INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2016; **26**:1718–1731
Published online 25 June 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3376

MIMO PID tuning via iterated LMI restriction

Stephen Boyd[1], Martin Hast[2,*,†] and Karl Johan Åström[2]

[1]*Electrical Engineering, Stanford University, Stanford, CA, 94305, USA*
[2]*Department of Automatic Control, Lund University, SE-22100, Lund, Sweden*

SUMMARY

We formulate multi-input multi-output proportional integral derivative controller design as an optimization problem that involves nonconvex quadratic matrix inequalities. We propose a simple method that replaces the nonconvex matrix inequalities with a linear matrix inequality restriction, and iterates to convergence. This method can be interpreted as a matrix extension of the convex–concave procedure, or as a particular majorization–minimization method. Convergence to a local minimum can be guaranteed. While we do not know that the resulting controller is globally optimal, the method works well in practice, and provides a simple automated method for tuning multi-input multi-output proportional integral derivative controllers. The method is readily extended in many ways, for example, to the design of more complex, structured controllers. Copyright © 2015 John Wiley & Sons, Ltd.

Received 5 March 2015; Revised 21 May 2015; Accepted 28 May 2015

KEY WORDS: MIMO PID; controller design; linear matrix inequalities; convex optimization

1. INTRODUCTION

Single-input single-output (SISO) proportional integral derivative (PID) control is the automatic control scheme most widely used in practice, with a long history going back at least 250 years; see [1, Section 1.4]. It has only three parameters to tune, and achieves reasonable or good performance on a wide variety of plants. The effect of the tuning parameters (or gains) on the closed-loop performance is well-understood, and there are well-known simple rules for tuning these parameters; see, for example, [1, Chap. 6] or [2]. Systems for automatically tuning SISO PID controllers have been developed, and are available in commercial controllers [1, 3–5]. The authors of this paper recently developed yet another SISO PID tuning method in [6], which is a precursor for the method described in this paper.

Single-input single-output PID controllers have been used for multiple-input multiple-output (MIMO) plants for many years. This is generally carried out by pairing inputs (actuators) and outputs (sensors), and connecting them with SISO PID controllers. These SISO PID controllers can be tuned one at a time (in 'successive loop closure') using standard SISO PID tuning rules. For MIMO plants that are already reasonably well-decoupled, multi-loop SISO PID design can work well. Unlike SISO PID design, however, MIMO PID design is more complex; the SISO loops have to be chosen carefully, and then tuned the correct way in the correct order.

An alternative to multi-loop SISO PID control is to design one MIMO PID controller, which uses matrix coefficients, all at once. Such a controller potentially uses all sensors to drive all actuators, but it is possible to specify a simpler structure by imposing a sparsity constraint on the controller gain matrices. Like SISO PID controllers for SISO plants, MIMO PID

*Correspondence to: Martin Hast, Department of Automatic Control, Lund University, Box 118, SE-22100, Lund, Sweden.
†E-mail: martin.hast@control.lth.se

Copyright © 2015 John Wiley & Sons, Ltd.

controllers can achieve very good performance on a wide variety of MIMO plants, even when the plant dynamics are quite coupled. The challenge is in tuning MIMO PID controllers, which require the specification of three matrices, each with a number of entries equal to the number of plant inputs times the number of plant outputs. For example, a MIMO PID controller for a plant with four inputs and four outputs requires the specification of up to 48 parameters. This would be very difficult, if not impossible, to tune by hand one parameter at a time. Hand tuning a MIMO PID controller with 10 inputs and 10 outputs would be impossible in practice.

In this paper, we describe a method for designing MIMO PID controllers. The method is based on solving a small number of convex optimization problems, specifically semidefinite programs (SDPs), which can be carried out efficiently. Our method is a local optimization method, and we cannot guarantee that it finds the globally optimal controller parameter values. On many examples, however, the method seems to work very well.

We first describe a basic form for the method. We impose constraints on the sensitivity and complementary sensitivity transfer functions, which guarantees closed-loop stability and a MIMO stability margin, and also a limit on actuator effort. The objective is to minimize the low-frequency sensitivity of the closed-loop system, which is an MIMO analog of maximizing the integral gain in the SISO case. In a later section, we describe a number of generalizations of the method. These optimization problems could be solved using other methods, for example, [7] and its references for a number of H_∞ synthesis approaches.

There is an enormous literature on automated SISO PID tuning, and a very large literature on MIMO PID tuning (e.g., [8] and its references or [9, 10]) including some methods that are very close to the one we describe, and others that are close in spirit. We give a more detailed technical analysis of other methods in Section 6.4, after describing the details of our method. But we mention here some earlier work that is very closely related to ours. In [11], the authors also form a linear matrix inequality (LMI)-based restriction of a problem with nonconvex quadratic matrix inequalities, and iterates to convergence. Another previous work that is close in spirit to ours is in [12], which formulates the design problem as involving bilinear matrix inequalities (BMIs), which are in turn solved (approximately) by iteratively solving a set of SDPs. (The connection between our method and BMI formulations will be discussed in Section 6.4.) The closest prior work appears in [13], which takes a very similar approach to ours. The authors consider MIMO PID design, using frequency-domain specifications, and their algorithm involves LMI restrictions of nonconvex matrix inequalities, as our method does. We will discuss some of the differences in Section 6.4.

2. MODEL AND ASSUMPTIONS

2.1. Plant

The linear time-invariant plant has m inputs (actuators) and p outputs (sensors), and is given by its transfer function $P(s) \in \mathbf{C}^{p\times m}$ or more specifically by its frequency response $P(i\omega)$, for $\omega \in \mathbf{R}_+$. We do not assume that P is rational; it can, for example, include transport delay. We assume that the entries of the plant input $u(t) \in \mathbf{R}^m$ are measured in appropriate units (or scaled), so their sizes are (roughly) the same order. We make the same assumption about the plant output $y(t) \in \mathbf{R}^p$. These assumptions justify the use of the (unweighted) ℓ_2-norm to measure the actuator effort and deviation of the plant outputs from the reference signal, and more generally, it justifies the use of (unweighted) matrix norms to measure closed-loop gains.

