Brief paper

Robust PID controller tuning based on the heuristic Kalman algorithm

Rosario Toscano *, Patrick Lyonnet

Université de Lyon, Laboratoire de Tribologie et de Dynamique des Systèmes CNRS UMR5513 ECL/ENISE, 58 rue Jean Parot 42023 Saint-Etienne cedex 2, France

ARTICLE INFO

Article history:
Received 16 September 2008
Received in revised form 28 January 2009
Accepted 15 May 2009
Available online 30 June 2009

Keywords:
PID controller
$\mathcal{H}_\infty$ control
Robustness
Non-convex optimization problems
Heuristic Kalman algorithm

ABSTRACT

This paper presents a simple but effective tuning strategy for robust PID controllers satisfying multiple $\mathcal{H}_\infty$ performance criteria. Finding such a controller is known to be computationally intractable via the conventional techniques. This is mainly due to the non-convexity of the resulting control problem which is of the fixed order/structure type. To solve this kind of control problem easily and directly, without using any complicated mathematical manipulations and without using too many "user defined" parameters, we utilize the heuristic Kalman algorithm (HKA) for the resolution of the underlying constrained non-convex optimization problem. The resulting tuning strategy is applicable both to stable and unstable systems, without any limitation concerning the order of the process to be controlled. Various numerical studies are conducted to demonstrate the validity of the proposed tuning procedure. Comparisons with previously published works are also given.

1. Introduction

It is a matter of fact that the PID (proportional-integral-derivative) controller is the most widely used in industrial applications. This is mainly due to its ability in solving a broad class of practical control problems as well as its structural simplicity, allowing the operators to use it without much difficulty. In addition, many PID tuning rules have been reported in the literature (see Aström and Hägglund (1995) for a good overview), which are simple and easy to use. However, most of these tuning methods have a limited domain of applications mainly due to restrictive assumptions concerning the process model.

Consequently, developing PID tuning techniques for "arbitrary" process models, satisfying some performance specifications remains an important issue. $\mathcal{H}_\infty$ control theory is a good approach to tackle this problem. Indeed, many robust stability and performance problems can be cast and solved into the $\mathcal{H}_\infty$ framework, without any limitation in the order of the plant. However, the order of the controller thus obtained is almost always greater than or equal to that of the process. This is of course unacceptable for a correct implementation with most of the commercially available PID controllers (Grassi & Tsakalis, 2000). In these conditions, the design step must take into account the structure of the controller.

Unfortunately, the problem of designing a robust controller with a given fixed structure (e.g. a PID) remains an open issue. This is mainly due to the fact that the set of all fixed order/structure stabilizing controllers is non-convex and disconnected in the space of controller parameters. This is a major source of computational intractability and conservatism (Rockafellar, 1993). Nevertheless, due to their practical importance, some new approaches for structured control have been proposed in the literature. Most of them are based on the resolution of Linear Matrix Inequalities LMIs (see for instance: Apkarian, Noll, and Duong Tuan (2003), Cao, Lam, and Sun (1998), Ebihara, Tokuyama, and Hagiwara (2004), Genc (2000), Grigoriadis and Skelton (1994), He and Wang (2006), Iwasaki and Skelton (1995), Mattei (2000), and Saeki (2006)). However, a major drawback with these kinds of approaches is the use of Lyapunov variables, whose number grows quadratically with the system size. For instance, if we consider a system of order 70, this requires, at least, the introduction of 2485 unknown variables whereas we are looking for the parameters of a fixed order/structure controller which contains a comparatively very small number of unknowns. It is then necessary to introduce new techniques capable of dealing with the non-convexity of certain problems arising in automatic control without introducing extra unknown variables.

In this spirit, Kim, Maruta, and Sugie (2008) (see also Maruta, Kim, and Sugie (2008)) have proposed to solve the non-convex optimization problem arising in the design of optimal PI/PID controllers, by the use of an augmented Lagrangian particle swarm optimization (ALPSO) (Sedlacek & Eberhard, 2006). Although the results obtained with this method are very convincing, it seems that the weakness of this approach lie mainly in the large number of parameters which have to be set by the user, namely: the...
number of particles, the initial velocity of the particles and their initial positions, the value of the inertia factor, the value of the cognitive factor, the value of the social factor and the maximum number of iterations. In addition to these latter parameters, some other parameters have to be set by the user to handle the constraints of the optimization problem. The difficulty is that there is no systematic procedure to select correctly the above mentioned parameters. The only way is thus to proceed by trial and error, but this is time consuming and can be very difficult to do for a large number of parameters.

