

Article

A Randomized Algorithm for Optimal PID Controllers

Yossi Peretz

Department of Computer Sciences, Lev Academic Center, Jerusalem College of Technology,
P.O.B. 16031 Jerusalem, Israel; yosip@g.jct.ac.il; Tel.: +972-2-6751016; Fax: +972-2-6751046

Received: 7 May 2018; Accepted: 2 June 2018; Published: 5 June 2018



Abstract: A randomized algorithm is suggested for the syntheses of optimal PID controllers for MIMO coupled systems, where the optimality is with respect to the H_∞ -norm, the H_2 -norm and the LQR functional, with possible system-performance specifications defined by regional pole-placement. Other notions of optimality (e.g., mixed H_2/H_∞ design, controller norm or controller sparsity) can be handled similarly with the suggested algorithm. The suggested method is direct and thus can be applied to continuous-time systems as well as to discrete-time systems with the obvious minor changes. The presented algorithm is a randomized algorithm, which has a proof of convergence (in probability) to a global optimum.

Keywords: continuous-time linear systems; discrete-time linear systems; state-space representation; proportional-integral-differential controllers; static output feedbacks; optimal controllers; randomized algorithms; convergence in probability

1. Introduction

Proportional Integral Derivative (PID) controllers are the most widely-used controllers in industry despite of many new results in control theory that have been achieved in recent years. This is because in many cases PID controllers achieve close to optimal performance, because it is hard to compete with their investment-to-profit ratio and because of their well-understood influence on the closed-loop system behavior, as well as the availability of many tuning procedures for various performance objectives, at least for SISO systems or MIMO systems that can be well decoupled. Often, the performance achieved by PID controllers can be significantly improved by using more effective design techniques. Therefore the field of PID controllers is an on-going active field of research. A review of most recent techniques for tuning and designing PID-based control structures, together with methods for assessing their performance is presented in [1]. In [2] optimal decentralized MIMO PID controller design for a benchmark MIMO distillation column models is carried out using a Modified Firefly Algorithm (MFA), where the optimality is with respect to an objective function which is a weighted combination of the L_1 -norm and the L_2 -norm of the tracking error, with respect to a reference signal.

Tuning of MIMO coupled systems PID controllers is much more challenging than tuning PID controllers for SISO system or decoupled MIMO systems, because of the number of parameters and because they should be tuned all at once, since otherwise, tuning each channel separately might end up in a very poor performance of the whole system or even destabilize the whole system. In [3] the need for effective methods for the design of PID optimal controllers for MIMO coupled systems that cannot be well approximated by second-order systems, was raised. In [4] the parametrization of all the solutions of the standard H_∞ control problem is used to obtain Bilinear Matrix Inequalities (BMI's) constraints on the PID gains for each frequency. Next, for each frequency, Linear Matrix Inequalities (LMI's) restricted constraints for a stabilizing PID gain are derived from the BMI constraints. Finally, the set of LMI's for a chosen finite set of frequencies are solved by the Iterated Linear Matrix Inequalities

(ILMI) method. The sequence is shown to converge to a local optimum. In [5] a method for MIMO coupled systems PID controller design for stable plants (given by transfer function at an appropriate set of frequencies) is represented. The method relies on solving a sequence of SDP's, but it cannot guarantee finding a globally optimal design.

In this article, we suggest a randomized algorithm for PID optimal controllers. The algorithm is based on the known reduction of [3] to the Stabilizing Static-Output-Feedback (SOF) problem, which is then solved by the Ray-Shooting Method, where the method is used both for the search for feasible starting SOF and for the optimal SOF (see [6–8] for some applications of the method).

A closely related approach to ours is given in [9] where the problem is first reduced to the SOF problem, by the reduction of [3] and by using a descriptor system representation of the augmented system (in order to bypass the feasibility condition that $I - \hat{K}_3 C_2 B_2$ should be invertible—see the following). Next, the search problem of a starting stabilizing SOF controller is solved via the ILMI approach. Finally, a Bounded-Real Lemma version for descriptor systems is used in order to get a set of Quadratic Matrix Inequalities (QMI's) that their solution guaranties a γ suboptimal H_∞ gain. The last can be solved by the ILMI method (without a proof of convergence). The method of ILMI was discussed in many articles (see [3,10–12]). Unfortunately, the ILMI method needs a good starting point and has no proof of convergence in general.

Another closely related approach is given in [13] where the 2-Degree of Freedom PID (2-DOF-PID) problem is reduced to the SOF problem, which in order to incorporate the optimization problem of γ suboptimal H_∞ gain, is reformulated as a set of BMI's, where the last is solved via standard BMI solvers.

Many problems can be reduced to the SOF problem, imposing structural or other constraints on the controllers. These include the reduction of the minimal-degree dynamic-feedback problem and robust or decentralized stability via static-feedback (see [14,15], resp.). The design problem of remote digital PID controllers formulated as robust SOF problem for discrete-time Networked Control Systems (NCS), subject to time-delays and randomly missing-data is considered in [16]. A formulation of the reduced-order H_∞ filter problem as constrained SOF problem, is considered in [17]. Unfortunately, the SOF problem with bounded entries or other structural constraints is known to be NP-hard (see [14,18]). Therefore, minimal-gain SOF with bounded entries is obviously NP-hard problem. Exact Pole-placement and simultaneous and robust stabilization via SOF are also NP-hard (see [14,19], resp.). Thus, practically, one should expect that only approximation or randomized algorithms will be able to cope with these problems.

The article is organized as follows:

In Section 2 we define the feasibility problem of PID controllers and the specific reduction to the SOF problem that will be used. In Section 3 we define the PID controller optimization problems where the optimality is with respect to the H_∞ -norm, the H_2 -norm and the LQR functional, with possible regional pole-placement. In Section 4 we describe the suggested randomized algorithm and finally, in Section 5 we give a real-life example for each one of the above-mentioned optimization problems.

The notions are standard. By $\Re(z)$ and $\Im(z)$ we denote the real and imaginary parts of $z \in \mathbb{C}$, resp., where by \mathbb{C}_- we denote the left half-plane. For a matrix Z we denote by Z^T its transpose and by \bar{Z} we denote the conjugate matrix. By Z^* we denote conjugate-transpose of Z . By Z^+ we denote the Moore-Penrose pseudo-inverse of Z , where by L_Z and R_Z we denote the left and right orthogonal projections $I - Z^+Z$ and $I - ZZ^+$, resp.. For a square matrix Z , we denote by $\sigma(Z)$ the spectrum of Z and by $\text{trace}(Z)$ we denote the trace of Z . For matrices Z, W we denote by $Z \otimes W$ the Kronecker product. By $\|Z\|$ we denote the spectral norm of Z , i.e., the largest singular value of Z , where by $\|Z\|_F$ we denote its Frobenious norm, i.e., $(\text{trace}(Z^*Z))^{\frac{1}{2}}$. For matrices Z, W with the same size, we denote by $\langle Z, W \rangle_F = \text{trace}(W^*Z)$ the Frobenious inner product. For a vector v we denote by $\|v\|$ its Euclidean norm $(\sum_j |v_j|^2)^{\frac{1}{2}}$. By $\text{vec}(Z)$ we denote the vector obtained from the matrix Z by chaining all its columns to a single column. By mat we denote the inverse function of vec . For a transfer function $T(s)$ analytic in the open right half-plane, we denote by $\|T\|_{H_\infty}$ its H_∞ norm where $\|T\|_{H_\infty} = \text{ess - sup}_{\omega \in \mathbb{R}} \|T(j\omega)\|$ and by $\|T\|_{H_2}$ we denote its H_2 norm where

$\|T\|_{H_2} = \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{trace} (T(j\omega)^* T(j\omega)) d\omega \right)^{\frac{1}{2}}$. For a set X in a topological space, we denote by \bar{X} its closure and by $\text{int}(X)$ its interior. For two sets X, Y we denote by $CH(X, Y)$ the convex-hull of X and Y , i.e., the minimal closed convex set that contains $X \cup Y$.