We make several assumptions about the plant. We assume that $p \leqslant m$, that is, there are at least as many actuators as plant outputs, and that $P(0)$ is full rank. This makes it possible to achieve perfect reference tracking in steady state ($s = 0$). We will also assume that the plant is stable and strictly proper, that is, $P(s) \to 0$ as $s \to \infty$. Most of our assumptions can be relaxed or extended to more general settings, but our goal is to keep the ideas simple for now. We will discuss various ways these assumptions can be relaxed in Section 7.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

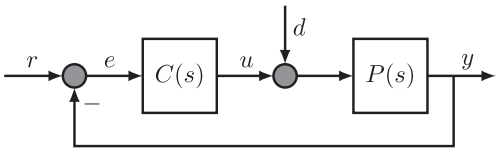

Figure 1. Classical feedback interconnection.

2.2. Proportional integral derivative controller

The controller is a PID controller, given by

$$C(s) = K_{\mathrm{P}} + \frac{1}{s}K_{\mathrm{I}} + \frac{s}{1+\tau s}K_{\mathrm{D}},$$

where $K_{\mathrm{P}}, K_{\mathrm{I}}, K_{\mathrm{D}} \in \mathbf{R}^{m\times p}$ are the *proportional gain matrix*, *integral gain matrix*, and *derivative gain matrix*, respectively. The $3mp$ entries in these matrices are the design parameters we are to choose. The constant $\tau > 0$ is the derivative action time constant, and is assumed to be fixed and responsibly chosen, for example, a modest fraction of the desired closed-loop response time.

The plant and controller are connected in the classical loop shown in Figure 1, described by the equations

$$e = r - y, \quad u = Ce, \quad y = P(u + d),$$

where r is the reference input, e is the error, and d is an input-referred plant disturbance. The signals u and y are the plant input and output, respectively.

2.3. Closed-loop transfer functions

We will be interested in several closed-loop transfer functions that we describe here.

Sensitivity. The transfer function from reference input r to error e is the *sensitivity* $S = (I + PC)^{-1}$. The size of S gives a measure of the tracking error; for low frequencies, S should be small. When $S(0) = 0$ (a constraint we will impose), we have perfect static tracking. The maximum size of S, which occurs near the crossover frequency, is closely related to closed-loop damping and system stability.

Q-parameter. The transfer function from r to u is denoted Q, defined as $Q = C(I + PC)^{-1}$. Its size is a measure of the actuator effort. (We use Q to match the notation often used in Youla's parametrization of closed-loop transfer functions; see [14].)

Complementary sensitivity function. The complementary sensitivity function is $T = PC(I + PC)^{-1}$. It is the closed-loop transfer function from r to y. It is near the identity for low frequencies, and will be small for high frequencies; its maximum size is also related to closed-loop damping.

These three closed-loop transfer functions are sufficient to guarantee a sensible controller design (because P is assumed stable). They are related in various ways, for example, we have

$$S + T = I, \quad T = PQ, \quad Q = CS.$$

Other closed-loop transfer functions. Several other closed-loop transfer functions can also be considered. For example, the closed-loop transfer function from the plant disturbance d to the tracking error e is $R = -(I + PC)^{-1}P$. It will be clear how our approach extends to other transfer functions.

2.4. Notation

For a complex matrix $Z \in \mathbf{C}^{p\times q}$, Z^* is its (Hermitian) conjugate transpose, $\|Z\|$ denotes the spectral norm, that is, the maximum singular value. For full rank Z, we let $\sigma_{\min}(Z)$ denote its minimum singular value. A square matrix is Hermitian if $Z = Z^*$. Between Hermitian matrices, the symbol $\succeq 0$ is used to denote matrix inequality, so $Z \succeq 0$ means that Z is Hermitian and positive semidefinite. We use the notation $Z^{-*} = (Z^*)^{-1}$.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

For a $p \times q$ transfer function H, $\|H\|_\infty$ is its $\mathbf{H}_\infty$-norm, $\|H\|_\infty = \sup_{\Re s \geq 0} \|H(s)\|$, which can be expressed as

$$\|H\|_\infty = \sup_{\omega \geq 0} \|H(i\omega)\|,$$

when H is stable (i.e., $H \in \mathbf{H}_\infty^{p\times q}$).

3. DESIGN PROBLEM

3.1. Objective and constraints

Sensitivity and complementary sensitivity peaking. We require $\|S\|_\infty \leqslant S_{\max}$, where $S_{\max} > 1$. Reasonable values of $S_{\max}$ are in the range 1.1 to 1.6; lower values give a more damped closed-loop system. This constraint ensures closed-loop stability. We also require $\|T\|_\infty \leqslant T_{\max}$, with $T_{\max} > 1$. Reasonable values of $T_{\max}$ are similar to those for $S_{\max}$.

Static and low-frequency sensitivity. Assuming that $P(0)K_{\mathrm{I}}$ is nonsingular, we have $S(0) = 0$, which means that we have zero error for constant reference signals. The next term in the expansion of the sensitivity near $s = 0$ is $S(s) \approx s(P(0)K_{\mathrm{I}})^{-1}$ for $|s|$ small. Our objective will be to attain the best possible low-frequency sensitivity, which means that we will minimize $\|(P(0)K_{\mathrm{I}})^{-1}\|$. This objective indirectly imposes the condition that we achieve perfect tracking because when it is finite, we have $P(0)K_{\mathrm{I}}$ nonsingular.

Actuator authority limit. We require that $\|Q\|_\infty \leqslant Q_{\max}$. This sets a maximum value for the size of the closed-loop actuator signal in response to the reference signal. Because the plant is stable, this constraint ensures that the closed-loop system is stable; see [14].

We can determine a reasonable value for $Q_{\max}$ as follows. Any controller that has a finite objective value (i.e., achieves perfect reference tracking at $s = 0$) satisfies $T(0) = I = P(0)Q(0)$; this has the simple interpretation that at $s = 0$, the controller inverts the plant. It follows that $\|Q(0)\| \geqslant 1/\sigma_{\min}(P(0))$, from which we conclude that $\|Q\|_\infty \geqslant 1/\sigma_{\min}(P(0))$. Thus, we must have $Q_{\max} \geqslant 1/\sigma_{\min}(P(0))$. The right-hand side can be interpreted as the minimum actuator effort (measured by $\|Q\|_\infty$) required to achieve static tracking. A reasonable value for $Q_{\max}$ is therefore a modest multiple of $1/\sigma_{\min}(P(0))$, say, 3 to 10.