Since the problem of selecting in advance many parameters is not obvious at all, it appears necessary to develop tuning strategies requiring the smallest possible number of “user defined” parameters. For this purpose, it seems interesting to use the heuristic Kalman algorithm (Toscano and Lyonnet, in press-b) because it possesses only three “user defined” parameters. The HKA (Heuristic Kalman Algorithm) enters into the category of the so called “evolutionary computation algorithms”. It shares with PSO interesting features such as: ease of implementation, low memory and CPU speed requirements, search procedure based only on the values of the objective function, no need of strong assumptions such as linearity, differentiability, convexity etc, to solve the optimization problem. In fact it could be used even when the objective function cannot be expressed in an analytic form, in this case, the objective function is evaluated through simulations.

The main objective of this paper is to develop a simple and easy to use tuning strategy for robust PID controllers satisfying multiple $H_{\infty}$ specifications. Finding such controller gain is known to be computationally intractable by the conventional techniques. Therefore, to solve this design problem easily and directly, without using too many “user defined” parameters, we utilize the heuristic Kalman algorithm for the resolution of the underlying constrained non-convex optimization problem. The resulting tuning method is applicable both to stable and to unstable systems, without any limitation concerning the order of the process to be controlled. However, it is difficult to guarantee its effectiveness in a theoretical way, because, as the PSO, HKA is essentially a stochastic method. Nevertheless, we have evaluated the effectiveness of the proposed method, empirically, through various numerical experiments.

The remaining part of this paper is organized as follows. In Section 2, the robust PID controller design based on the heuristic Kalman algorithm (HKA), is presented. Section 3 shows the validity of the proposed approach on various numerical applications, comparisons with previously published works are also given. Finally, Section 4 concludes this paper.

2. Robust PID controller design based on the heuristic Kalman algorithm (HKA)

In this section, a practical design procedure to determine the PID tuning parameters is presented. To this end, we first formulate the problem of designing a robust PID controller as an optimization problem.

2.1. Formulation of the optimization problem

Consider the general feedback setup shown in Fig. 1, in which $G(s)$ represents the transfer matrix of the generalized process to be controlled

\[
\begin{bmatrix}
 z \\
 y
\end{bmatrix} = G(s) \begin{bmatrix}
 w \\
 u
\end{bmatrix}, \quad \text{with: } G(s) = \begin{bmatrix}
 A & B_1 & B_2 \\
 C_1 & D_{11} & D_{12} \\
 C_2 & D_{21} & D_{22}
\end{bmatrix}
\]

and $K_{\text{PID}}(s)$ is the transfer matrix of the PID controller

\[
K_{\text{PID}}(s) = K_p + \frac{K_i}{s} + K_d \frac{s}{1 + \tau^2 s} = \begin{bmatrix}
 \frac{A_K}{C_K} & \frac{B_K}{D_K} \\
 0 & 0 & -\frac{1}{\tau^2} & -\frac{1}{\tau} K_d \\
 1 & 1 & \frac{K_p}{\tau^2} + \frac{K_d}{\tau}
\end{bmatrix}
\]

where $K_p \in \mathbb{R}^{n_u \times n_y}$ is the proportional gain, $K_i \in \mathbb{R}^{n_u \times n_y}$ and $K_d \in \mathbb{R}^{n_u \times n_y}$ are the integral and derivative gains respectively, and $\tau$ is the time constant of the filter applied to the derivative action.

This low-pass first-order filter ensures the properness of the PID controller and thus its physical realizability. In addition, since $G(s)$ is strictly proper (i.e. it is assumed that $D_{22} = 0$), the properness of the controller ensures the well-posedness of feedback loop.

As depicted Fig. 1, the closed-loop system has $m$ external input vectors $w_1 \in \mathbb{R}^{m_1}, \ldots, w_m \in \mathbb{R}^{m_m}$ and $m$ output vectors $z_1 \in \mathbb{R}^{o_1}, \ldots, z_m \in \mathbb{R}^{o_m}$. Roughly speaking, the global input vector $w = [w_1 \cdots w_m]^T$ captures the effects of the environment on the feedback system; for instance noise, disturbances and references. The global output vector $z = [z_1 \cdots z_m]^T$ contains all characteristics of the closed-loop system that are to be controlled.

To this end, the PID control law $K_{\text{PID}}(s)$, utilizes the measured output vector $y \in \mathbb{R}^y$, to elaborate the control action vector $u \in \mathbb{R}^u$ which modify the natural behavior of the process $G(s)$.