2. The Problem of Stabilization via PID Controllers

Let a system be given by:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \\ z(t) = C_1x(t) + D_{1,1}w(t) + D_{1,2}u(t) \\ y(t) = C_2x(t) + D_{2,1}w(t), \end{cases}$$

where $A \in \mathbb{R}^{n_x \times n_x}, B_1 \in \mathbb{R}^{n_x \times n_w}, B_2 \in \mathbb{R}^{n_x \times n_u}, C_1 \in \mathbb{R}^{n_z \times n_x}, D_{1,1} \in \mathbb{R}^{n_z \times n_w}, D_{1,2} \in \mathbb{R}^{n_z \times n_u}, C_2 \in \mathbb{R}^{n_y \times n_x}, D_{2,1} \in \mathbb{R}^{n_y \times n_w}$, where x is the state, z is the regulated output, y is the measurement, u is the control input and w is the noise. Assume that $D_{2,1} = 0$ and let

$$u(t) = r(t) - K_P y(t) - K_I \int_0^t y(\tau) d\tau - K_D \dot{y}(t),$$

where $r(t)$ is the new reference input. Let $\hat{x}(t) = \begin{bmatrix} x(t) \\ \int_0^t y(\tau) d\tau \end{bmatrix}$ be the augmented state, then:

$$\begin{aligned} u(t) &= r(t) - K_P y(t) - K_I \int_0^t y(\tau) d\tau - K_D \dot{y}(t) \\ &= r - K_P \begin{bmatrix} C_2 & 0 \end{bmatrix} \hat{x}(t) - K_I \begin{bmatrix} 0 & I \end{bmatrix} \hat{x}(t) + \\ &\quad - K_D C_2 \dot{\hat{x}}(t) \\ &= r - K_P \begin{bmatrix} C_2 & 0 \end{bmatrix} \hat{x}(t) - K_I \begin{bmatrix} 0 & I \end{bmatrix} \hat{x}(t) + \\ &\quad - K_D C_2 (Ax(t) + B_1w(t) + B_2u(t)), \end{aligned}$$

implying that:

$$\begin{aligned} u(t) &= M^{-1}r(t) - M^{-1}K_P \begin{bmatrix} C_2 & 0 \end{bmatrix} \hat{x}(t) + \\ &\quad - M^{-1}K_I \begin{bmatrix} 0 & I \end{bmatrix} \hat{x}(t) + \\ &\quad - M^{-1}K_D \begin{bmatrix} C_2 A & 0 \end{bmatrix} \hat{x}(t) - M^{-1}K_D C_2 B_1 w, \end{aligned}$$

where $M = I + K_D C_2 B_2$ is assumed to be invertible. Now:

$$\begin{aligned} \hat{x}(t) &= \begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} \\ &= \begin{bmatrix} Ax(t) + B_1w(t) + B_2u(t) \\ C_2x(t) \end{bmatrix} \\ &= \begin{bmatrix} A & 0 \\ C_2 & 0 \end{bmatrix} \hat{x}(t) + \\ &\quad + \begin{bmatrix} B_1 - B_2 M^{-1} K_D C_2 B_1 \\ 0 \end{bmatrix} \hat{w}(t) + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \hat{u}(t), \end{aligned}$$

where:

$$\begin{cases} \hat{w}(t) = w(t) \\ \hat{u}(t) = \hat{r}(t) - \hat{K}\hat{y}(t) \\ \hat{r}(t) = M^{-1}r(t) \\ \hat{y}(t) = \begin{bmatrix} C_2x(t) \\ \int_0^t y(\tau) d\tau \\ C_2Ax(t) \end{bmatrix} \\ \hat{K} = M^{-1} \begin{bmatrix} K_P & K_I & K_D \end{bmatrix}. \end{cases}$$

Let

$$\begin{cases} \hat{A} = \begin{bmatrix} A & 0 \\ C_2 & 0 \end{bmatrix}, \hat{B}_2 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \\ \hat{B}_1 = \begin{bmatrix} B_1 - B_2M^{-1}K_D C_2B_1 \\ 0 \end{bmatrix} \\ \hat{C}_1 = \begin{bmatrix} C_1 & 0 \end{bmatrix}, \hat{C}_2 = \begin{bmatrix} C_2 & 0 \\ 0 & I \\ C_2A & 0 \end{bmatrix} \\ \hat{D}_{1,1} = D_{1,1} - D_{1,2}M^{-1}K_D C_2B_1, \hat{D}_{1,2} = D_{1,2}. \end{cases}$$

Let $\hat{z}(t) = z(t)$ then, the augmented system is given by:

$$\begin{cases} \dot{\hat{x}}(t) = (\hat{A} - \hat{B}_2\hat{K}\hat{C}_2) \hat{x}(t) + \hat{B}_1\hat{w}(t) + \hat{B}_2\hat{r}(t) \\ \hat{z}(t) = (\hat{C}_1 - \hat{D}_{1,2}\hat{K}\hat{C}_2) \hat{x}(t) + \hat{D}_{1,1}\hat{w}(t) + \hat{D}_{1,2}\hat{r}(t) \\ \hat{y}(t) = \hat{C}_2\hat{x}(t). \end{cases} \tag{1}$$

Once that a stabilizing SOF matrix $\hat{K} = \begin{bmatrix} \hat{K}_1 & \hat{K}_2 & \hat{K}_3 \end{bmatrix}$ for the augmented system (1) has been found, we have $K_D = M\hat{K}_3$. Thus, $M = I + M\hat{K}_3C_2B_2$ from which we conclude that:

$$M = (I - \hat{K}_3C_2B_2)^{-1}.$$

Finally, we have:

$$\begin{cases} K_P = M\hat{K}_1 = (I - \hat{K}_3C_2B_2)^{-1} \hat{K}_1 \\ K_I = M\hat{K}_2 = (I - \hat{K}_3C_2B_2)^{-1} \hat{K}_2 \\ K_D = M\hat{K}_3 = (I - \hat{K}_3C_2B_2)^{-1} \hat{K}_3. \end{cases} \tag{2}$$

We conclude that the stabilization problem via PID controller is reducible to the stabilization via SOF problem, with the constraint that $I - \hat{K}_3C_2B_2$ is invertible. In the following we will assume that the couples (\hat{A}, \hat{B}_2) and (\hat{A}^T, \hat{C}_2^T) are controllable, in order to make the task of finding a feasible starting SOF \hat{K} easier.