Design problem. Putting it all together, we obtain the problem

$$\begin{array}{ll} \text{minimize} & \|(P(0)K_{\mathrm{I}})^{-1}\|, \\ \text{subject to} & \|S\|_\infty \leqslant S_{\max}, \\ & \|T\|_\infty \leqslant T_{\max}, \\ & \|Q\|_\infty \leqslant Q_{\max}. \end{array} \tag{1}$$

The variables to be chosen are the coefficient matrices $K_{\mathrm{P}}, K_{\mathrm{I}}, K_{\mathrm{D}}$; the problem data are the plant transfer function P, the controller derivative time constant τ, and the design parameters $S_{\max}$, $T_{\max}$, and $Q_{\max}$. This problem is not convex because the cost function and constraints are not affine expressions of the coefficient matrices. Note that the objective contains the implied constraint that $P(0)K_{\mathrm{I}}$ is invertible, which implies that perfect static tracking is achieved for constant reference inputs.

3.2. Sampling semi-infinite constraints

The constraints on the closed-loop transfer functions can be expressed as, for example,

$$\|S(i\omega)\| \leqslant S_{\max}, \quad \forall \omega \geqslant 0.$$

This is a so-called semi-infinite constraint, because it consists of an infinite number of constraints, one for each $\omega \geqslant 0$. Semi-infinite constraints such as these (with one parameter, ω) are readily handled by choosing a reasonable finite (but large) set of frequency samples $0 < \omega_1 < \cdots < \omega_K$, and replacing the semi-infinite constraints with the finite set of constraints at each of the given frequencies. For example, we replace the constraint $\|S\|_\infty \leqslant S_{\max}$ with $\|S(i\omega_k)\| \leqslant S_{\max}$, $k = 1, \ldots, N$. Our optimization method has a computational complexity that grows linearly with

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

N, so we can choose a large enough value of N (say, several hundred or more) that this sampling has no practical effect. By 'reasonable', we mean that the frequency sampling is fine enough to catch any rapid changes in the closed-loop transfer function with frequency, and also cover an appropriate range; in particular, we assume that at ω_1 and ω_N, the transfer functions are near their asymptotic values,

$$S(0) = 0, \quad T(0) = I, \quad Q(0) = K_{\mathrm{I}}(P(0)K_{\mathrm{I}})^{-1}$$

and

$$S(\infty) = I, \quad T(\infty) = 0, \quad Q(\infty) = K_{\mathrm{P}} + (1/\tau)K_{\mathrm{D}},$$

respectively.

We will use subscripts to denote a transfer function evaluated at the frequency $s = i\omega_k$. For example, $P_k = P(i\omega_k)$, which is a (given) complex matrix. Note that the quantity $C_k = C(i\omega_k)$ is a complex matrix, and is an affine function of the design variables K_{P}, K_{P}, K_{D}. (We will use this observation to arrive at the LMI restriction in Section 5.)

The sampled problem is then

$$\begin{array}{ll} \text{minimize} & \|(P(0)K_{\mathrm{I}})^{-1}\|, \\ \text{subject to} & \|S_k\| \leqslant S_{\max}, \\ & \|T_k\| \leqslant T_{\max}, \\ & \|Q_k\| \leqslant Q_{\max}, \\ & k = 1, \ldots, N. \end{array} \tag{2}$$

This problem has $3N$ constraints, each of which has the form of a matrix norm inequality. The arguments of the matrix norm inequalities, however, are complex functions of the design variables K_{P}, K_{I}, K_{D}, given by the various formulas for the closed-loop transfer functions.

4. QUADRATIC MATRIX INEQUALITY FORM

In this section, we show how the (frequency sampled) design problem (2) can be casted in a simple form in which every constraint has the same *quadratic matrix inequality* (QMI) form

$$Z^*Z \succeq Y^*Y, \tag{3}$$

where both Z and Y are affine functions of the variables.

We start with the objective. First, we note that we can just as well maximize $\sigma_{\min}(P(0)K_{\mathrm{I}}) = 1/\|(P(0)K_{\mathrm{I}})^{-1}\|$. We introduce a new scalar variable t, which we maximize subject to $\sigma_{\min}(P(0)K_{\mathrm{I}}) \geqslant t$. (This is the standard epigraph transformation; see [15, Section 4.2.4].) We then observe that

$$\sigma_{\min}(P(0)K_{\mathrm{I}}) \geqslant t \Leftrightarrow (P(0)K_{\mathrm{I}})^*(P(0)K_{\mathrm{I}}) \succeq t^2 I,$$

which has the form (3) with $Z = P(0)K_{\mathrm{I}}$ and $Y = tI$, both of which are affine functions of the variables.

Now, consider the sensitivity peaking limit $\|S_k\| \leqslant S_{\max}$. As mentioned earlier, we have

$$\begin{aligned} \|S_k\| \leqslant S_{\max} &\Leftrightarrow (I + P_kC_k)^{-*}(I + P_kC_k)^{-1} \preceq S_{\max}^2 I \\ &\Leftrightarrow (I + P_kC_k)^*(I + P_kC_k) \succeq (1/S_{\max}^2)I, \end{aligned}$$

where in the second line, we have multiplied the left and right sides by $(I + P_kC_k)^*$ and by $(I + P_kC_k)$, respectively. This has the QMI form (3) with $Z = I + P_kC_k$ and $Y = (1/S_{\max})I$.

For the complementary sensitivity constraint $\|T_k\| \leqslant T_{\max}$, we have

$$\begin{aligned} \|T_k\| \leqslant T_{\max} &\Leftrightarrow (I + P_k^*C_k^*)^{-1}C_k^*P_k^*P_kC(I + P_kC_k)^{-1} \preceq T_{\max}^2 I \\ &\Leftrightarrow (I + P_kC_k)^*(I + P_kC_k) \succeq (1/T_{\max}^2)(P_kC_k)^*(P_kC_k). \end{aligned}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

This has the QMI form (3) with $Z = I + P_k C_k$ and $Y = (1/T_{\max}) P_k C_k$.

In a similar way, we have

$$\|Q_k\| \leq Q_{\max} \Leftrightarrow (I + P_k C_k)^*(I + P_k C_k) \succeq (1/Q_{\max}^2) C_k^* C_k,$$

which has the QMI form with $Z = I + P_k C_k$ and $Y = (1/Q_{\max}) C_k$.

We arrive at a problem with the form

$$\begin{array}{l} \text{maximize } t \\ \text{subject to } Z_k^* Z_k \succeq Y_k^* Y_k, \quad k = 1, \ldots, M, \end{array} \tag{4}$$

with variables t, $K_{\mathrm{P}}, K_{\mathrm{I}}, K_{\mathrm{D}}$. (Here, we have $M = 3N + 1$.) This is the PID controller design problem in QMI form.