The objective is then to determine the PID parameters ($K_p$, $K_i$, $K_d$, $\tau$) allowing to satisfy some performance specifications such as: a good seed point tracking, a satisfactory load disturbance rejection, a good robustness to model uncertainties and so on. A powerful way to enforce these kinds of requirements is first to formulate the performance specifications as an optimization problem and then to solve it by an appropriate method. In the $H_{\infty}$ framework, the optimization problem can take one of the following forms:

\[
\begin{aligned}
\min_{\lambda} & \quad J_{\infty}(\lambda) = \|T_{wz}(s, x)\|_{\infty} \\
\text{s.t.} & \quad g_1(\lambda) = \text{arg max}\{\text{Re}(\lambda_i(\lambda)), \forall i\} - \lambda_{\min} \leq 0 \\
& \quad g_2(\lambda) = \|T_{wz_1}(s, x)\|_{\infty} - \gamma_2 \leq 0 \\
& \quad \vdots \\
& \quad g_m(\lambda) = \|T_{wz_m}(s, x)\|_{\infty} - \gamma_m \leq 0 \\
\end{aligned}
\]

or also:

\[
\begin{aligned}
\min_{\lambda} & \quad J_{\lambda}(\lambda) = \text{arg max}\{\text{Re}(\lambda_i(\lambda)), \forall i\} \\
\text{s.t.} & \quad g_1(\lambda) = \|T_{wz_1}(s, x)\|_{\infty} - \gamma_2 \leq 0 \\
& \quad g_2(\lambda) = \|T_{wz_2}(s, x)\|_{\infty} - \gamma_2 \leq 0 \\
& \quad \vdots \\
& \quad g_m(\lambda) = \|T_{wz_m}(s, x)\|_{\infty} - \gamma_m \leq 0 \\
\end{aligned}
\]

where $T_{wz}(s, x)$ denotes the closed-loop transfer matrix from $w_i$ to $z_i$, $x \in \mathbb{R}^x$ is the vector of decision variables regrouping the entries of the matrices $K_p$, $K_i$, $K_d$ and the time constant $\tau$: $x =
loss of generality, we assume that the vectors are ordered by their increasing cost function i.e.:
\[ J(x_1^k) < J(x_2^k) < \cdots < J(x_N^k). \] (8)

The principle of the algorithm is to modify the parameters of the Gaussian generator so that its mean vector \( m_k \) coincides with the optimum \( x_{\text{opt}} \). More precisely, let \( N_k \) be the number of considered best samples, that is such that \( J(x_{\text{opt}}^k) < J(x_i^k) \) for all \( i > N_k \).

The problem is how to modify the parameters of the Gaussian generator to achieve a reliable estimate of the optimum.

To solve this problem, a measurement process followed by a Kalman estimator is introduced. The measurement process consists in computing the average of the candidates that are the most representative of the optimum. For the iteration \( k \), the measurement, denoted \( \xi_k \), is then defined as follows:
\[ \xi_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i^k \] (9)

where \( N_k \) is the number of considered candidates. The Kalman estimator is used to update the parameters of the Gaussian generator in accordance with the information drawn from the samples, i.e. the value of \( \xi_k \) and the variance vector associated to the best samples:
\[ V_k = \frac{1}{N_k} \left( \sum_{i=1}^{N_k} (x_{1,k}^i - \xi_{1,k})^2, \ldots, \sum_{i=1}^{N_k} (x_{n,k}^i - \xi_{n,k})^2 \right)^T. \] (10)

Based on the Kalman equations, the updating rules of the Gaussian generator are as follows (see (Toscano and Lyonnet, in press-a; in press-b) for a detailed derivation):
\[ m_{k+1} = m_k + L_k (\xi_k - m_k) \] (11)
\[ S_k+1 = S_k + a_k (W_k - S_k) \] with:
\[ L_k = \Sigma_k (\Sigma_k + \text{diag}(V_k))^{-1} \] (12)
\[ W_k = [\text{vec}^d((I - L_k)\Sigma_k)]^{1/2} \] (13)
\[ \alpha_k = \min \left( \frac{1}{\mu}, \frac{1}{\sqrt{n} \sum_{i=1}^{n} \sqrt{v_{i,k}}} \right)^2 + \max_{1 \leq i \leq n} (w_{i,k}) \] where \( S_k \) is the standard deviation vector of the Gaussian generator \( S_k = [\text{vec}^d(\Sigma_k)]^{1/2} \), \( \text{vec}^d(\cdot) \) is the diagonal vector of the matrix \( \cdot \), \( v_{i,k} \) represents the \( i \)-th component of the variance vector \( V_k \) defined in (10), \( w_{i,k} \) is the \( i \)-th component of the vector \( W_k \), and the scalar \( \alpha \in (0, 1) \) is given by the designer. The flowchart of the HKA is given in Fig. 3.