Remark 1. Please note that although the method relies on a specific known state-space model of the plant, it does not require an exact model, since the modeling errors can be included in the dynamic uncertainty and they can be taken into account by selecting a nominal model with a suitable weighting function (see [20]). In any case, after designing a PID controller for the nominal model, its performance should be checked on the non-linear plant model. Note also that the requirement that $I - \hat{K}_3C_2B_2$ should be invertible is not a problem since that if a stabilizing SOF \hat{K} exists, there exist an open neighborhood of it where all the members are stabilizing SOF's, where on the other hand, the set of all matrices that does not satisfy the invertibility condition has measure 0.

Remark 2. Since the problem of SOF is (polynomial-time) reducible to the PID controller problem (by restriction to $K_I = 0, K_D = 0$), therefore the PID controller problem is as hard as the SOF problem. Since the SOF problem with structural constraints (on K_P), or the problem of exact pole-placement via SOF are NP-hard, it follows that these problems with PID controllers are NP-hard. We therefore conclude that unless $P = NP$, no deterministic polynomial-time algorithm can solve these problems and we therefore are led to use randomization with the hope that randomized algorithms will be able to cope more efficiently with these problems.

3. Statement of the Problem

The H_∞ and H_2 norm minimization relates to the transfer function $T_{\hat{w},\hat{z}}(s)$ from the noise \hat{w} to the output \hat{z} (when the reference signal \hat{r} is zero and the system is derived only by the noise). The transfer function is given by:

$$T_{\hat{w},\hat{z}}(s) = \hat{D}_{1,1} + (\hat{C}_1 - \hat{D}_{1,2}\hat{K}\hat{C}_2)(sI - \hat{A} + \hat{B}_2\hat{K}\hat{C}_2)^{-1}\hat{B}_1. \tag{3}$$

In order that the H_2 -norm of $T_{\hat{w},\hat{z}}(s)$ would be finite, we need that $\hat{D}_{1,1} = 0$. We therefore have the following lemma (given without a proof):

Lemma 1. We have:

$$\hat{D}_{1,1} = D_{1,1} - D_{1,2}M^{-1}K_D C_2 B_1 = D_{1,1} - D_{1,2}\hat{K}_3 C_2 B_1 = 0, \tag{4}$$

if and only if:

$$\left((C_2 B_1)^T \otimes D_{1,2} \right) \cdot \left((C_2 B_1)^T \otimes D_{1,2} \right)^+ \text{vec}(D_{1,1}) = \text{vec}(D_{1,1}). \tag{5}$$

In this case, the set of all matrices \hat{K}_3 satisfying (4) is given by:

$$\hat{K}_3 = \text{mat} \left(\left((C_2 B_1)^T \otimes D_{1,2} \right)^+ \text{vec}(D_{1,1}) + L_{(C_2 B_1)^T \otimes D_{1,2}} \text{vec}(Z) \right), \tag{6}$$

where Z is arbitrary matrix with the same size as \hat{K}_3 .

For the LQR minimization, we assume that $\hat{w}(t) = 0$ and $\hat{r}(t) = 0$, i.e., the system is derived by the initial state $\hat{x}_0 := \hat{x}(0)$ (initial step input). Assume that a SOF \hat{K} was found, such that $\hat{A}_{cl}(\hat{K}) := \hat{A} - \hat{B}_2\hat{K}\hat{C}_2$ is stable. Let the LQR functional be defined by:

$$J(\hat{x}_0) := \int_0^\infty \left(\hat{x}(t)^T Q \hat{x}(t) + \hat{z}(t)^T R \hat{z}(t) \right) dt, \tag{7}$$

where $Q > 0$ and $R \geq 0$. Let $\hat{C}(\hat{K}) := \hat{C}_1 - \hat{D}_{1,2}\hat{K}\hat{C}_2$. Then, $\hat{z}(t) = \hat{C}(\hat{K})\hat{x}(t)$ and substitution of the last into (7) yields

$$J(\hat{x}_0, \hat{K}) = \int_0^\infty \hat{x}(t)^T \left(Q + \hat{C}(\hat{K})^T R \hat{C}(\hat{K}) \right) \hat{x}(t) dt = \hat{x}_0^T P(\hat{K}) \hat{x}_0, \tag{8}$$

where $P(\hat{K}) > 0$ is the unique solution to the Lyapunov equation

$$\hat{A}_{cl}(\hat{K})^T P + P \hat{A}_{cl}(\hat{K}) = - \left(Q + \hat{C}(\hat{K})^T R \hat{C}(\hat{K}) \right) \tag{9}$$

given explicitly by

$$P(\hat{K}) = -\text{mat} \left(\left(I \otimes \hat{A}_{cl}(\hat{K})^T + \hat{A}_{cl}(\hat{K})^T \otimes I \right)^{-1} \cdot \text{vec} \left(Q + \hat{C}(\hat{K})^T R \hat{C}(\hat{K}) \right) \right). \tag{10}$$

Thus, we look for SOF \hat{K} that minimizes the functional $J(x_0, \hat{K}) = \hat{x}_0^T P(\hat{K}) \hat{x}_0$. When \hat{x}_0 is unknown, we seek a SOF \hat{K} for which

$$\sigma_{max}(\hat{K}) := \max(\sigma(P(\hat{K}))) \tag{11}$$

is minimal. In this case, we get a robust LQR via SOF, in the sense that it minimizes $J(\hat{x}_0, \hat{K})$ for the worst possible (unknown) \hat{x}_0 . Please note that

$$\begin{aligned} \hat{x}_0^T P(\hat{K}) \hat{x}_0 &= \left\| P(\hat{K})^{\frac{1}{2}} \hat{x}_0 \right\|^2 \leq \\ &\leq \left\| P(\hat{K})^{\frac{1}{2}} \right\|^2 \|\hat{x}_0\|^2 = \\ &= \|P(\hat{K})\| \|\hat{x}_0\|^2 = \sigma_{max}(\hat{K}) \|\hat{x}_0\|^2, \end{aligned}$$

and that there exists $\hat{x}_0 \neq 0$ for which equality holds. Therefore $\frac{J(\hat{x}_0, \hat{K})}{\|\hat{x}_0\|^2} \leq \sigma_{max}(\hat{K})$, with equality in the worst case.

