. At iteration i, a sensible selection of the controller has to meet two different and contrasting objectives:

- on the one hand, the controller has to be cautious to avoid a possible destabilization of the control system;
- on the other hand, it should not be overconservative, otherwise the corresponding performance improvement is not significant.

The robust controller design at step $2'$ performs in a single step the best compromise between the above two objectives according to the present level of uncertainty. In this way, the achieved performance rapidly improves through iterations, while preserving the robust stability of the closed-loop system. This is in contrast with standard iterative schemes, where neglecting uncertainty has the consequence of requiring the splitting of the control design step into a number of sub-steps, with corresponding experimental over-effort.

The iterative algorithm outlined above describes the essential idea of iterative robust control at a general level. This idea is more concretely developed in the subsequent sections of this paper by performing specific choices for the implementation of steps $1'$ and $2'$. See also [3] and [6] for extra discussion.

C. Structure of the paper

The estimation of the nominal model and model uncertainty in step $1'$ is discussed in Section II, while in Section III our robust control approach, based on a probabilistic robust method, is presented. The complete iterative scheme is wrapped-up in Section IV, and a simulation example is finally given in Section V.

II. The identification and uncertainty estimation step

For the identification of the nominal model in step $1'.a$, a typical choice, [12], is to resort to standard Prediction Error Methods (PEM) or to Correlation Approach techniques (as e.g. Instrumental Variable (IV) identification). The uncertainty evaluation is performed by means of the corresponding asymptotic theory, [12]. As is well known, in this theory the uncertainty is assessed through a probability density $f(\vartheta)$ ($f: \Theta \to \mathbb{R}$, where $\vartheta \in \Theta \subseteq \mathbb{R}^n$ is a vector parameterizing the model class) describing the likelihood that the model corresponding to ϑ is the true system. In many cases, this probability density turns out to be Gaussian with mean and variance which can be estimated from the available data.

Although all these methods have a long-standing history in the literature, in our framework some care is needed.

First of all, for the success of the iterative procedure it is usually advisable to consider different classes of models for the two steps $1'.a$ and $1'.b$. The reason lies in the fact that the nominal model and the uncertainty description play different roles in the subsequent control design step $2'$ (see also Section III where step $2'$ is discussed). Precisely, the nominal model is used to design a nominal controller and therefore a class of low order models is advisable to obtain controllers of low complexity. The uncertainty description, instead, is used to detune the previous controller parameters so as to meet a robust stability requirement. It is clear that in this second phase it is important to use a high order model class so as to capture the uncertainty besetting the nominal model.

In the sequel, we will denote by $\mathscr{M}_\lambda$, $\lambda \in \Lambda \subseteq \mathbb{R}^m$, the parametric class of low order models, whereas $M(\widehat{\lambda}_i)$ will denote the nominal model identified at iteration i. Finally, $\mathscr{P}_\vartheta$, $\vartheta \in \Theta \subseteq \mathbb{R}^n$ will be the full-order model class used for model quality assessment.

Before proceeding, one should be also aware that the choice of the model class $\mathscr{P}_\vartheta$ is somehow critical since, for certain classes of models and in condition of poor excitation, the asymptotic theory of system identification may lead to unreliable results. Indeed, as shown in [8], there are situations where the estimated density turns out to be extremely peaky – suggesting that uncertainty is restricted – even though the real plant dynamics is located far from the peak. Note also that poor excitation conditions are likely to occur in iterative control, especially during the first iterations when the controller is over-conservative. According to the discussion in [8] and [7], the above problem is avoided when IV identification is used. When one instead resorts to PEM methods, the class of models $\mathscr{P}_\vartheta$ has to be carefully chosen. See [8] and [7] for a more detailed discussion.

III. The robust controller design step

The objective of this section is to describe how the information supplied by step $1'$ (i.e. $M(\widehat{\lambda}_i)$ and $f_i(\vartheta)$) can be used to design the "best possible robust controller C_i" in step $2'$.

A two-stage design method is here considered:

a. Design a *nominal controller* $\widehat{C}_i$ based on the nominal model $M(\widehat{\lambda}_i)$.

b. *Detune* the nominal controller parameters according to the existing level of uncertainty as described by $f_i(\vartheta)$, so as to meet a robust stability requirement.

These two points are now discussed in order.