2.3. Initialization and parameter settings

The initial parameters of the Gaussian generator are selected to cover the entire search space. To this end, the following rule is used:
\[ m_0 = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{n_k} \end{bmatrix}, \quad \Sigma_0 = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{n_k}^2 \end{bmatrix} \] (13)
\[ \mu_i = \frac{\bar{x}_i + X_i}{2}, \quad \sigma_i = \frac{\bar{x}_i - X_i}{6}, \quad i = 1, \ldots, n_k \]

where \( \bar{x}_i \) (respectively \( X_i \)) is the \( i \)-th upper bound (respectively lower bound) of the hyperbox search domain.

Fig. 2. Principle of the algorithm.
Table 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( N )</th>
<th>( N_t )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of function evaluations</td>
<td>( \triangleleft )</td>
<td>( \triangleleft )</td>
<td>( \triangleleft )</td>
</tr>
<tr>
<td>Average error</td>
<td>( \triangleright )</td>
<td>( \triangleright )</td>
<td>( \triangleright )</td>
</tr>
</tbody>
</table>

With this rule, 99% of the samples are generated in the intervals \( \mu_i \pm 3\sigma_i \), \( i = 1, \ldots, n_x \). We have thus to set the three following parameters: the number of points \( N \), the number of best candidates \( N_t \) and the coefficient \( \alpha \). To facilitate this task, Table 1 summarizes the influence of these parameters on the number of function evaluations (and so on the CPU time) and on the average error.

### 2.4. The feasibility issue

Note that the initialization rule (13) requires the knowledge of the bounds of a feasible search domain. In many engineering problems these bounds are often known a priori because they are linked to purely material, physical considerations. This is not so clear in control problem for which we have to impose a priori an hyperbox search domain containing stabilizing controllers (i.e. potential solutions of the optimal control problem). Finding a priori such a hyperbox is not trivial at all. However, for a given hyperbox search domain it is possible, using HKA, to say whether or not the problem is feasible. More precisely, the feasibility problem can be stated as follows. Given the hyperbox search domain \( D = \{ x_i \in \mathbb{R} : x_{i_d} \leq x_i \leq x_{i_u}, i = 1, \ldots, n_x \} \) is there a stabilizing controller? This important issue can be treated via HKA by solving the following optimization problem:

Minimize \( J_{1}(x) = \arg \max\{ \text{Re}(\lambda_{i}(x)), \quad \forall i \} \) subject to: \( x_{i_d} \leq x_i \leq x_{i_u}, \quad i = 1, \ldots, n_x \)

where \( \lambda_{i}(x) \) represents the \( i \)th pole of the closed-loop system.

Let \( x^* \) the solution found by HKA to the problem (14). If \( J_{1}(x^*) < 0 \), then the problem is feasible within \( D \). In this case, \( m_0 \) (see relation (13)) can be initialized with \( x^* \), leading thus to a search domain centered around a feasible controller parameters. In the case where \( J_{1}(x^*) > 0 \), because of the stochastic nature of HKA, this does not necessarily mean that the problem is not feasible (in this case we will say that the problem is probably not feasible).

Note that since the hyperbox search domain does not contain, a priori, all feasible solutions, the proposed method can only provide a local optimal solution. However, this is not a major disadvantage because a local optimal solution is often sufficient in practice. Further, it must be noticed that the theoretical guarantee to obtain a global optimal solution can only be given for convex problems. This is not the case of the considered optimization problems which are non-convex and non-smooth in nature and then not solvable via usual techniques. Therefore the proposed approach seems to be an interesting way for solving this kind of difficult problems in a straightforward manner.

### 3. Numerical experiments

In this section, the ability of HKA in solving the problem of designing a robust SISO or MIMO PID controller is tested on various numerical examples including stable or unstable plants, non-minimum phase, SISO or MIMO, low or high order systems. In all cases our results are compared to those obtained via other synthesis methods. The various experiments were performed using a 1.2 GHz Celeron personal computer.