Let Ω denote a subset of \mathbb{C}_- , which is symmetric with respect to the x -axis, containing a positive length segment of the real axis, with some neighborhood of it, and such that $\Omega = \overline{int(\Omega)}$ (i.e., Ω is the closure of its interior). Let $\mathcal{S}^{q \times r}$ denote the set of all matrices $\hat{K} \in \mathbb{R}^{q \times r}$, such that $\hat{A}_{cl}(\hat{K})$ is stable, i.e., $\sigma(\hat{A}_{cl}(\hat{K})) \subset \mathbb{C}_-$. By $\mathcal{S}_\Omega^{q \times r}$, we denote the set of all matrices $\hat{K} \in \mathbb{R}^{q \times r}$, such that $\sigma(\hat{A}_{cl}(\hat{K})) \subset \Omega$. In this case, we say that $\hat{A}_{cl}(\hat{K})$ is Ω -stable (or, the augmented system (1) is Ω -stable). In the sequel, we will occasionally write \mathcal{S}_Ω instead of $\mathcal{S}_\Omega^{q \times r}$, when it is clear what the size of the related matrices is. Please note that \mathcal{S}_Ω is a closed set, since Ω is closed and that \mathcal{S}_Ω has nonempty interior (if it is nonempty) since Ω has nonempty interior.

Under the assumptions that $D_{2,1} = 0$ and that $(\hat{A}, \hat{B}_2), (\hat{A}^T, \hat{C}_2^T)$ are controllable (and the additional assumption (5) when dealing with the H_2 -norm minimization), for the PID controller problem with minimal H_∞ -norm, we need to solve:

$$\begin{aligned} \min f_{H_\infty}(\hat{K}) &:= \|T_{\hat{w}, \hat{z}}\|_{H_\infty} \\ \text{such that:} & \\ \begin{cases} \hat{K} \in \mathcal{S}_\Omega \\ I - \hat{K}_3 C_2 B_2 \text{ is invertible} \end{cases} & \end{aligned} \tag{12}$$

where $T_{\hat{w}, \hat{z}}$ is given by (3). For the PID controller problem with minimal H_2 -norm, we need to solve:

$$\begin{aligned} \min f_{H_2}(\hat{K}) &:= \|T_{\hat{w}, \hat{z}}\|_{H_2} \\ \text{such that:} & \\ \begin{cases} \hat{K} \in \mathcal{S}_\Omega \\ \hat{K}_3 \text{ has the structure (6)} \\ I - \hat{K}_3 C_2 B_2 \text{ is invertible} \end{cases} & \end{aligned} \tag{13}$$

where $T_{\hat{w}, \hat{z}}$ is given by (3) and, for the PID controller problem with minimal LQR functional-value, we need to solve:

$$\begin{aligned} \min f_{LQR}(\hat{K}) &:= \sigma_{max}(\hat{K}) \\ \text{such that:} & \\ \begin{cases} \hat{K} \in \mathcal{S}_\Omega \\ I - \hat{K}_3 C_2 B_2 \text{ is invertible} \end{cases} & \end{aligned} \tag{14}$$

where $\sigma_{max}(\hat{K})$ is given by (11).

4. The Suggested Algorithm

Assume that $\hat{K}^{(0)} \in \text{int}(\mathcal{S}_\Omega)^{q \times r}$ was found by the Ray-Shooting algorithm ([6]) or by any other method (see [21–23]). We assume that $I - \hat{K}_3^{(0)} C_2 B_2$ is invertible (and in addition, we assume that $\hat{K}_3^{(0)}$ has the form (6) for some initial matrix $Z^{(0)}$, when dealing with the H_2 -norm minimization). Let $h > 0$ and let $U^{(0)}$ be a unit vector w.r.t. the Frobenius norm, i.e., $\|U^{(0)}\|_F = 1$. Let $L^{(0)} = \hat{K}^{(0)} + h \cdot U^{(0)}$ and let \mathcal{L} be the hyperplane defined by $L^{(0)} + V$, where $\langle V, U^{(0)} \rangle_F = 0$. Let $r_\infty > 0$ and let \mathcal{R}_∞ denote the set of all $F \in \mathcal{L}$, such that $\|F - L^{(0)}\|_F \leq r_\infty$. Let $\mathcal{R}_\infty(\epsilon) = \mathcal{R}_\infty + \overline{\mathbb{B}(0, \epsilon)}$, where $\overline{\mathbb{B}(0, \epsilon)}$ denotes the closed ball centered at 0 with radius ϵ ($0 < \epsilon \leq \frac{1}{2}$), with respect to the Frobenius norm on $\mathbb{R}^{q \times r}$. Let $\mathcal{D}^{(0)} = CH(K^{(0)}, \mathcal{R}_\infty(\epsilon))$ denote the convex-hull of the vertex $\hat{K}^{(0)}$ with the basis $\mathcal{R}_\infty(\epsilon)$. Let $\mathcal{S}_\Omega^{(0)} = \mathcal{S}_\Omega \cap \mathcal{D}^{(0)}$ and please note that $\mathcal{S}_\Omega^{(0)}$ is compact (but generally not convex). We wish to minimize a continuous function $f(\hat{K})$ over the compact set $\mathcal{S}_\Omega \cap \overline{\mathbb{B}(\hat{K}^{(0)}, h)}$. Let $\hat{K}_* \in \mathcal{S}_\Omega \cap \overline{\mathbb{B}(\hat{K}^{(0)}, h)}$ denote a point in $\mathcal{S}_\Omega \cap \overline{\mathbb{B}(\hat{K}^{(0)}, h)}$ where a global minimum of $f(\hat{K})$ is accepted. Obviously, $\hat{K}_* \in \mathcal{D}^{(0)}$, for some direction $U^{(0)}$, as above. When dealing with the H_2 -norm minimization, and if $U^{(0)} = \begin{bmatrix} U_1^{(0)} & U_2^{(0)} & U_3^{(0)} \end{bmatrix}$ then, the directions will be constrained to have the form $U_3^{(0)} = \text{mat}\left(L_{(C_2 B_1)^T \otimes D_{1,2}} \text{vec}(Z)\right)$, in order that $L^{(0)}$ would have the form (6). We will also restrict V as we have restricted $U^{(0)}$, in order that any element of \mathcal{L} would have the structure (6).

The suggested Algorithm 1 goes as follows:

We start with a point $\hat{K}^{(0)} \in \text{int}(\mathcal{S}_\Omega)$, found by the Ray-Shooting algorithm [6]. Assuming that $\hat{K}_* \in \mathcal{D}^{(0)}$, the inner-loop ($j = 1, \dots, n$) uses the Ray-Shooting Method in order to find an approximation of the global minimum of the function $f(\hat{K})$ over $\mathcal{S}_\Omega^{(0)}$ - the portion of \mathcal{S}_Ω bounded in the cone $\mathcal{D}^{(0)}$. The proof of convergence in probability of the inner-loop and its complexity can be found in [6] (see also [7]). In the inner-loop, we choose a search direction by choosing a point F in $\mathcal{R}_\infty(\epsilon)$ - the base of the cone $\mathcal{D}^{(0)}$. Next, in the most inner-loop ($k = 1, \dots, s$) we scan the ray $\hat{K}(t) := (1 - t)\hat{K}^{(0)} + tF$ and record the best controller on it. Repeating this sufficiently many times, we reach \hat{K}_* (or an ϵ -neighborhood of it) with high probability, under the assumption that $\hat{K}_* \in \mathcal{D}^{(0)}$. In [8] it was shown that by taking $h = r_\infty$ and $m = \lceil e \cdot \sqrt{2\pi\ell} \rceil$ iterations in the outer-loop (where $\ell := qr$), we have $\hat{K}_* \in \mathcal{D}^{(0)}$ almost surely. Specifically, when $\ell \geq 12$, it was suggested to take $m = 2\ell$ and $\ell \times \ell$ orthogonal matrix $U = \begin{bmatrix} u_1 & u_2 & \dots & u_\ell \end{bmatrix}$, and to take the directions $U_j^{(0)} = \pm \text{mat}(u_j)$, $j = 1, \dots, \ell$ in the outer-loop.