The nominal controller $\widehat{C}_i$ is typically obtained by optimizing the control cost $J(C, M(\widehat{\lambda}_i))$ with respect to C. The result is a high performing controller for the identified nominal model $M(\widehat{\lambda}_i)$ (i.e. $J(\widehat{C}_i, M(\widehat{\lambda}_i)) \leq k$), which, however, is also very sensitive to model inaccuracy. At the beginning, when $M(\widehat{\lambda}_i)$ is not properly tuned to the the plant dynamics, $\widehat{C}_i$ can even destabilize P.

The detuning of the nominal controller is achieved by modifying the structure of $\widehat{C}_i$ (introducing e.g. new poles and/or zeroes; changing some of the controller parameters; etc.) so obtaining a new controller $C_i(\gamma)$, where $\gamma \in \Gamma \subseteq \mathbb{R}^l$ is the vector of all the parameters introduced in the detuning phase. By varying γ, robustness may be incorporated in the nominal controller so as to make $C_i(\gamma)$ suited to stabilize the plant even though the latter is different from the nominal model. The price to pay is typically a degradation of the nominal performance.

As is obvious, the value of γ for the current iteration has to be chosen according to the existing level of uncertainty as described by $f_i(\vartheta)$. In this work, following [5], we adopt an approach which is robust in probability.

Denote by (γ, ϑ) the closed-loop system obtain when $C_i(\gamma)$ is connected to a generic system $P(\vartheta)$ in the uncertainty class. For simplicity, we assume that the smaller is $\|\gamma\|$, the weaker the effect of the detuning on the nominal controller. Moreover, $C_i(0) = \widehat{C}_i$.

The optimal robust controller $C_i(\gamma^o)$ for the present level of uncertainty is then obtained as the solution of the following optimization problem.

$$\begin{array}{l} \min_{\gamma \in \Gamma} \|\gamma\| \\ \text{subject to } \mathbb{P}\{(\gamma, \vartheta) \text{ is not stable}\} \leq \alpha. \end{array} \tag{1}$$

Here, α is a parameter belonging to $(0,1)$ and $\mathbb{P}\{A\}$ denotes the probability of the event A with respect to the density function $f_i(\vartheta)$. Note that $\mathbb{P}\{(\gamma, \vartheta) \text{ is not stable}\} = \int_\Theta \mathbf{1}_{(\gamma,\vartheta) \text{ is not stable}} \cdot f_i(\vartheta) \cdot \mathrm{d}\vartheta$, where $\mathbf{1}_\cdot$ denotes the indicator function. $\mathbb{P}\{(\gamma,\vartheta)$ is not stable$\}$ is thus a function of the sole variable γ.

As it appears, in this approach one tries to keep the detuning effect as moderate as possible, provided that a robust requirement on stability is satisfied. The main feature here is that the guarantee on stability is given in probability, with a level of risk no greater than α. Obviously, by selecting α to be a small number, the controller can be made as safe as wanted. On the other hand, an exceedingly small value for α may result in an overconservative controller and this might slow down the performance improvement through iterations. To this purpose, the degree of freedom in the choice of α is a point of strength of the probabilistic approach as one can tune the robustness level depending on the application at hand.

Another important feature of the probabilistic robust controller is that it can be computed at a low effort by means of a randomized approach (see e.g. [14] and [4]). For the sake of completeness, a short resume of the results useful in this iterative control context is provided in the following.

Let $\{\gamma_1,\dots,\gamma_p\}$ be p samples of Γ. We search for the best controller parameter among $\{\gamma_1,\dots,\gamma_p\}$, rather than over the entire feasible set Γ. We suppose that the samples $\{\gamma_1,\dots,\gamma_p\}$ are drawn in such a way they densely cover the feasible set Γ.

In order to compute γ^o, an empirical counterpart of the probability $\mathbb{P}$ is used. Precisely, define

$$\widehat{\mathbb{P}}\{(\gamma,\vartheta) \text{ is not stable}\} = \frac{1}{q}\sum_{k=1}^{q}\mathbf{1}_{(\gamma,\vartheta_k)\text{ is not stable}},$$

where ϑ_k's are q parameter vectors independently extracted from Θ according to the probability density $f_i(\vartheta)$. Note that $\widehat{\mathbb{P}}\{(\gamma,\vartheta)$ is not stable$\}$ is again a function of the sole γ.

The approximation introduced when $\widehat{\mathbb{P}}$ is used in place of $\mathbb{P}$ can be kept moderate by suitably selecting the number q of extracted ϑ_k's. The well known Hoeffding theorem ([14], [4]) can be used to this aim.