#### 3.1. Mixed sensitivity approach

In this section, for comparison purpose, the same optimization problem as the one presented in Kim et al. (2008), is considered:

\[
\begin{align*}
\min & \quad J_{1}(x) = \arg \max\{ \text{Re}(\lambda_{i}(x)), \quad \forall i \} \\
\text{s.t.} & \quad g_1(x) = \sup_{\omega > 0} \{ |W_z(s)\|_{\text{H}_{\infty}}|S(s,x)\|_{\text{H}_{\infty}}| - 1 \leq 0 \\
& \quad g_2(x) = \sup_{\omega > 0} \{ |W_r(s)\|_{\text{H}_{\infty}}|T(s,x)\|_{\text{H}_{\infty}}| - 1 \leq 0 \\
& \quad g_3(x) = J_{1}(x) - \lambda_{\text{dmin}} \leq 0
\end{align*}
\]

where \( x = [x_1 \ x_2 \ x_3 \ x_4]^T \) is the vector of decision variables, \( s \) is the Laplace variable, \( \omega \) is the frequency (rad/s), \( j \) is the unit imaginary number, \( S(s,x) \) is the sensitivity function defined as \( S(s,x) = 1/(1 + L(s,x)) \), \( T(s,x) \) is the closed-loop system defined as \( T(s,x) = L(s,x)/(1 + L(s,x)) \), \( L(s,x) \) is the open-loop transfer function defined as \( L(s,x) = G(s)K(s,x) \) where \( G(s) \) is the transfer function of the system to be controlled and \( K(s,x) \) is the transfer function of the PID controller which depends upon the decision variables as follows:

\[
K(s,x) = 10^{a_{11}} \left( 1 + \frac{1}{10^{a_{12}}} + \frac{10^{a_{13}}}{1 + 10^{a_{14}}s} \right).
\]

Note that the relationship between the decision variables of the optimization problem and the parameters of the PID controller are defined as:

\[
K_p = 10^{a_{11}}, \quad T_i = 10^{a_{12}}, \quad T_d = 10^{a_{13}}, \quad N = 10^{a_{14}}
\]

This is done to ensure a broader parameter space of \((K_p, T_i, T_d, N)\). The frequency-dependent weighting functions \( W_z(s) \) and \( W_r(s) \) are set in order to meet the performance specifications of the closed-loop system. Note that the optimization problem (15) is of the form (4).
Table 2
Comparison of the best solutions found via ALPSO and HKA, (a) without violation constraint, (b) case of a small violation constraint.

<table>
<thead>
<tr>
<th></th>
<th>ALPSO</th>
<th>HKA (a)</th>
<th>HKA (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>3.2548</td>
<td>3.2542</td>
<td>3.2556</td>
</tr>
<tr>
<td>$x_2$</td>
<td>-0.8424</td>
<td>-0.8634</td>
<td>-0.8354</td>
</tr>
<tr>
<td>$x_3$</td>
<td>-0.7951</td>
<td>-0.7493</td>
<td>-0.7539</td>
</tr>
<tr>
<td>$x_4$</td>
<td>2.3337</td>
<td>2.3139</td>
<td>2.3127</td>
</tr>
<tr>
<td>$g_1(x)$</td>
<td>$6.1 \times 10^{-3}$</td>
<td>$-9.6 \times 10^{-4}$</td>
<td>$4.9 \times 10^{-3}$</td>
</tr>
<tr>
<td>$g_2(x)$</td>
<td>$-4.0 \times 10^{-4}$</td>
<td>$-1.2 \times 10^{-3}$</td>
<td>$-2.8 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f(x)$</td>
<td>-1.7197</td>
<td>-1.7106</td>
<td>-1.7435</td>
</tr>
</tbody>
</table>

Table 3
Statistical results, (a) without violation constraint, (b) case of a small violation constraint.

<table>
<thead>
<tr>
<th></th>
<th>ALPSO</th>
<th>HKA (a)</th>
<th>HKA (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>-1.7197</td>
<td>-1.7106</td>
<td>-1.7435</td>
</tr>
<tr>
<td>Mean</td>
<td>-1.7023</td>
<td>-1.7381</td>
<td>-1.7323</td>
</tr>
<tr>
<td>Worst</td>
<td>-1.6891</td>
<td>-1.7332</td>
<td>-1.7323</td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.0048</td>
<td>0.0030</td>
<td>0.0030</td>
</tr>
<tr>
<td>CPU time</td>
<td>687 s</td>
<td>266 s</td>
<td>248 s</td>
</tr>
<tr>
<td>NbFuncEval</td>
<td>25 000</td>
<td>5427</td>
<td>5072</td>
</tr>
</tbody>
</table>