The outer-loop ($i = 1, \dots, m$) is used instead of executing the Ray-Shooting algorithm again and again, by taking $\hat{K}^{(\text{best})}$ as the new vertex of the search cone instead of $\hat{K}^{(0)}$. The replacement of $\hat{K}^{(0)}$ can be avoided if r_∞ is chosen sufficiently large. The replacement of $\hat{K}^{(0)}$ by $\hat{K}^{(\text{best})}$ can be considered as a heuristic step, which is made instead of running the Ray-Shooting algorithm many times in order to generate "the best starting point", which is relevant only if we actually evaluate $f(\hat{K})$ on each such point and take the point with the best value as the best starting point. Since in any case, we evaluate $f(\hat{K})$ in the main algorithm, we could avoid the repeated execution of the Ray-Shooting algorithm. The outer-loop is similar to what is done in the Hide-And-Seek algorithm (see [24,25]). The convergence in probability of the Hide-And-Seek algorithm can be found in [26].

Algorithm 1 Optimal PID controller randomized algorithm

Require: An algorithm for deciding Ω -stability and an algorithm for computing $f(\hat{K})$.

Input: $\hat{K}^{(0)}$ such that $\hat{A}_{cl}(\hat{K}^{(0)})$ is Ω -stable and $I - \hat{K}_3^{(0)}C_2B_2$ is invertible (with the needed structure (6) if H_2 -norm minimization is sought), $\epsilon > 0, r_\infty > 0, h > 0$ and integers m, n, s .

Output: \hat{K}_* that globally minimize $f(\hat{K})$ (or within ϵ distance from a global minimum of $f(\hat{K})$) in h -radius closed-ball centered at $\hat{K}^{(0)}$.

1. $\hat{K}^{(best)} \leftarrow \hat{K}^{(0)}$
2. $f^{(best)} \leftarrow f(\hat{K}^{(0)})$
3. $i \leftarrow 0$
4. $k \leftarrow 1$
5. **while** $i \leq m$ **do**
6. $i \leftarrow i + 1$
7. choose $U^{(0)}$ such that $\|U^{(0)}\|_F = 1$, uniformly at random
8. set $L^{(0)} \leftarrow \hat{K}^{(0)} + h \cdot U^{(0)}$
9. **for** $j = 1$ to n **do**
10. choose $F \in \mathcal{R}_\infty(\epsilon)$ uniformly at random
11. **for** $k = 1$ to s **do**
12. $t \leftarrow \frac{k}{s}$
13. $\hat{K}(t) \leftarrow (1 - t)\hat{K}^{(0)} + tF$
14. **if** $\hat{K}(t) \in \mathcal{S}_\Omega$ **then**
15. **if** $f(\hat{K}(t)) < f^{(best)}$ **then**
16. $\hat{K}^{(best)} \leftarrow \hat{K}(t)$
17. $f^{(best)} \leftarrow f(\hat{K}(t))$
18. **end if**
19. **end if**
20. **end for**
21. **end for**
22. **if** $i \geq k \lceil e \cdot \sqrt{2\pi\ell} \rceil$ **then**
23. $\hat{K}^{(0)} \leftarrow \hat{K}^{(best)}$
24. $k \leftarrow k + 1$
25. **end if**
26. **end while**
27. **if** $I - \hat{K}_3^{(best)}C_2B_2$ is invertible **then**
28. **return** $\hat{K}^{(best)}, f^{(best)}$
29. **else**
30. **return** failure
31. **end if**

Remark 3. Please note that $\hat{K}^{(0)}$ should satisfy the feasibility condition that $I - \hat{K}_3^{(0)}C_2B_2$ is invertible, in order to raise the probability that $I - \hat{K}_3^{(best)}C_2B_2$ would be invertible. Indeed, one can execute the algorithm without this condition, i.e., execute the algorithm from an infeasible point.

Note also that during the optimization in the main loop of Algorithm 1, we do not check this feasibility condition and we may pass to a better point which might not be feasible, because such points might lead to a better feasible points. This is made because the set of non-feasible points has measure 0 and thus, we should not be disturbed by the possibility of temporarily stepping on infeasible points, during the search. The feasibility condition is checked at the end of Algorithm 1 and experience with the algorithm shows that almost always we end up with a feasible point, when we start from a feasible point (which is almost always the case, if $\hat{K}^{(0)}$ was found).

5. Experimental Section

To synthesize LQR or H_∞ optimal PID controller, we execute the Ray-Shooting algorithm ([6]) in order to get a starting SOF $\hat{K}^{(0)}$ such that the augmented system (1) is Ω -stable and $I - \hat{K}_3^{(0)}C_2B_2$ is invertible. Next, we apply Algorithm 1 in order to get $\hat{K}^{(best)}$ from which the PID controller is derived,

using (2). For the H_2 optimal PID controller, we execute the Ray-Shooting algorithm, in order to get a starting SOF $\hat{K}^{(0)}$, as above. Next, we execute the algorithm from [7], in order to get a SOF $\hat{K}^{(1)}$ with the structure (6). This step is the most demanding step as can be seen from the following example. Finally, we execute Algorithm 1 in order to get $\hat{K}^{(best)}$ from which the PID controller is derived, using (2).

Example 1. For the AC1 system (taken from COMPl_{ib}—see [27–29]) with the following state-space model:

$$A = \begin{bmatrix} 0 & 0 & 1.132 & 0 & -1 \\ 0 & -0.0538 & -0.1712 & 0 & 0.0705 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.0485 & 0 & -0.8556 & -1.013 \\ 0 & -0.2909 & 0 & 1.0532 & -0.6859 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.03593 & 0 & 0.01672 \\ 0 & 0.00989 & 0 \\ 0 & -0.07548 & 0 \\ 0 & 0 & 0.05635 \\ 0.00145 & 0 & 0.06743 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 0 & 0 \\ -0.12 & 1 & 0 \\ 0 & 0 & 0 \\ 4.419 & 0 & -1.665 \\ 1.575 & 0 & -0.0732 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 0.707106781186547 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.707106781186547 & 0 & 0 \end{bmatrix},$$

and

$$D_{1,1} = 0_{2 \times 3}$$

$$D_{1,2} = \begin{bmatrix} 0.707106781186547 & 0 & 0 \\ 0 & 0.707106781186547 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D_{2,1} = 0_{3 \times 3}$$

with

$$\sigma(A) = \left\{ \begin{matrix} 0 \\ -0.780052457480417 \pm 1.029636749384622i \\ -0.017597542519583 \pm 0.182585375210158i \end{matrix} \right\},$$

let the objective be to find a PID LQR-optimal controller, where the closed-loop eigenvalues should be in the rectangular region defined by:

$$\Omega = \{z \in \mathbb{C} \mid -1 \leq \Re(z) \leq -0.1, |\Im(z)| \leq 1\}.$$