Theorem 1 (Hoeffding): Fix two real numbers $\varepsilon > 0$ and $\delta > 0$. If

$$q > (2\varepsilon^2)^{-1}\ln(2p/\delta), \tag{2}$$

then $\mathbb{P}\{(\gamma,\vartheta)$ is not stable$\} \leq \widehat{\mathbb{P}}\{(\gamma,\vartheta)$ is not stable$\} + \varepsilon$, $\forall\,\gamma \in \{\gamma_1,\dots,\gamma_p\}$, with a probability greater than $1-\delta$. □

Remark 1: Theorem 1 says that $\mathbb{P}\{(\gamma,\vartheta)$ is not stable$\}$ can be approximated by $\widehat{\mathbb{P}}\{(\gamma,\vartheta)$ is not stable$\}$ with arbitrary precision as long as the number q of ϑ_k extractions is sufficiently high. Note however that the result holds true with a certain probability $1-\delta$ only. This is a consequence of the fact that $\widehat{\mathbb{P}}\{(\gamma,\vartheta)$ is not stable$\}$ is a random element depending on the extracted $\vartheta_1,\dots,\vartheta_q$: $|\widehat{\mathbb{P}}\{(\gamma,\vartheta)$ is not stable$\} - \mathbb{P}\{(\gamma,\vartheta)$ is not stable$\}|$ can be smaller than ε for some multi-samples and not for others, and δ refers to the probability of extracting a "bad" multi-sample $\vartheta_1,\dots,\vartheta_q$. Finally, note that q depends on the logarithm of δ so that a very small values of δ can be forced-in without lifting q too much. □

Remark 2: Note that, in contrast to other non-random numerical methods which can be used to compute $\mathbb{P}\{(\gamma,\vartheta)$ is not stable$\}$, q does not depend on the size n of the space in which Θ is embedded. This allows to keep the computational effort of randomized algorithms small. □

The optimal controller robust in probability can be computed at low computational effort by solving the following optimization problem in place of (1):

$$\begin{array}{l}\min_{\gamma\in\{\gamma_1,\dots,\gamma_p\}}\|\gamma\| \\ \text{subject to } \frac{1}{q}\sum_{k=1}^{q}\mathbf{1}_{(\gamma,\vartheta_k)\text{ is not stable}} \leq \alpha-\varepsilon.\end{array}$$

Thanks to Theorem 1, the found γ^o is such that $\mathbb{P}\{(\gamma,\vartheta)$ is not stable$\} \leq \alpha$ holds with high probability greater than $1-\delta$.

Remark 3: Before proceeding, we are well advised to raise a delicate point, namely the *curse of dimensionality*. Indeed, in order to explore the entire controller set, the number p of samples $\{\gamma_1,\dots,\gamma_p\}$, must increase exponentially with l, the dimensionality of the controller parameter space. In this way, p becomes very large even for relatively small values of l and, correspondingly, the computational burden of the algorithm for the search of the best controller becomes rapidly intractable. However, in contrast to what happened for Θ, the dimensionality of Γ is not required to be large. In fact, as we will see in Section V, in many cases $l=1$ suffices, so that this problem automatically cools down. □

IV. A COMPLETE ITERATIVE ROBUST CONTROLLER DESIGN SCHEME

By complementing the algorithm described in Section I with all the points discussed in previous two sections, we obtain the following iterative robust algorithm:

0. (*initialization step*) connect an initial controller C_0 in feedback to the plant. Choose the model class $\mathscr{M}_\lambda$ along with the model class $\mathscr{P}_\vartheta$. Choose also the detuning parameter space Γ and sample it with $\{\gamma_1,\dots,\gamma_p\}$. Select α, ε and δ, and let $q > \frac{1}{2\varepsilon^2}\ln\frac{2p}{\delta}$. Set $i=1$;
1. from the data collected in closed-loop:
   1.*a* identify a low-order model $M(\widehat{\lambda}_i)$ in $\mathscr{M}_\lambda$;
   1.*b* estimate the probability density $f_i(\vartheta)$ over the high order class of models $\mathscr{P}_\vartheta$;
2. design $\widehat{C}_i$ based on $M(\widehat{\lambda}_i)$, and from $\widehat{C}_i$ build $C_i(\gamma)$. Extract ϑ_k, $k=1\dots q$, according to $f_i(\vartheta)$ and let
$$\begin{array}{rcl}\gamma^o & = & \arg\min_{\gamma\in\{\gamma_1,\dots,\gamma_p\}}\|\gamma\| \\ & & \text{subject to } \frac{1}{q}\sum_{k=1}^{q}\mathbf{1}_{(\gamma,\vartheta_k)\text{ is not stable}} \leq \alpha-\varepsilon;\end{array}$$
3. connect $C_i(\gamma^o)$ to the plant and check for the result: if $J(C_i(\gamma^o),P) \leq k$ then stop; else $i=i+1$ and go to 1.