3.1.1. Application to a magnetic levitation system

The optimization problem (15), has been solved for the magnetic levitation system described in Sugie, Simizu, and Imura (1993). The process model is defined as:

$$ P(s) = \frac{7.147}{(s - 22.55)(s + 20.9)(s + 13.99)}. $$

The frequency-dependent weighting functions $W_5(s)$ and $W_1(s)$ are respectively given as:

$$ W_5(s) = \frac{5}{s + 0.1} $$
$$ W_1(s) = \frac{43.867(s + 0.066)(s + 31.4)(s + 88)}{(s + 10^4)^2}. $$

The search space is:

$$ 2 \leq x_1 \leq 4, \quad -1 \leq x_2 \leq 1, \quad -1 \leq x_3 \leq 1, \quad 1 \leq x_4 \leq 3. $$

In this test, we performed the minimization 30 times and we compared our results with those obtained via ALPSO. The following parameters have been used: $N = 50, N_t = 5$ and $\alpha = 0.4$.

The best solutions obtained via ALPSO and HKA are listed in Table 2(a) and the statistical results are shown in Table 3(a) (the mark “−” means that the corresponding result is not available).

The better value of the objective function obtained with ALPSO is due to the violation of the constraint $g_1(x)$, this is not at all the case in our solution for which all constraints are satisfied. From Table 3(a) we can observe that the number of function evaluations and the related CPU time are very small compared to ALPSO. It is interesting to note that if, as in ALPSO, a small violation of the constraint $g_1(x)$ is tolerated, we obtain the results listed in Tables 2(b) and 3(b).

From Table 2(b), it can be seen that the best solution found by HKA is significantly better than the solution found by ALPSO with, in addition, a smaller violation constraint. Table 3(b), shows that the worst solution found by HKA is better than the solution found via ALPSO. In addition, the number of function evaluations and so the corresponding CPU time) remains very small compared to ALPSO.

3.2. PID loop-shaping design

$\mathcal{H}_\infty$ loop-shaping design procedure proposed by McFarlane and Glover (1992) is an efficient method to design robust controllers and has been successfully applied to a variety of practical problems. In this framework, the plant $G(s)$ is first shaped with a pre-compensator $W_1(s)$ and a post-compensator $W_2(s)$.

The ponderations $W_1$ and $W_2$ are chosen so that the weighted plant $W_1G(s)W_2$ has a desired loop shape, typically a large gain at low frequencies for performance and a small gain at high frequencies for noise attenuation. Once the desired loop shape is achieved, $\mathcal{H}_\infty$ norm of the transfer function matrix from disturbances $w_1$ and $w_2$ to the outputs $z_1$ and $z_2$ (see Fig. 4) is minimized over all stabilizing controllers$^1$:

$$ K^* = \arg \min_{K} \| T_w(K) \|_\infty $$

$$ = \arg \min_{K} \left\{ \| KH \| \right\} $$

with: $H = (I - W_2GW_1)^{-1}$

Subject to: $\max|\text{Re}(\lambda_i(K))|, \forall i < 0$.

The final controller is then implemented as $W_1K^*W_2$ and has no specific structure. The quantity $\epsilon = 1/\| T_w(K^*) \|_\infty$ is known as the robust stability margin; usually value of $\epsilon > 0.2$ or 0.3 is considered as very satisfactory in the sense that the controller $K^*$ does not significantly alter the desired open-loop frequency response. Moreover, this ensures robustness of the closed-loop system to coprime factor uncertainties (McFarlane & Glover, 1992).

For loop-shaping design with PID, we adopt the strategy introduced in Apkarian, Bompart, and Noll (2007) and Genc (2000). In this approach, the controller $K$ is structured as $K = K_{\text{PID}}$, where $K_{\text{PID}}$ is the PID controller (2). Thus the optimization problem (20) becomes:

$$ K_{\text{PID}}^* = \arg \min_{K_{\text{PID}}} \| T_w(K_{\text{PID}}) \|_\infty $$

Subject to: $\max|\text{Re}(\lambda_i(K_{\text{PID}}))|, \forall i < 0$.

where $T_w(K_{\text{PID}})$ is given by:

$$ T_w(K_{\text{PID}}) = \begin{bmatrix} \frac{W_1}{K_{\text{PID}}}HGW_1 & W_1^{-1}K_{\text{PID}}H \end{bmatrix} $$

$$ H = (I - W_2K_{\text{PID}})^{-1}. $$

The final controller is then implemented as $K_{\text{PID}}^*W_2$. Since $W_2$ is usually chosen as a low-pass filter, the resulting controller has better noise attenuation in the high frequency range than an usual PID controller.