5. LINEAR MATRIX INEQUALITY RESTRICTION

We first show how to form an (convex) LMI restriction for the QMI $Z^*Z \succeq Y^*Y$. (See [16] for background on matrix inequalities.) The QMI is already convex in Y, so we can focus on Z. We start with the simple matrix inequality

$$0 \preceq (Z - \tilde{Z})^*(Z - \tilde{Z}) = Z^*Z - Z^*\tilde{Z} - \tilde{Z}^*Z + \tilde{Z}^*\tilde{Z},$$

valid for any matrices Z and $\tilde{Z}$. Re-arranging, we obtain

$$Z^*Z \succeq Z^*\tilde{Z} + \tilde{Z}^*Z - \tilde{Z}^*\tilde{Z}.$$

The left-hand side is a quadratic function of Z; the right-hand side is an affine function of Z. It follows that the matrix inequality

$$Z^*\tilde{Z} + \tilde{Z}^*Z - \tilde{Z}^*\tilde{Z} \succeq Y^*Y,$$

which is convex in (Z, Y), implies $Z^*Z \succeq Y^*Y$; that is, it is a convex restriction of the QMI. We can write this convex QMI as an LMI

$$\begin{bmatrix} Z^*\tilde{Z} + \tilde{Z}^*Z - \tilde{Z}^*\tilde{Z} & Y^* \\ Y & I \end{bmatrix} \succeq 0. \tag{5}$$

(Here, Z and Y are the variables; $\tilde{Z}$ is an arbitrary matrix.) For any matrix $\tilde{Z}$, the LMI (5) implies the QMI $Z^*Z \succeq Y^*Y$. We call it the *LMI restriction* of the QMI, obtained at point $\tilde{Z}$.

The LMI restriction of the QMI is illustrated in Figure 2, for the simple case of real scalar Z and Y. In this case, the QMI is $z^2 \geqslant y^2$, which gives the two lightly shaded cones in the figure, with blue boundary. The LMI restriction at $\tilde{z}$ is given by $2\tilde{z}z - \tilde{z}^2 \geqslant y^2$, shown as the shaded region bounded by the parabola, with red boundary. The two boundaries touch at the point where $z = \tilde{z}$.

Now, consider the QMI form PID controller design problem (4). Given any matrices $\tilde{Z}_1, \ldots, \tilde{Z}_M$, we can form the *LMI restricted problem*

$$\begin{array}{l} \text{maximize } t \\ \text{subject to } \begin{bmatrix} Z_k^*\tilde{Z}_k + \tilde{Z}_k^* Z_k - \tilde{Z}_k^*\tilde{Z}_k & Y_k^* \\ Y_k & I \end{bmatrix} \succeq 0, \quad k = 1, \ldots, M. \end{array} \tag{6}$$

This problem has linear objective and LMI constraints, and so is an SDP [17]. It is readily solved (globally). Note that any solution of the LMI restriction is feasible for the QMI problem. The LMI restriction, however, need not be feasible; this depends on the choice of $\tilde{Z}_k$.

Simplifications. In the inequalities associated with S_k, the matrix Y_k does not depend on the controller parameters (i.e., is constant). In this case, we can work directly with the smaller (equivalent) LMIs

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

$$Z_k^* \tilde{Z}_k + \tilde{Z}_k^* Z_k - \tilde{Z}_k^* \tilde{Z}_k \succeq Y_k^* Y_k,$$

which can give a modest computational advantage.

Another simplification is to maximize the variable $w = t^2$; in this case, the single inequality associated with the objective can be handled as the smaller LMI

$$Z_1^* \tilde{Z}_1 + \tilde{Z}_1^* Z_1 - \tilde{Z}_1^* \tilde{Z}_1 \succeq wI,$$

where $Z_1 = P(0)K_\mathrm{I}$. This simplification leads to slightly faster convergence, because t is only an upper bound on $\|(P(0)K_\mathrm{I})^{-1}\|$, whereas $w^{-1/2}$ is actually equal to the objective, after optimization.

6. THE METHOD

6.1. *Controller initialization*

We first initialize the controller with

$$K_\mathrm{P} = 0, \quad K_\mathrm{I} = \epsilon P(0)^\dagger, \quad K_\mathrm{D} = 0,$$

where ϵ is small and positive, and $P(0)^\dagger = P(0)^T(P(0)P(0)^T)^{-1}$ is the pseudo-inverse of the DC gain. For small enough ϵ, this controller is feasible. Indeed, as $\epsilon \to 0$, we have

$$S(s) \to \frac{s}{\epsilon + s} I, \quad T(s) \to \frac{\epsilon}{\epsilon + s} I, \quad Q(s) \to \frac{\epsilon}{\epsilon + s} P(0)^\dagger,$$

from which it follows that the constraints $\|S\|_\infty \leqslant S_\mathrm{max}$, $\|T\|_\infty \leqslant T_\mathrm{max}$, and $\|Q\|_\infty \leqslant Q_\mathrm{max}$ are feasible for small enough ϵ (assuming $S_\mathrm{max} > 1$, $T_\mathrm{max} > 1$, and $Q_\mathrm{max} > 1/\sigma_\mathrm{min}(P(0))$). Note that $\|Q\|_\infty$ finite implies closed-loop stability of this initial controller.

6.2. *Iteration*

We then repeat the following steps. We form the LMI restriction (6), using $\tilde{Z}_k = Z_k^\mathrm{curr}$, where Z_k^curr is the current value of Z_k. This choice guarantees that the LMI restriction is feasible. We solve this SDP to obtain the updated values of the design variables. These are feasible, because they were constrained by the restrictions, and the objective t (which is an upper bound on the original objective $\|(P(0)K_\mathrm{I})^{-1}\|$) cannot decrease.