With the following parameters for the algorithm: $m = 100, n = 100, s = 100, h = r_\infty = 100, \epsilon = 10^{-16}, Q = I_8, R = I_2$ we had the following feasible SOF

$$\hat{K}^{(0)} = \begin{bmatrix} \hat{K}_1^{(0)} & \hat{K}_2^{(0)} & \hat{K}_3^{(0)} \end{bmatrix},$$

where

$$\hat{K}_1^{(0)} = \begin{bmatrix} -1.139087350327457 & -0.076805805847702 & -1.175733971595813 \\ 0.175620269295905 & 1.249662743416654 & -0.394592760970558 \\ -2.524143751375688 & -0.262506326222373 & -3.380417685135832 \end{bmatrix}$$

$$\hat{K}_2^{(0)} = \begin{bmatrix} -0.265188982290714 & 0.082432589634872 & -0.236137372902669 \\ 0.106177753653144 & 0.443637620004996 & -0.074457362520092 \\ -0.527328403725350 & 0.200910957443737 & -0.850599462623321 \end{bmatrix}$$

$$\hat{K}_3^{(0)} = \begin{bmatrix} -1.120493844818183 & 0.190581999014316 & -0.522738193473694 \\ -0.036425300944087 & -0.028600461083115 & -0.183279009240151 \\ -3.229623103192055 & 0.550761163041108 & -2.205172558841497 \end{bmatrix},$$

found by the Ray-Shooting algorithm in CPU – time = 0.0001[sec] and with LQR functional value $f_{LQR}(\hat{K}^{(0)}) = 16.746246825328360$. Executing Algorithm 1, we had

$$\hat{K}^{(best)} = \left[\hat{K}_1^{(best)} \quad \hat{K}_2^{(best)} \quad \hat{K}_3^{(best)} \right],$$

where

$$\hat{K}_1^{(best)} = \begin{bmatrix} -1.085160433924800 & -0.152380406023858 & -1.111840435453025 \\ 0.173224251962231 & 1.253268146513151 & -0.449756208653024 \\ -2.518304745419560 & -0.284285880826183 & -3.437786815141788 \end{bmatrix}$$

$$\hat{K}_2^{(best)} = \begin{bmatrix} -0.221989603556075 & 0.073112868340369 & -0.241805447889791 \\ 0.119158793726812 & 0.403709017174812 & -0.120711030008226 \\ -0.541730211464951 & 0.187380061037601 & -0.903873015737950 \end{bmatrix}$$

$$\hat{K}_3^{(best)} = \begin{bmatrix} -1.068327578247724 & 0.146908957308029 & -0.525373002842116 \\ -0.006608329355743 & 0.015609869717001 & -0.182159600265849 \\ -3.166355434538844 & 0.515059657329335 & -2.156140670934107 \end{bmatrix},$$

with $f_{LQR}(\hat{K}^{(best)}) = 13.601550793243616$ (a 18.778512% relative improvement), in CPU – time = 76.343750[sec]. The resulting PID controller matrices are:

$$K_P = \begin{bmatrix} -1.040671697032480 & 0.034045803431248 & -1.158219284410088 \\ 0.177951420403750 & 1.273076937581805 & -0.454684211704099 \\ -2.362328186885230 & 0.369320416016725 & -3.600390066086302 \end{bmatrix}$$

$$K_I = \begin{bmatrix} -0.200613800205014 & 0.131015509566988 & -0.255249027751546 \\ 0.121430088217771 & 0.409861485359054 & -0.122139482935926 \\ -0.466787102815339 & 0.390385480823640 & -0.951005918753228 \end{bmatrix}$$

$$K_D = \begin{bmatrix} -1.050500770135795 & 0.146612915752025 & -0.542836776832826 \\ -0.004714134766140 & 0.015578413703667 & -0.184015220520444 \\ -3.103855027173255 & 0.514021741975605 & -2.217368288447396 \end{bmatrix}$$

and the resulting closed-loop eigenvalues are

$$\sigma\left(\hat{A}_{cl}\left(\hat{K}^{(best)}\right)\right) = \left\{ \begin{array}{l} -0.915131493828863 \pm 0.914610403731497i \\ -0.881963764205315 \pm 0.191516911259354i \\ -0.730009436848610 \\ -0.616780364831122 \\ -0.329413066341451 \pm 0.071827905231358i \end{array} \right\}.$$

Example 2. For the same system as in Example 1 with the objective to find a PID H_∞ -norm optimal controller, where the closed-loop eigenvalues should be in the same rectangular region Ω , starting from the same $\hat{K}^{(0)}$ with $f_{H_\infty}(\hat{K}^{(0)}) = 0.095033139822324$, we had

$$\hat{K}^{(best)} = \begin{bmatrix} \hat{K}_1^{(best)} & \hat{K}_2^{(best)} & \hat{K}_3^{(best)} \end{bmatrix},$$

where

$$\hat{K}_1^{(best)} = \begin{bmatrix} -1.087304300685670 & -0.045931838436646 & -1.085574892719301 \\ 0.133605778924987 & 1.184070254573485 & -0.359343489961314 \\ -2.678924736751655 & -0.264961819769927 & -3.490264530663762 \end{bmatrix}$$

$$\hat{K}_2^{(best)} = \begin{bmatrix} -0.275424882288615 & 0.138303327101369 & -0.075398358659212 \\ 0.090631547438998 & 0.331640807629037 & -0.054696779167659 \\ -0.516085927047396 & 0.184332519865895 & -0.733373655058220 \end{bmatrix}$$

$$\hat{K}_3^{(best)} = \begin{bmatrix} -1.165473798084745 & 0.076978746950025 & -0.595094516639020 \\ -0.052626935909114 & -0.061681173045286 & -0.250098357975546 \\ -3.316496450850004 & 0.694282662038967 & -2.314168285879468 \end{bmatrix},$$

with $f_{H_\infty}(\hat{K}^{(best)}) = 0.072175978563672$ (a 24.051779% relative improvement), in CPU – time = 126[sec]. The resulting PID controller matrices are:

$$K_P = \begin{bmatrix} -1.068321790034687 & 0.039576561177247 & -1.102041006775961 \\ 0.118395560901336 & 1.115554481033627 & -0.346149598520451 \\ -2.507718675999537 & 0.506251036541779 & -3.638774839724999 \end{bmatrix}$$

$$K_I = \begin{bmatrix} -0.266534452864325 & 0.160949047585517 & -0.078679654648945 \\ 0.083507865416445 & 0.313495349833722 & -0.052067557547737 \\ -0.435901834009705 & 0.388577609900052 & -0.762968147108973 \end{bmatrix}$$

$$K_D = \begin{bmatrix} -1.159203620514524 & 0.071881042424526 & -0.607938725615309 \\ -0.057651074688413 & -0.057596508025015 & -0.239806609825754 \\ -3.259944795394558 & 0.648305713745055 & -2.430012099310619 \end{bmatrix}$$

and the resulting closed-loop eigenvalues are

$$\sigma(\hat{A}_{cl}(\hat{K}^{(best)})) = \left\{ \begin{array}{l} -0.965762603991179 \pm 0.955842860416991i \\ -0.507539612520429 \pm 0.539780823728534i \\ -0.929257645094484 \pm 0.106820410747074i \\ -0.402381802562664 \pm 0.042181153576293i \end{array} \right\}.$$