V. APPLICATION EXAMPLE

In this section an application example of the iterative algorithm is presented. This application example has been chosen for its simplicity in order to focus on some issues of the new iterative controller design scheme rather than on technical details. Many implementation features discussed herein are of general breath.

A. The plant description

Consider the Grenoble transmission system presented in [10]. This system is constituted by three pulleys connected by two elastic belts as shown in Figure 3. The system input $u(t)$ is the angular position of the first pulley, while the output $y(t)$ is the angular position of the third pulley. The control objective was to make the angular position of the third pulley as close as possible, over a suitable bandwidth, to a given reference signal $r(t)$ (tracking control problem).

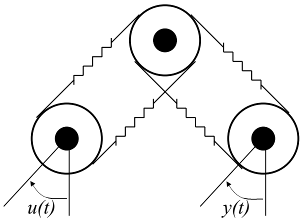

Fig. 3. The Grenoble transmission system.

In the simulation, we assumed to work with a sampled model of the Grenoble transmission system, i.e. the input-output dynamic behavior of the plant was described by the following discrete-time linear transfer function (see [10]):

$$P(z) = \frac{B_P(z)}{A_P(z)} = \frac{0.033z + 0.054}{z^4 - 2.85z^3 + 3.72z^2 - 2.65z + 0.87}.$$

Such transfer function is characterized by two pairs of complex conjugate stable poles, giving rise to two resonant peaks. A zero outside the unit circle (non minimum phase system) is also present.

At the beginning, $P(z)$ was operated with $C_0(z) = 0.05 \cdot \frac{z-0.9}{z-1}$, a (linear) PI controller which resulted in a stable but slow closed-loop system. A 1-degree-of-freedom control scheme where the controller is fed by the difference between the reference and the actual output was adopted.

B. Identification and uncertainty estimation

As required by the iterative approach, identification was performed in closed-loop. A square wave with period $T = 100$ and amplitude equal to 1 was used as reference input, and data collection lasted $N = 3000$ data points. In addition, during the identification phase, the system output was corrupted by an additive noise $d(t) = \frac{z-2}{z-0.9}e(t)$, where $e(t) = WGN(0, 0.0001)$ (WGN = White Gaussian Noise). Note that $d(t)$ is a highly-correlated stochastic noise as it is typical of many real applications. Its standard deviation is 0.026.

The nominal model $M(\widehat{\lambda}_i)$ of reduced complexity was identified in the following class of ARMAX(4,2,4) models:

$$\mathscr{M}_\lambda = \left\{ y(t) = \frac{B(z,\lambda)}{A(z,\lambda)}u(t) + \frac{C(z,\lambda)}{A(z,\lambda)}\eta(t) \right\},$$

where $\eta(t) = WN(0, \sigma^2)$ and λ is the vector of the coefficients of A, B, C.

As for the estimation of $f_i(\vartheta)$, the following high-order model class $\mathscr{P}_\vartheta$ was considered

$$\mathscr{P}_\vartheta = \left\{ y(t) = P(z,\vartheta)u(t) + v(t) \right\},$$

where $v(t)$ is a noise process and $P(z,\vartheta)$ is parameterized through a Finite Impulse Response (FIR) filter, i.e. $P(z,\vartheta) = \vartheta_1 z^{-1} + \vartheta_2 z^{-2} + \ldots + \vartheta_n z^{-n}$ with $n = 100$.

Identification was performed through the Instrumental Variable (IV) method, and the probability density $f_i(\vartheta)$ was evaluated by resorting to the asymptotic theory of IV techniques. The chosen $\mathscr{P}_\vartheta$ presents the following advantages:

1. The asymptotic theory for IV does not suffer from reliability problems, see [7].
2. High order FIR models are well suited to provide a full description of the true plant since the number n of parameters necessary to describe $P(z)$ can be determined in real applications by simply inspecting the impulse plant response.