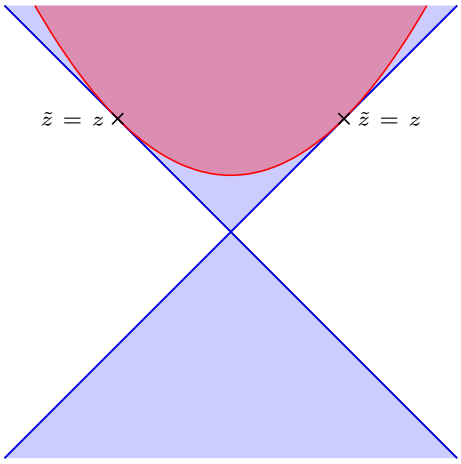

Figure 2. LMI restriction (shown in red) of QMI (shown in blue) for simple case when Z and Y are scalar and real.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

6.3. Convergence

The iterates are all feasible (and we have closed-loop stability since $\|Q\|_\infty$ is finite), and the objective is nonincreasing. Because it is non-negative, the objective converges. We can stop when not much progress is being made, which is typically after 10 or fewer iterations. At convergence, the optimal value of t, which is in general an *upper bound* on the original objective $\|(P(0)K_I)^{-1}\|$, is actually *equal* to this value. (When we optimize with the variable w, it is directly equal to the objective.)

6.4. Connections, interpretations, and prior work

Our method is similar to, but not the same as, the convex–concave method for the SISO case described in [6]. In that paper, we linearize a scalar inequality of the form $|Z| \geqslant |Y|$, which is not the same as linearizing (as we do here) the quadratic inequality $|Z|^2 \geqslant |Y|^2$. The idea of linearizing concave terms in an otherwise convex optimization problem, which gives a convex restriction, is an old one that has been (re-)invented many times, for many applications; see, for example, the references in [18] or [19]. The idea of linearizing a matrix inequality is given in [18].

The convex–concave procedure is in turn a special case of a very general method for finding a local minimum of a nonconvex optimization problem. In each iteration, we replace the objective function and each constraint function by a convex majorization that is tight at the given point, and solve the resulting convex problem. This idea traces back at least to 1970 [20], and has been widely used since then; see [18].

There is also a very close connection of our method to BMIs and methods for them, such as alternating minimization over the two groups of variables. The connection is easiest to see and states when Z and Y have the same dimensions (in general, the inequality $Z^*Z \succeq Y^*Y$ implies that they have the same number of columns). Define

$$U = (1/2)(Z + Y), \qquad V = (1/2)(Z - Y),$$

so $Z = U + V$ and $Y = U - V$. Then we have

$$\begin{aligned} Z^*Z \succeq Y^*Y &\Leftrightarrow (U+V)^*(U+V) \succeq (U-V)^*(U-V) \\ &\Leftrightarrow U^*U + V^*U + U^*V + V^*V \succeq U^*U - V^*U - U^*V + V^*V \\ &\Leftrightarrow V^*U + U^*V \succeq 0, \end{aligned}$$

which we recognize as a BMI in U and V. Thus, our QMI can be expressed as an equivalent BMI. MIMO PID design via BMIs is discussed in [12].

The closest prior work is [13], which contains many of the ideas we use in the present paper. The authors develop quadratic matrix inequalities similar to the ones we use here, and derive a method for MIMO PID design that uses LMI restrictions, as we do. While the two methods are clearly closely related, we are unable to derive our exact algorithm from theirs. We can identify several differences in the approach. First, we consider separate closed-loop transfer functions (e.g., S, T, and Q), where they lump them together into one block closed-loop transfer function. One advantage of considering these classical closed-loop transfer functions separately is that we can give simple and universal choices for the upper bound (such as 1.4, for example, for S). Second, we consider stable plants, which allow us to give a simple low-gain PID initialization. Finally, we consider a generic QMI, and develop a simple universal LMI restriction, which allows us to derive a simple algorithm for design of MIMO PID controllers that minimize low-frequency sensitivity subject to classical constraints on S, T, and Q.

7. EXTENSIONS AND VARIATIONS

In this section, we list various extensions and variations on the MIMO PID controller design problem, starting with simple ones and moving to more complex ones. While the basic iteration will work in all of these variations, the design must start from a feasible initial controller, which may be a challenge to find, depending on the variation. (We make more specific comments about this in the succeeding text.)

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

Exchanging objectives and constraints. As always, we can exchange constraints and objective; for example, we could impose a constraint on $\|(P(0)K_{\mathrm{I}})^{-1}\|$, and instead minimize another objective, such as $\|Q\|_\infty$. In this example, we would be minimizing actuator effort for a given fixed limit on the low-frequency sensitivity.

Frequency-dependent bounds. The bounds $S_{\max}, T_{\max}, Q_{\max}$ could be functions of frequency. This could be used to shape the various closed-loop transfer functions in more sophisticated ways than described here. This extension is also used in, for example, [13].

Other closed-loop transfer functions. The same approach works for other closed-loop transfer functions. For example, consider $R = -(I + PC)^{-1}P$, which is the closed-loop transfer function from disturbance to error. A limit on R, say, $\|R\|_\infty \leq R_{\max}$, can be expressed as a QMI as follows:

$$\|R_k\| \leq R_{\max} \Leftrightarrow (I + P_k C_k)(I + P_k C_k)^* \succeq (1/R_{\max}^2) P_k P_k^*,$$

which has the QMI form with $Z = (I + P_k C_k)^*$ and $Y = (1/R_{\max})P_k^*$. (Note the Hermitian conjugates in this case.) (The method proposed in [13] can also accommodate constraints on these other transfer functions.)

Low-frequency disturbance optimization. We can optimize low-frequency values of R instead of S. At low frequencies, we have

$$R(s) = -S(s)P(s) \approx -s(P(0)K_{\mathrm{I}})^{-1}P(0),$$

and we arrive at a very similar problem, which is also easily expressed in our QMI form.

High-frequency roll-off. Our method relies only on the fact that $C(s)$ is a linear function of the design variables $K_{\mathrm{P}}, K_{\mathrm{I}}, K_{\mathrm{D}}$. This allows us to use many other variations on the PID controller. As an example of simple variation that is very useful in practice, we can use the controller

$$C(s) = \left(\frac{1}{1 + s\tau + (s\tau)^2/2}\right)\left(K_{\mathrm{P}} + \frac{1}{s}K_{\mathrm{I}} + sK_{\mathrm{D}}\right),$$

where $\tau > 0$ is a (fixed) time constant. Here, we have an ideal PID controller, with a second-order high-frequency roll-off.

Unstable plants. The method can be extended to handle unstable plants, but in this case, the initial controller must stabilize and also satisfy the constraints; see, for example, [6]. (To satisfy the constraints, they can initially be relaxed.) When initialized this way, all iterates (and the final controller design) will be stabilizing. More sophisticated methods (say, with the addition of slack variables) could handle the case when an initial stabilizing controller is not known; for example, the method presented in [13] handles unstable plants.