Example 3. For the system from Example 1, with the same Ω and the same $\hat{K}^{(0)}$, the algorithm from [7] failed to find structured SOF $\hat{K}^{(1)}$. Indeed, from another starting SOF, the above-mentioned algorithm succeeded to find a structured SOF. For

$$\hat{K}^{(0)} = \begin{bmatrix} \hat{K}_1^{(0)} & \hat{K}_2^{(0)} & \hat{K}_3^{(0)} \end{bmatrix},$$

where

$$\hat{K}_1^{(0)} = \begin{bmatrix} -0.001662379279286 & -0.330320315815391 & -0.204742383931633 \\ 0.107146675564809 & 0.994156785847185 & 0.016601261567192 \\ -0.352905474519842 & -0.942806073864177 & 0.096372856789800 \end{bmatrix}$$

$$\hat{K}_2^{(0)} = \begin{bmatrix} 0.021994917878383 & -0.051106382240571 & 0.041069916425995 \\ 0.041186023579647 & 0.283840911655084 & 0.016793398034650 \\ 0.000278730967043 & -0.136566568491544 & 0.140207270149126 \end{bmatrix}$$

$$\hat{K}_3^{(0)} = \begin{bmatrix} -0.066971504986511 & 0.041204949529465 & 0.574933673498160 \\ -0.152954829804652 & -0.044219575277023 & 0.285308855216965 \\ -1.286404253117917 & 0.132606565745259 & 1.514386124380330 \end{bmatrix},$$

found by the Ray-Shooting algorithm from [6], in CPU – time = 0.0001[sec], applying the algorithm from [7] we have found the structured SOF

$$\hat{K}^{(1)} = \begin{bmatrix} \hat{K}_1^{(1)} & \hat{K}_2^{(1)} & \hat{K}_3^{(1)} \end{bmatrix},$$

where

$$\hat{K}_1^{(1)} = \begin{bmatrix} -0.001662379279286 & -0.330320315815391 & -0.204742383931633 \\ 0.107146675564809 & 0.994156785847185 & 0.016601261567192 \\ -0.352905474519842 & -0.942806073864177 & 0.096372856789800 \end{bmatrix}$$

$$\hat{K}_2^{(1)} = \begin{bmatrix} 0.021994917878383 & -0.051106382240571 & 0.041069916425995 \\ 0.041186023579647 & 0.283840911655084 & 0.016793398034650 \\ 0.000278730967043 & -0.136566568491544 & 0.140207270149126 \end{bmatrix}$$

$$\hat{K}_3^{(1)} = \begin{bmatrix} -0.000000000000001 & 1.630868948596140 & 0.213689638336193 \\ -0.000000000000003 & 0.516222895350342 & 0.067639698397123 \\ -3.828454587894386 & 1.446059480160095 & 0.287354792999600 \end{bmatrix},$$

in CPU – time = 2126.84375[sec] = 35.447395[*min*], with objective-function value $f_{H_2}(\hat{K}^{(1)}) = 0.364564164927770$ (note the small values of the 1, 1 and the 2, 1 entries of $\hat{K}_3^{(1)}$ obtained from the structural constraints). Finally, executing Algorithm 1 with $\hat{K}^{(1)}$ as a starting SOE, we have found

$$\hat{K}^{(best)} = \begin{bmatrix} \hat{K}_1^{(best)} & \hat{K}_2^{(best)} & \hat{K}_3^{(best)} \end{bmatrix},$$

where

$$\hat{K}_1^{(best)} = \begin{bmatrix} 0.076825955660614 & -0.288411611380877 & -0.080218436303677 \\ 0.092501965527188 & 1.046837016022694 & 0.126682619794964 \\ -0.341825773843233 & -0.939476093514249 & -0.014192822579841 \end{bmatrix}$$

$$\hat{K}_2^{(best)} = \begin{bmatrix} 0.054840679045778 & -0.080221174899522 & 0.114230924697635 \\ 0.052402918799798 & 0.301314353311879 & -0.093055199101202 \\ -0.061395620772426 & -0.248330164720565 & 0.198088337294925 \end{bmatrix}$$

$$\hat{K}_3^{(best)} = \begin{bmatrix} -0.000000000000001 & 1.623084863796692 & 0.212669704596573 \\ -0.000000000000003 & 0.692852565144541 & 0.090783146121880 \\ -3.950508497484917 & 1.418275521031274 & -0.019182452364697 \end{bmatrix},$$

with $f_{H_2}(\hat{K}^{(best)}) = 0.212807638848134$ (a 41.626835% relative improvement), in CPU – time = 99.890625[sec]. The resulting PID controller matrices are:

$$K_P = \begin{bmatrix} 0.346143328950049 & 3.208734600167995 & 0.360572630013263 \\ 0.207466522885631 & 2.539677456718627 & 0.314844821170003 \\ -0.106492276227244 & 2.116381906236221 & 0.370976920044842 \end{bmatrix}$$

$$K_I = \begin{bmatrix} 0.203018492396510 & 0.925289371971877 & -0.231015500114683 \\ 0.115656159814819 & 0.730540561515251 & -0.240431886773992 \\ 0.068084340579495 & 0.630299798860383 & -0.103593086737016 \end{bmatrix}$$

$$K_D = \begin{bmatrix} -0.000000000000010 & 3.233767462898737 & 0.423714364176847 \\ -0.000000000000007 & 1.380410927195360 & 0.180872602940673 \\ -3.950508497484925 & 2.825713760036134 & 0.165231487536706 \end{bmatrix}$$

and the resulting closed-loop eigenvalues are

$$\sigma(\hat{A}_{cl}(\hat{K}^{(best)})) = \left\{ \begin{array}{l} -0.350618286842131 \pm 0.990354187385021i \\ -0.963555748849909 \pm 0.645732021985088i \\ -0.127768318435090 \pm 0.216451742947409i \\ \quad \quad \quad -0.159486067934620 \\ \quad \quad \quad -0.462031919768989 \end{array} \right\}.$$

Please note that

$$\hat{D}_{1,1} = D_{1,1} - D_{1,2}\hat{K}_3^{(1)}C_2B_1$$

$$= 10^{-16} \cdot \begin{bmatrix} 0.254063466480327 & 0.173472347597681 & 0.118228253814391 \\ 0.762190399440980 & -0.273218947466347 & 0.354684761443172 \end{bmatrix}$$

and

$$\hat{D}_{1,1} = D_{1,1} - D_{1,2}\hat{K}_3^{(best)}C_2B_1$$

$$= 10^{-16} \cdot \begin{bmatrix} 0.254063466480326 & 0.156125112837913 & 0.118228253814391 \\ 0.762190399440979 & -0.277555756156289 & 0.354684761443172 \end{bmatrix}$$

which we treat numerically as 0.