3.2.1. Application to a separating tower

We consider now the application of the HKA to the control design for a chemical process described in Apkarian et al. (2007) and Genc (2000). It consists of a 24-tray tower for separating methanol and water. The transfer matrix model for controlling the temperature on the 4th and 17th trays is given as:

$$ \begin{bmatrix} t_1 \\ t_4 \end{bmatrix} = \begin{bmatrix} -2.2e^{-5} & 1.3e^{-0.3s} \\ 7s + 1 & 7s + 1 \\ -2.8e^{-1s} & 4.3e^{-0.35s} \\ 9.5s + 1 & 9.2s + 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. $$
The transfer matrix (23) is approximated by a rational model using a 2nd-order Padé approximation of the delays. This leads to a 12th-order model. The weighting matrix $W_1$ and $W_2$ are taken from Genc (2000) and Apkarian et al. (2007):

$$W_1(s) = \begin{bmatrix} 5s + 2 & 0 \\ s + 0.001 & 5s + 2 \\ 0 & s + 0.001 \end{bmatrix},$$

$$W_2(s) = \begin{bmatrix} 10 \\ s + 10 \\ 0 \\ s + 10 \end{bmatrix}.$$ (24)

The complete system incorporating the compensators is therefore of 18th-order. Our objective is to find the PID parameters $x = [x_1, \ldots, x_{13}] (-3 \leq x_i \leq 3, i = 1, \ldots, 13)$ defined as follows:

$$K_{PID}(s, x) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 1 \\ x_{13} \end{bmatrix} s,$$

$$\tau = 0.06.$$ (26)

The corresponding robustness margin is $\epsilon = 1/4.02 = 0.249$. In Apkarian et al. (2007) the same problem was solved using a nonsmooth optimization technique. The algorithm was initialized with the solution (26) and the following PID was found in about 1 min:

$$K_p = \begin{bmatrix} 2.6047 \\ -1.1253 \end{bmatrix}, K_i = \begin{bmatrix} 0.8527 \\ 0.0701 \end{bmatrix}, K_d = \begin{bmatrix} -0.2551 \\ -1.5610 \end{bmatrix}.$$ (27)

The corresponding robustness margin is $\epsilon = 1/2.91 = 0.343$. This is an impressive improvement in terms of CPU time and robustness margin compared to the result reported by Genc (2000).

In our case, we solved the optimization problem 15 times ($N = 50, N_w = 2$ and $\alpha = 0.5$), the corresponding statistical results are shown Table 4. The best solution found via HKA is as follows:

$$K_p = \begin{bmatrix} 2.6091 \\ -1.1068 \end{bmatrix}, K_i = \begin{bmatrix} 0.8025 \\ -0.0390 \end{bmatrix}, K_d = \begin{bmatrix} -0.2852 \\ -1.5979 \end{bmatrix}.$$ (28)

The corresponding robustness margin is $\epsilon = 1/2.93 = 0.341$. Step responses are shown in Fig. 5. The best solution (28) was found in about 66 s on 1.2 GHz Celeron personal computer. Note that this PID controller is very close to the result obtained by Apkarian et al. (2007). However, it must be noticed that the proposed approach is very easy to use and does not require any complicated mathematical derivation. Compared to $D$–$K$ iteration or nonsmooth optimization, HKA seems to be a good alternative in terms of simplicity, near optimality of the solutions and computation time.
between its channels (i.e., Box I does not exhibit open-loop column diagonal dominance).

To apply our approach, the transfer matrix Box I is approximated by a rational model using 2nd-order Padé approximation of the delays. This lead to a 60th-order model. The weighting matrix \( W_1 \) and \( W_2 \) are as follows:

\[
W_1(s) = \frac{268(1 + 90s)}{1 + 5285s} - I_4, \quad W_2(s) = \frac{5}{s} + I_4.
\] (29)

The complete system incorporating the compensators is therefore of 72nd-order. Our objective is to find the PID parameters \( \bar{x} = [x_1, \ldots, x_4] \) within the hyperbox search domain \( \mathcal{D} = [x_i \in \mathbb{R} : -10 \leq x_i \leq 10, i = 1, \ldots, 49] \) satisfying (21). First, we use the approach described in Section 2.4 to evaluate the feasibility problem. Solving problem (14) via HKA, we get an initial stabilizing controller parameters \( x_{\text{stab}} \) (see Appendix) which shows that \( \mathcal{D} \) is feasible (i.e. it contain potential solutions). Thus, problem (21) can then be solved within \( \mathcal{D} \). Starting HKA from \( x_{\text{stab}} \) we solved the optimization problem (21) 15 times \( (N = 50, N_t = 5 \text{ and } \alpha = 0.4) \), the corresponding statistical results are shown Table 5.