More outputs than actuators. We can handle the case when $p > m$ (more outputs than actuators), so perfect static tracking cannot be achieved. In this case, we cannot have zero sensitivity at $s = 0$, but we can optimize over the value of $S(0)$, for example, minimize or limit its norm.

Convex constraints on controller parameters. Any convex constraints on the controller parameters can be imposed. For example, we can limit the values of any of the coefficients. A very interesting option here is to limit the sparsity pattern of C by requiring some entries to be zero. This gives structured MIMO PID controller design [21].

The simple initialization method described in Section 6.1 will generally not work when the controller parameters are constrained, for example, when a specific sparsity pattern is imposed.

Convex cost terms. We can add any convex function of the controller parameters to the objective. For example, we can add regularization to the objective, that is, a function that encourages the controller parameters to be small. The classical example is the sum of squares term

$$\lambda \sum_{ij} \left((K_{\mathrm{P}})_{ij}^2 + (K_{\mathrm{I}})_{ij}^2 + (K_{\mathrm{D}})_{ij}^2\right),$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

where $\lambda > 0$ is a parameter used to trade off low-frequency rejection and the size of the controller parameters (measured by the sum of squares). This can be useful to reduce injection of measurement noise into the control loop.

A very interesting regularization is one that encourages sparsity in the controller parameters, such as

$$\lambda \sum_{ij} \max\{(K_{\mathrm{P}})_{ij}, (K_{\mathrm{I}})_{ij}, (K_{\mathrm{D}})_{ij}\}.$$

This regularization will encourage sparsity in $C(s)$; for similar work; see, for example, [22] (for sparse controller design) and [23] (for sparsity of blocks of regressors in statistics).

Closed-loop convex constraints. We can also add any constraint or objective term that is convex in the closed-loop transfer functions; see the book in [14]. For example, we could include time-domain constraints such as a maximum step response settling time. The very same method (with some added terms to handle the added constraints) will work.

Robustness to plant variations. We can wrap robustness to plant variations into the method. A particularly simple (but very effective) method that gives robustness is to require that the constraints hold not only for one plant but also for several or many plausible values of the plant transfer function. This leads to a bigger problem to solve, but the same method works.

More general controllers. Finally, it should be clear that, just as in the method presented in [13], our method works for any linearly parametrized controller, and not just the simple PID structure that we have focused here. For more general structures, the design initialization can become a challenge, however.

8. EXAMPLES

In this section, we describe numerical results for a classic MIMO plant, the two-input two-output Wood–Berry binary distillation column described in [24]. The computations were carried out using the Matlab-based convex modeling framework CVX [25, 26] using the SDPT3 software [27, 28] for solving the SDP.

The plant transfer function is

$$P(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.2+1} \end{bmatrix}.$$

Each entry is a first-order system with a time delay. The dynamics are quite coupled, so finding a good MIMO PID controller is not simple. Several design methods and actual designs for this plant have been proposed in the literature, including [29–31]. Our method produced quite similar results, with the same or better metrics judged by our objectives (naturally).

We used design parameters

$$S_{\max} = 1.4, \quad T_{\max} = 1.4, \quad Q_{\max} = 3/\sigma_{\min}(P(0)) = 0.738.$$

The derivative action time constant is chosen to be $\tau = 0.3$. The semi-infinite constraints are sampled using $N = 300$ logarithmically spaced frequency samples in the interval $\left[10^{-3}, 10^{3}\right]$. The initial design uses the method described in Section 6.1 with $\epsilon = 0.01$.

The algorithm converges in seven iterations (which takes a 153 s to run in our simple implementation on a standard desktop computer with an Intel Core I7 processor) to the values

$$K_{\mathrm{P}} = \begin{bmatrix} 0.1750 & -0.0470 \\ -0.0751 & -0.0709 \end{bmatrix}, \quad K_{\mathrm{I}} = \begin{bmatrix} 0.0913 & -0.0345 \\ 0.0402 & -0.0328 \end{bmatrix}, \quad K_{\mathrm{D}} = \begin{bmatrix} 0.1601 & -0.0051 \\ 0.0201 & -0.1768 \end{bmatrix},$$

which achieve objective value $\|(P(0)K_{\mathrm{I}})^{-1}\| = 2.25$. The resulting closed-loop transfer function singular values are plotted versus frequency in Figure 3, along with the imposed limits.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

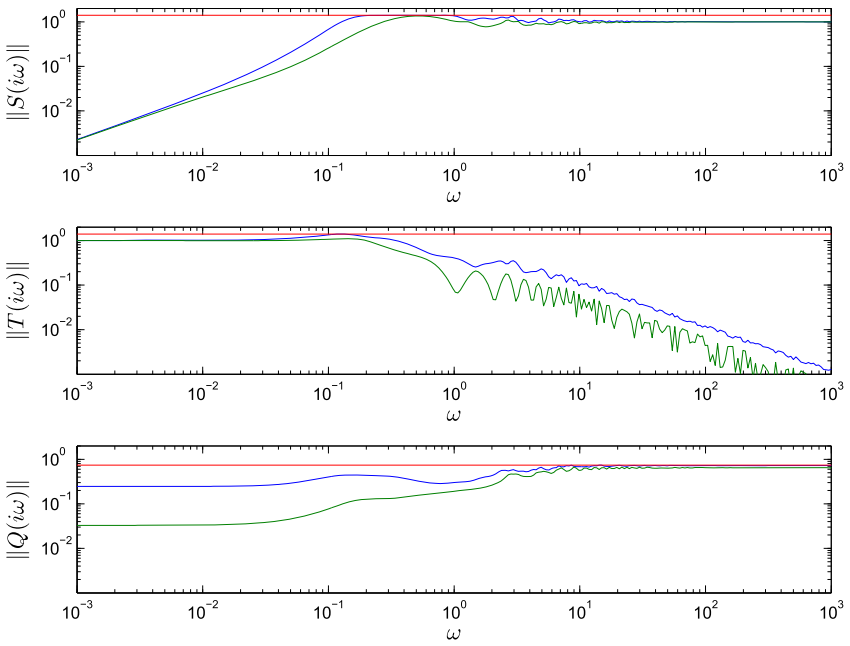

Figure 3. Closed-loop transfer function singular values versus frequency, with constraints shown in red.