6. Concluding Remarks

In this article, a randomized method for synthesizing PID optimal controllers (for the standard LQR, H_∞ and H_2 optimization problems) for coupled MIMO systems was introduced. The method is based on a (direct) reduction of the problem to the problem of SOF (occasionally with some structure). The randomization circumvents the need to solve any SDP or BMI programs. Therefore, the method can be applied to more involved PID controller design that can be formulated as structured SOF problems, e.g., the design of 2-DOF-PID controllers and robust and gain-scheduled PID controllers for Linear Parameter Varying (LPV) systems (see [13]), without the need to solve any BMI programs and thus can be applied to larger systems. Moreover, the method has a proof of convergence (in probability) to a global optimum of the objective function.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Visioli, A. Research Trends for PID Controllers. *Acta Polytech.* **2012**, *52*, 144–150.
2. Suresh, A.; Meena, S.; Chitra, K. Controller design for mimo process using optimization algorithm. *Int. J. Pure Appl. Math.* **2017**, *117*, 163–170.
3. Zheng, F.; Wang, Q.G.; Lee, T.H. On the design of multivariable PID controllers via LMI approach. *Automatica* **2002**, *38*, 517–526. [[CrossRef](#)]
4. Saeki, M. Fixed structure PID controller design for standard H_∞ control problem. *Automatica* **2006**, *42*, 93–100. [[CrossRef](#)]
5. Boyd S.; Hast M.; Åström, K.J. MIMO PID tuning via iterated LMI restriction. *Int. J. Robust Nonlinear Control* **2016**, *26*, 1718–1731. [[CrossRef](#)]
6. Peretz, Y. A randomized approximation algorithm for the minimal-norm static-output-feedback problem. *Automatica* **2016**, *63*, 221–234. [[CrossRef](#)]
7. Peretz, Y. On applications of the Ray-Shooting method for structured and structured-sparse static-output-feedbacks. *Int. J. Syst. Sci.* **2017**, *48*, 1902–1913. [[CrossRef](#)]
8. Peretz, Y. On application of the Ray-Shooting Method for LQR via static-output-feedback. *Algorithms* **2018**, *11*, 8. [[CrossRef](#)]
9. Lin, C.; Wang, Q.G.; Lee, T.H. An improvement on multivariable PID controller design via iterative LMI approach. *Automatica* **2004**, *40*, 519–525. [[CrossRef](#)]
10. Cao, Y.Y.; Lam, J.; Sun, Y.X. Static Output Feedback Stabilization: An ILMI Approach. *Automatica* **1998**, *34*, 1641–1645. [[CrossRef](#)]
11. Cao, Y.Y.; Sun, Y.X.; Lam, J. Simultaneous Stabilization via Static Output Feedback and State Feedback. *IEEE Trans. Autom. Control* **1999**, *44*, 1277–1282.
12. Rosinová, D.; Veselý, V. Robust Static output feedback for discrete-time systems—LMI approach. *Periodica Polytech. Ser. El. Eng.* **2004**, *48*, 151–163.
13. Bianchi, F.D.; Mantz, R.J.; Christiansen, C.F. Multivariable PID control with set-point weighting via BMI optimisation. *Automatica* **2008**, *44*, 472–478. [[CrossRef](#)]
14. Blondel, V.; Tsitsiklis, J.N. NP-hardness of some linear control design problems. *SIAM J. Control Optim.* **1997**, *35*, 2118–2127. [[CrossRef](#)]
15. Mesbahi, M. A semi-definite programming solution of the least order dynamic output feedback synthesis problem. In Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, AZ, USA, 7–10 December 1999; Volume 2, pp. 1851–1856.
16. Zhang, H.; Shi, Y.; Mehr, A.S. Robust Static Output Feedback Control and Remote PID Design for Networked Motor Systems. *IEEE Trans. Ind. Electron.* **2011**, *58*, 5396–5405. [[CrossRef](#)]
17. Borges, R.A.; Calliero, T.R.; Oliveira, C.L.F.; Peres, P.L.D. Improved conditions for reduced-order H_∞ filter design as a static output feedback problem. In Proceedings of the American Control Conference 2011, San-Francisco, CA, USA, 29 June–1 July 2011.
18. Nemirovskii, A. Several NP-hard problems arising in robust stability analysis. *Math. Control Signal Syst.* **1993**, *6*, 99–105. [[CrossRef](#)]
19. Fu, M. Pole placement via static output feedback is NP-hard. *IEEE Trans. Autom. Control.* **2004**, *49*, 855–857. [[CrossRef](#)]
20. Sanchez Peña, R.; Sznaier, M. *Robust Systems, Theory and Applications*; Wiley: New York, NY, USA, 1998.
21. Henrion, D.; Lofberg, J.; Kočvara, M.; Stingl, M. Solving Polynomial static output feedback problems with PENBMI. In Proceedings of the IEEE Conference on Decision Control and European Control Conference, Sevilla, Spain, 12–15 December 2005.
22. Yang, K.; Orsi, R. Generalized pole placement via static output feedback: A methodology based on projections. *Automatica* **2006**, *42*, 2143–2150. [[CrossRef](#)]
23. Gumussoy, S.; Henrion, D.; Millstone, M.; Overton, M.L. Multiobjective Robust Control with HIFOO 2.0. In Proceedings of the IFAC Symposium on Robust Control Design, Haifa, Israel, 16–18 June 2009.
24. Romeijn, H.E.; Smith, R.L. Simulated Annealing for Constrained Global Optimization. *Glob. Optim.* **1994**, *5*, 101–126. [[CrossRef](#)]
25. Zabinsky, Z.B. *Stochastic Adaptive Search for Global Optimization*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2003.

26. Bélisle, C.J.P. Convergence Theorems for a Class of Simulated Annealing Algorithms on \mathbb{R}^d . *Appl. Probab.* **1992**, *29*, 885–895. [[CrossRef](#)]
27. Leibfritz, F. COMPl_eib: Constrained Matrix-Optimization Problem Library—A Collection of Test Examples for Nonlinear Semidefinite Programs, Control System Design and Related Problems. Available online: http://www.friedemann-leibfritz.de/COMPlib_Data/COMPlib_Main_Paper.pdf (accessed on 1 June 2018).
28. Leibfritz, F.; Lipinski, W. COMPl_eib 1.0—User Manual and Quick Reference. Available online: http://www.friedemann-leibfritz.de/COMPlib_Data/COMPlib_User_Guide.pdf (accessed on 1 June 2018).
29. Leibfritz, F. Description of the Benchmark Examples in COMPl_eib 1.0. Available online: http://www.friedemann-leibfritz.de/COMPlib_Data/COMPlib_Test_Set.pdf (accessed on 1 June 2018).



© 2018 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).