We finally recall that selecting n to be large has no consequence on the randomized procedure of Section III.

C. Nominal controller and detuning

As nominal controller we used a linear deadbeat controller which can be obtained from the identified u to y nominal transfer function $B(z,\widehat{\lambda}_i)/A(z,\widehat{\lambda}_i)$ as follows:

$$\widehat{C}_i(z) = \frac{A(z,\widehat{\lambda}_i)}{B(1,\widehat{\lambda}_i)z^k - B(z,\widehat{\lambda}_i)},$$

where $k = 4$ is the order of $A(z,\widehat{\lambda}_i)$. If $B(z,\widehat{\lambda}_i) = B_P(z)$ and $A(z,\widehat{\lambda}_i) = A_P(z)$ (i.e. the real plant $P(z)$ is exactly identified), the complementary sensitivity function obtained when $\widehat{C}_i(z)$ is connected with $P(z)$ is $\widehat{F}_i(z) = B(z,\widehat{\lambda}_i)/B_i(1,\widehat{\lambda}_i)z^k$, a FIR system. This means that the reference signal is tracked in a finite number of steps (the controller is high-performing *in the nominal case*). When instead $P(z)$ is not correctly identified, $\widehat{C}_i(z)$ may even lead to instability.

The detuning was obtained with a well known technique in the Internal Model Control (IMC) context (see e.g. [11]). Precisely, $C_i(z,\gamma)$ was derived from $\widehat{C}_i(z)$ as follows:

$$C_i(z,\gamma) = \frac{A(z,\widehat{\lambda}_i)(1-\gamma)^k}{B(1,\widehat{\lambda}_i)(z-\gamma)^k - B(z,\widehat{\lambda}_i)(1-\gamma)^k},$$

where $\gamma \in [0,1)$. Note that $C(z,0) = \widehat{C}_i(z)$.

Let $F_i(z,\gamma)$ be the complementary sensitivity function obtained when $P(z)$ is operated with $C_i(z,\gamma)$. Then, the following comments are in order. First, the $F_i(z,\gamma)$ steady-state gain is always 1. Second, when $\gamma \to 1$, the poles of $F_i(z,\gamma)$ tend to the poles of $A_P(z)B_i(1,\widehat{\lambda}_i)(z-\gamma)^k$, which are stable provided that $P(z)$ is stable, independently of the mismatch between $P(z)$ and the identified model (*robust stability*). On the other hand, when $\gamma \to 1$, the dominant poles are those placed in γ, which means that the control system response is very slow. Thus, altogether, γ plays the role of a detuning parameter: as $\gamma \to 0$, nominal performance are achieved, whereas $\gamma \to 1$ leads to a degradation of the performance but also guarantees internal stability of the closed-loop system.

D. Simulation results

By applying the iterative controller design scheme of Section IV with the implementation choices described above to the Grenoble system we achieved to the following results. The reduced order u to y transfer function $B(z,\widehat{\lambda}_1)/A(z,\widehat{\lambda}_1)$ estimated at the first iteration (i=1) is depicted in Figure 4.

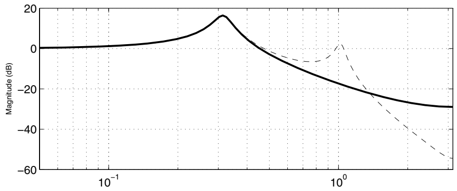

Fig. 4. Estimated nominal model at the first iterations and true plant Bode diagrams (continuous and dashed lines, respectively).

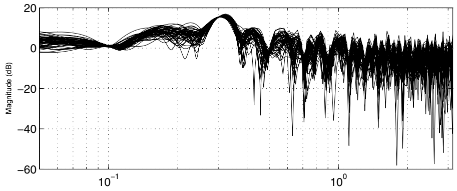

Fig. 5. Uncertainty at the first iteration (FIR models).

As for the estimated probability density, Figure 5 represents the Bode plot of some models extracted according to $f_1(\vartheta)$. The uncertainty estimated at iteration 1 was quite large.

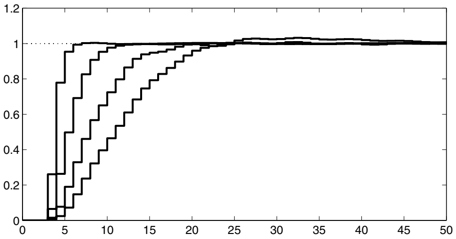

Fig. 6. Step response of the closed loop – $i = 1,\ldots,4$.