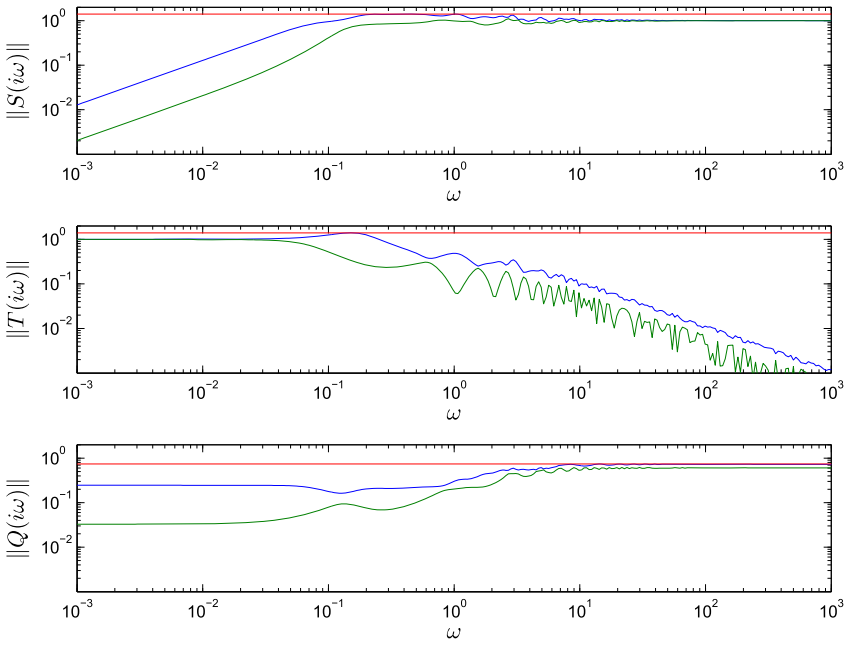

Figure 4. Closed-loop transfer function singular values versus frequency, with constraints shown in red, for diagonal PID design.

To demonstrate one simple extension, we also carry out MIMO PID design with the additional constraint that the controller is diagonal, that is, consists of two SISO PID loops. We initialize the algorithm with low-gain PI control from y_1 to u_1 and from y_2 to u_2, using the (diagonal) controller

$$K_{\mathrm{P}} = 10^{-3}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad K_{\mathrm{I}} = 10^{-3}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad K_{\mathrm{D}} = 0.$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

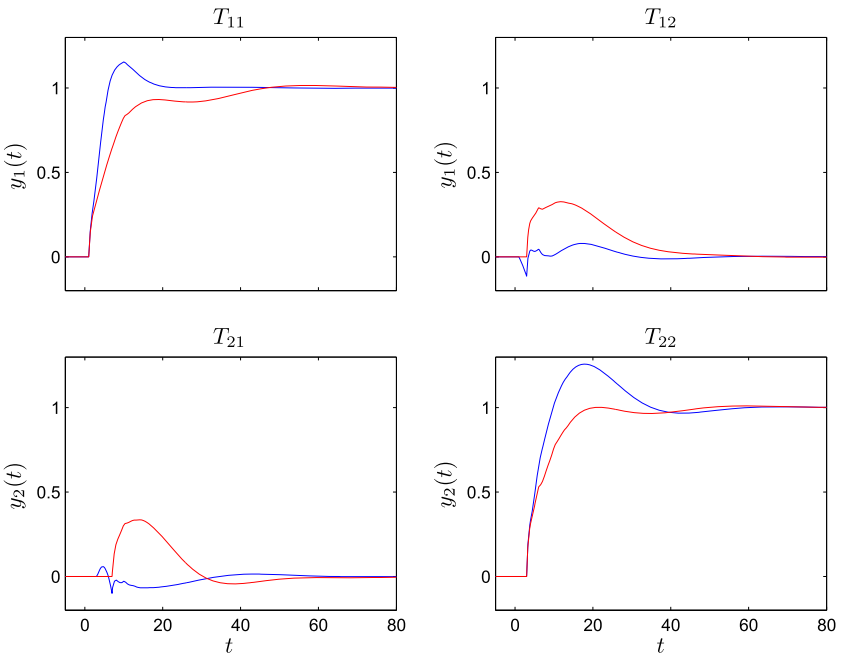

Figure 5. Closed-loop step response from r to y for the MIMO PID controller (blue) and the diagonal PID controller (red).

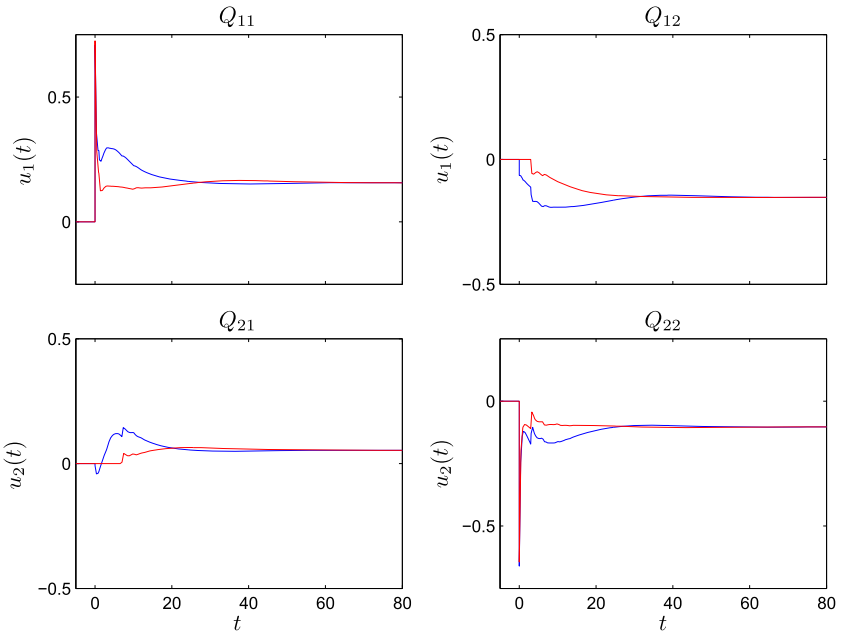

Figure 6. Closed-loop step response from r to u for the MIMO PID controller (blue) and the diagonal PID controller (red).