The best solution found via HKA is as follows:

\[
K_p = \begin{bmatrix}
  -0.7378
  2.1080
  1.4210
  1.4178
\end{bmatrix}, \quad K_i = \begin{bmatrix}
  -0.0142
  0.0068
  0.0170
  0.0066
\end{bmatrix}, \quad K_d = \begin{bmatrix}
  0.0827
  1.5728
  0.4279
  0.3528
\end{bmatrix}
\] (30)

\[\tau = 0.6646.\]

The corresponding robustness margin is \( \epsilon = 1/2.983 = 0.33 \). Step responses are shown in Fig. 6. The best solution (30) was found in about 2580 s. For comparison, Fig. 7 gives the step responses obtained with a decentralized control using a static decoupler (see Chen and Seborg (2003)). From Figs. 6 and 7 we can appreciate the quality of the responses obtained via \( H_\infty \) loop-shaping technique: better transient responses as well as lowest interactions between channels.
Table 5
Statistical results (4 × 4 chemical process).

<table>
<thead>
<tr>
<th>$\mathcal{H}<em>\infty$ ($\mathcal{K}</em>{PI}$)</th>
<th>HKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best – Mean – Worst</td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.105</td>
</tr>
<tr>
<td>CPU time (Average)</td>
<td>2760 s</td>
</tr>
<tr>
<td>Iterations (Average)</td>
<td>178</td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper, a straightforward design method for robust PID controllers satisfying multiple $\mathcal{H}_\infty$ performance criteria was developed. This sort of control problem usually results in a non-convex constrained optimization problem which is known to be very difficult to deal with. This is why, to solve in a direct way this kind of control problem, we have proposed to use the Heuristic Kalman Algorithm. Indeed, HKA runs without any conservative assumption usually required in the conventional methods, in addition it allows to determine PID gains by solving the constrained optimization problem in a direct way without requiring too many design parameters unlike other stochastic algorithms such as ALPSO.

Various simulation studies have demonstrated the validity of the proposed approach, and comparisons with previously published works have shown that HKA leads to better results notably concerning the minimality of the solutions found and the CPU time.

Appendix. Initial stabilizing controller $x_{stab}$

$$x_{stab} = \begin{bmatrix} -0.1888, -1.6427, 0.2887, -0.0951, 2.3610, \\ -2.7410, -1.5741, -2.6494, 2.1092, -0.7752, \\ -0.8620, -2.6128, 1.8126, -1.4310, 2.2920, \\ -2.7665, -0.0638, 0.0078, -0.0798, 0.0500, 0.0588, \\ 0.0171, -0.0315, 0.0587, -0.0186, -0.0026, \\ -0.0342, -0.0959, 0.0015, -0.0321, 0.0555, 0.0002, \\ 0.3900, -0.3476, 0.3904, -0.0142, -0.0929, -0.3375, \\ 0.1638, -0.4016, -0.2162, -0.4938, 0.0735, 0.0113, \\ -0.1029, -0.4360, 0.3509, 0.2246, 0.7052^T \end{bmatrix} \quad (A.1)$$

References


Rosario Toscano was born in Catania, Italy. He received his masters degree with specialization in control from the Institut National des Sciences Appliquées de Lyon in 1996. He received the Ph.D. degree from the Ecole Centrale de Lyon in 2000. He received the HDR degree (Habilitation to Direct Research) from the University Jean Monnet of Saint-Etienne in 2007. He is currently an associate professor at the Ecole Nationale d’Ingénieurs de Saint-Etienne (ENISE). His research interests include dynamic reliability, fault detection, robust control, and multimodel approach applied to diagnosis and control.

Patrick Lyonnet was born in Roman, France. He received the M.S. degree from the Université de Technologie de Compiègne, France, in 1979 and the Ph.D. degree from the same université in 1991. He is currently professor at the Ecole Nationale d’Ingénieurs de Saint-Etienne (ENISE). His research interests include dynamic reliability, fault detection, fault detection, robust control, and multimodel approach applied to diagnosis and control.