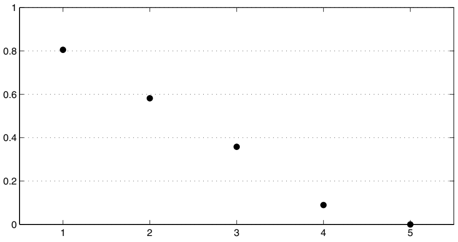

Fig. 7. $\overline{\gamma}_i$ at each iteration.

The randomized algorithms were applied with $\alpha = 0.05$, $\varepsilon = 0.03$ and $\delta = 0.0001$ while the parameter set of feasible controllers $\Gamma = [0,1)$ was sampled in $p = 30$ points. By applying formula (2), the resulting number q of models extracted according to the estimated probability density was 7392.

The detuning parameter γ^o obtained at the first iteration turned out to be equal to 0.8. Its large value indicates a conservative choice which is justified by the high level of uncertainty. The step-response of the corresponding closed-loop system is depicted in Figure 6.

Carrying on the iterative procedure the identified nominal model became a more and more accurate description of the real plant, and, correspondingly, uncertainty tended to concentrate around the true system. This led to the selection of γ^o's as indicated in Figure 7. Figure 6 shows that the control performance rapidly improved through iterations, preserving always the robust stability.

VI. Conclusions

In this paper, we introduced a new robust iterative controller design scheme through which the controller performance can be rapidly improved through iterations, without experimental over-effort. Moreover, the robust stability is always preserved. Many implementation issues have been discussed, and in the simulation example the proposed methods provided good results.

References

[1] Anderson, B.D.O., X. Bombois, M. Gevers and C. Kulcsàr (1998). Caution in iterative modeling and control design. *IFAC Workshop on Adaptive Control and Signal Processing* (Glasgow, UK), 13–19.

[2] Bitmead, R.R., M.Gevers and A.G. Partanen (1997). Introducing caution in iterative controller design. *11th IFAC Symposium on System Identification* (Fukuoka, Japan), 1701–1706.

[3] Bittanti, S., M.C. Campi and S. Garatti (2002). An iterative controller design scheme based on average robust control. *15th World IFAC Congress* (Barcelona, Spain).

[4] Campi, M.C. and M. Prandini (2003). Randomized algorithms for the synthesis of cautious adaptive controllers. *Systems & Control Letters* **49**, 21–36.

[5] Campi, M.C. and G. Calafiore (2003). Robust Convex Programs: Randomized Solutions and Applications in Control. *42nd IEEE Conference on Decision and Control* (Hawaii, USA).

[6] De Callafon, R.A. and P.M.J. Van den Hof (1997). Suboptimal feedback control by a scheme of iterative identification and control design. *Mathematical Modelling of System* **3**, 77–101.

[7] Garatti, S., M.C. Campi and S. Bittanti (2003). On the assessment of the model quality through the asymptotic theory for an instrumental variable setting. *42nd IEEE Conference on Decision and Control* (Hawaii, USA), 6015-6020.

[8] Garatti, S., M.C. Campi and S. Bittanti (2004). Assessing the quality of identified models trough the asymptotic theory - when is the result reliable? *Automatica* **40**, 1319-1332.

[9] Gevers, M. (2000). A decade of progress in iterative control design: from theory to practice. *IFAC Symposium on Advanced Control of Chemical Processes* (Pisa, Italy), 677–694.

[10] Landau, I.D., D. Rey, A. Karimi, A. Voda and A. Franco (1995). A flexible transmission system as a benchmark for robust digital control. *European Journal of Control - special issue*.

[11] Lee, W.S., B.D.O. Anderson, I.M.Y. Mareels and R.L. Kosut (1995). On some key issues in the windsurfer approach to adaptive robust control. *Automatica* **31**, 1619–1636.

[12] Ljung, L. (1999). *System Identification: Theory for the User*. Prentice-Hall. Upper Saddle River, NJ.

[13] Van den Hof, P.M.J. and R.J.P. Schrama (1995). Identification and control - closed-loop issues. *Automatica* **31**, 1751–1770.

[14] Vidyasagar, M. (2001). Randomized algorithms for robust controller synthes is using statistical learning theory. *Automatica* **37**, 1515–1528.