(The minus sign in the 2,2 entry is due to the negative 2,2 value of the 2,2 entry of $P(0)$.) The algorithm converges in eight iterations (taking a few minutes in our simple implementation) to the controller

$$K_{\mathrm{P}} = \begin{bmatrix} 0.1535 & 0 \\ 0 & -0.0692 \end{bmatrix}, \quad K_{\mathrm{I}} = \begin{bmatrix} 0.0210 & 0 \\ 0 & -0.0136 \end{bmatrix}, \quad K_{\mathrm{D}} = \begin{bmatrix} 0.1714 & 0 \\ 0 & -0.1725 \end{bmatrix},$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

which achieves objective value $\|(P(0)K_{\mathrm{I}})^{-1}\| = 13.36$, considerably worse than the objective value obtained with a general MIMO PID controller. The resulting closed-loop transfer function singular values are plotted in Figure 4. We can see that low-frequency sensitivity is considerably worse than that achieved by the full MIMO controller, for example, by noting the value of $\|S(\omega)\|_2$ for $\omega = 10^{-2}$.

The step responses of T, the transfer function from r to y, are plotted in Figure 5, for both the full MIMO PID controller and the diagonal PID controller. Here, too, we can observe the worse low-frequency rejection for the diagonal PID design, for example, in the larger off-diagonal entries of the step response.

The step responses of Q, the transfer function from r to u, are plotted in Figure 6, for both the full MIMO PID controller and the diagonal PID controller.

9. CONCLUSIONS

In this paper, we have described a simple method for effectively designing MIMO PID controllers for stable plants given by transfer function (at an appropriate set of frequencies). The method relies on solving a short sequence of SDPs (typically 10 or fewer), and although it cannot guarantee finding the globally optimal design, it appears to find very good designs in practical problems. The method is related to several other methods for MIMO PID design, and relies on ideas that have been used in several other contexts in optimization, such as the convex–concave procedure, and iterative convex restriction.

REFERENCES

1. Åström KJ, Hägglund T. *Advanced PID Control*. Instrumentation, Systems, and Automation Society: Research Triangle Park, NC, 2006.
2. Luyben WL. Simple method for tuning SISO controllers in multivariable systems. *Industrial & Engineering Chemistry Process Design and Development* 1986; **25**(3):654–660.
3. Garpinger O, Hägglund T. A software tool for robust PID design. *Proceedings 17th IFAC World Congress*, Seoul, Korea, 2008; 6416–6421.
4. Garpinger O, Hägglund T, Åström KJ. Criteria and trade-offs in PID design. *Ifac Conference on Advances in pid Control*, Brescia, Italy, March 2012; 47–52.
5. Vilanova R, Visioli A. *PID Control in the Third Millennium: Lessons Learned and New Approaches*. Springer Verlag: New York, 2012.
6. Hast M, Åström K, Bernhardsson B, Boyd S. PID design by convex–concave optimization. *Proceedings European Control Conference*, Zürich, Switzerland, July 2013; 4460–4465.
7. Apkarian P, Noll D. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control* 2006; **51**(1):71–86.
8. Åström KJ, Panagopoulos H, Hägglund T. Design of PI controllers based on non-convex optimization. *Automatica* 1998; **34**(5):585–601.
9. Panagopoulos H, Åström KJ, Hägglund T. Design of PID controllers based on constrained optimisation. *IEE Proceedings on Control Theory and Applications* 2002; **149**(1):32–40.
10. Saeki M, Kashiwagi K, Wada N. Design of multivariable H_∞ PID controller using frequency response. *Proceedings of the 2007 IEEE International Conference on Control Applications*, Singapore, October 2007; 1565–1570.
11. Lin C, Wang QG, Lee TH. An improvement on multivariable PID controller design via iterative LMI approach. *Automatica* 2004; **40**(3):519–525.
12. Bianchi FD, Mantz RD, Christiansen CF. Multivariable PID control with set-point weighting via BMI optimisation. *Automatica* 2008; **44**(2):472–478.
13. Saeki M, Ogawa M, Wada N. Low-order H_∞ controller design on the frequency domain by partial optimization. *International Journal of Robust and Nonlinear Control* 2010; **20**:323–333.
14. Boyd S, Barratt C. *Linear Controller Design—Limits of Performance*. Prentice-Hall: Englewood Cliffs, NJ, 1991.
15. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press: Cambridge, NY, 2004.
16. Boyd S, Ghaoui LE, Feron E, Balakrishnan V. *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics, vol. 15. SIAM: Philadelphia, PA, 1994.
17. Vandenberghe L, Boyd S. Semidefinite programming. *SIAM Review* March 1996; **38**(1):49–95.
18. Lipp T, Boyd S. Extensions and variations on the convex–concave procedure, 2014.
19. Yuille A, Rangarajan A. The concave–convex procedure. *Neural Computation* 2003; **15**(4):915–936.
20. Ortega R, Rheinboldt W. *Iterative Solutions of Nonlinear Equations in Several Variables*. Academic Press: New York, NY, 1970.
21. Saeki M. Fixed structure PID controller design for standard H_∞ control problem. *Automatica* 2006; **42**(1):93–100.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

22. Lin F, Fardad M, Jovanovic M. Sparse feedback synthesis via the alternating direction method of multipliers. *Proceedings of the American Control Conference*, Montreal, Canada, 2012; 4765–4770.
23. Zou H, Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2005; **67**(2):301–320.
24. Wood R, Berry M. Terminal composition control of a binary distillation column. *Chemical Engineering Science* 1973; **28**(9):1707–1717.
25. Research CVX. CVX: Matlab software for disciplined convex programming, version 2.0, 2012. (Available from: http://cvxr.com/cvx) [Accessed on 1 June 2015].
26. Grant M, Boyd S. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, Blondel V, Boyd S, Kimura H (eds)., Lecture Notes in Control and Information Sciences. Springer-Verlag Limited: London, 2008; 95–110.
27. Toh K-C, Todd MJ, Tütüncü RH. SDPT3—A Matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software* 1999; **11**(1-4):545–581.
28. Tütüncü RH, Toh K-C, Todd MJ. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming* 2003; **95**(2):189–217.
29. Dong J, Brosilow C. Design of robust multivariable PID controllers via IMC. *Proceedings of the American Control Conference* 1997; **5**:3380–3384.
30. Wang Q, Zou B, Lee T, Bi Q. Auto-tuning of multivariable PID controllers from decentralized relay feedback. *Automatica* 1997; **33**(3):319–330.
31. Tan W, Chen T, Marquez H. Robust controller design and PID tuning for multivariable processes. *Asian Journal of Control* 2002; **4**(4):439–451.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc