Available online at www.sciencedirect.com
ScienceDirect
IFAC Papers Online CONFERENCE PAPER ARCHIVE
IFAC PapersOnLine 50-1 (2017) 1539–1544

Discrete-Time Trials for Tuning without a Model ⋆

Franco Blanchini * **Gianfranco Fenu** ** **Giulia Giordano** ***
Felice Andrea Pellegrino **

* *Dipartimento di Scienze Matematiche, Informatiche e Fisiche, Università degli Studi di Udine, 33100 Udine, Italy.* blanchini@uniud.it
** *Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, 34127 Trieste, Italy.* {fenu,fapellegrino}@units.it
*** *Department of Automatic Control and LCCC Linnaeus Center, Lund University, Box 118, 221 00 Lund, Sweden.* giulia.giordano@control.lth.se

Abstract: Given a static plant described by a differentiable input-output function, which is completely unknown, but whose Jacobian takes values in a known polytope in the matrix space, we consider the problem of tuning the output (*i.e.*, driving the output to a desired value), by suitably choosing the input. To this aim, we assume to have at our disposal a discrete sequence of trials only, as it happens, for instance, when we iteratively run a software, with new input data at each iteration, in order to achieve the desired output value. In this paper we prove that, if the polytope is robustly non-singular (or has full row rank, in the general non-square case), then a suitable discrete-time tuning law drives the output to the desired point. The computation of the tuning law is based on a convex-optimisation problem to be solved on-line. An application example is proposed to show the effectiveness of the approach.

Keywords: Tuning, plant-tuning, model-free, discrete-time.

1. INTRODUCTION

We consider the problem of suitably determining the input vector of a static plant, so that the output vector assumes prescribed values. We work in the absence of a model: only information on the trends among variables is available. This amounts to knowing the sign of the partial derivatives along with (even rough) upper and lower bounds or, more in general, to knowing that the Jacobian of the unknown function is confined in a given polytope. We call this problem *model-free plant tuning*.

The tuning problem is particularly relevant to large-scale systems with several inputs and outputs, such as electrical networks, power generation systems, electronic circuits, systems for heat generation and transmission, and flow networks in general. For these systems, it is fundamental to tune the operating point. Yet, whenever the plant model is not known exactly, plant tuning often requires a frustrating trial-and-error approach: when attempting to set an output to the desired value, the unknown interactions among the variables can unpredictably drive the other outputs out of tune.

In several situations, continuous tuning is not possible: the input value cannot be continuously changed in time. Only a sequence of trials is viable, and these trials are by their nature discrete-time events. In general, not even a regular sampling can be assumed, so that the sequence of discrete events is not associated with a scheduled time sequence. This happens, for instance, when a software is iteratively run by changing the input data in order to get desired output values. Under the assumption that the system equations are unknown and only qualitative information on the system Jacobian is available, we want to choose the input sequence u_k for the plant, so as to drive the corresponding output sequence y_k to the desired value.

⋆ G.G. acknowledges support from the Swedish Research Council through the LCCC Linnaeus Center and the eLLIIT Excellence Center at Lund University.

We solve the model-free plant tuning problem by adopting a Lyapunov approach and we prove the following results.

- The robust discrete-time tuning problem of steering y_k to 0 (or any target value) can be solved by means of a proper tuning law, provided that the Jacobian matrix of the input-output function is included in a robustly non-singular (or robustly full-rank, in the non-square case) polytope.
- The *tuning scheme* is based on an auxiliary control variable, which is the increment v_k of the original control sequence $u_k \in \mathbb{R}^m$, namely, $v_k = u_k - u_{k-1}$.
- A Lyapunov-like positive-definite function of the output variable $y_k \in \mathbb{R}^p$ is considered. We prove that this function, which is non-increasing in general, is indeed decreasing when the non-singularity assumption is satisfied.
- The control computation requires the on-line solution of a convex optimisation problem.
- To illustrate the technique, we consider an example in which a software for designing a thermal plant produces proper outputs based on assigned input data. Typically, a single run of the code may require even hours, so that trial-and-error or gridding methods can be very inefficient and time-consuming; conversely, the proposed scheme provides convergence in a quite small number of steps.

The considered setup bears some resemblance to Broyden's quasi-Newton methods (Broyden 1965) for solving nonlinear vector equations $g(x) = 0$ without re-computing the Jacobian matrix at each iteration. Yet, to apply these methods, it is necessary to know the function g and to actually compute the Jacobian at the first iteration (and resort to an approximation based on a rank-one update at the following iterations).

When plant tuning can be performed in a continuous fashion, under the assumption that the Jacobian is confined in a polytope (or, more in general, in a convex and compact set) that is

10.1016/j.ifacol.2017.08.306

robustly non-singular, a continuous-time tuning scheme with guaranteed convergence has been proposed in our previous work (Blanchini et al. 2015, 2016). The results technically rely on the min-max theorem (Luenberger 1969) and on a suitable Lyapunov-like function. Similar approaches have been previously proposed in the literature for robust stabilisation (Gutman & Leitmann 1976; Gutman 1979; Meılakhs 1979; Blanchini 2000; Blanchini & Pesenti 2001); however, in the model-free plant-tuning case there is nothing to be stabilised and the Lyapunov-like function is not defined in the state-space. Possible analogies with methods for parameter tuning (Åström 1983; Fradkov 1980), iterative learning control (Ahn et al. 2007; Bristow et al. 2006), multi-dimensional extremum-seeking techniques (Tan et al. 2006; Khong et al. 2013; Nešić et al. 2013) and robust optimisation (Beyer & Sendhoff 2007) are thoroughly discussed by Blanchini et al. (2016).

Note that, in the discrete-time case, the scenario becomes completely different from the continuous-time case. In the discrete-time formulation of the problem, the inclusion in a generic convex and compact set is no longer enough for achieving the results. Moreover, a substantial technical difficulty arises: while the continuous-time result relies on the existence of a saddle point for a min-max zero-sum game, in discrete-time the saddle point does not exist, due to the lack of concavity of the functional for the maximiser. Hence, the min-max theorem (on which the continuous-time scheme is based) does no longer hold. Instead, as we show here, the tuning law is computed based on the on-line solution of a convex-optimisation problem.

2. PROBLEM STATEMENT

The problem we consider is the following: given an unknown function that relates the output y to the input u, we aim at driving the output to a desired value (which we can set to zero without loss of generality) by means of a suitable input sequence u. A crucial assumption is that the updating of u is performed at discrete time instants: u_k.

Problem 1. Given the static plant

$$y = g(u), \tag{1}$$

where $g : \mathbb{R}^m \to \mathbb{R}^p$, $p \leq m$, is a continuously differentiable function and $g(\bar{u}) = 0$ for some unique *unknown* $\bar{u}$, find a dynamic algorithm such that, as $k \to \infty$,

$$y_k \to 0, \tag{2}$$

$$u_k \to \bar{u}, \tag{3}$$

where $\bar{u}$ solves the equation

$$0 = g(u). \tag{4}$$

$\diamond$

Assumption 2. The following inclusion holds

$$G_u \doteq \left[\frac{\partial g}{\partial u}\right] \in \mathscr{G} \tag{5}$$

where $\mathscr{G}$ is a known polytope of matrices, with vertices G_i:

$$\mathscr{G} = \left\{ G = \sum_{i=1}^{r} G_i \alpha_i, \ \ \alpha_i \geq 0, \ \ \sum_{i=1}^{r} \alpha_i = 1 \right\}. \tag{6}$$

Remark 3. The elements u_k of the sequence are not necessarily equispaced in time (namely, there is no fixed “sampling time”).

Lemma 4. For any pair u and v, there exists a matrix $G_{u,v} \in \mathscr{G}$ (depending on u and v) such that

$$g(u+v) = g(u) + G_{u,v} v. \tag{7}$$

Proof. Based on a known formula (Khalil 1996),

$$g(u+v) = g(u) + \int_0^1 \nabla g(u+\sigma v) d\sigma \ v.$$

The result descends from the fact that, since $\nabla g \in \mathscr{G}$, and $\mathscr{G}$ is a convex set, also its integral over the interval $[0,1]$ (which is the average) is in $\mathscr{G}$ (see Blanchini et al. 2015 for details). $\square$

The lemma stated above suggests to choose, as the decision variable for our problem, the increment v_k of u_k:

$$v_k = u_{k+1} - u_k. \tag{8}$$

Then, according to Lemma 4, the sequence

$$y_{k+1} = g(u_{k+1}) = g(u_k + v_k) \tag{9}$$

can be equivalently expressed as

$$\begin{aligned} y_{k+1} &= g(u_{k+1}) = g(u_k + v_k) \\ &= g(u_k) + G(k) v_k = y_k + G(k) v_k, \end{aligned}$$

for some *unknown* sequence $G(k) \in \mathscr{G}$. Since we ignore both function g and matrix G_u, we consider the uncertain model

$$y_{k+1} = y_k + G(k) v_k, \quad G(k) \in \mathscr{G}, \tag{10}$$

and face the problem of driving y_k to zero *for any* $G(k) \in \mathscr{G}$, based exclusively on the knowledge of the polytope $\mathscr{G}$ in (6).

The set of trajectories of system (10) is richer than the set of trajectories of the original system (9). Therefore, any solution of the problem with system (10) solves the problem with system (9) as well. This is in principle conservative; yet, since we ignore the function g, we need to provide a robust solution.

3. MAIN RESULT

In this section we consider the tuning problem, Problem 1, under the assumption that there are as many outputs as inputs, hence $p = m$. We begin by recalling the fundamental definition of robust non-singularity (Barmish 1994).

Definition 5. The polytope $\mathscr{G}$ is *robustly non-singular* if every matrix in $\mathscr{G}$ is non-singular.

Then, the main result is the following.

Theorem 6. Given the discrete-time system (10), there exists an algorithm $v = \Phi(y)$ that drives y_k to 0 for any sequence $G(k) \in \mathscr{G}$ and for any initial value y_0 if and only if $\mathscr{G}$ is robustly non-singular. $\square$

Proof. ***Necessity***. Assume, by contradiction, that there exists a singular matrix $\bar{G} \in \mathscr{G}$, and let $\zeta \neq 0$ be a unit vector in its left kernel: $\zeta^\top \bar{G} = 0$. For any k and any value y_k that has a nonzero component along ζ, we can write $y_k = z_k + w_k$, where $z_k = (\zeta^\top y_k)\zeta$ is the component along ζ and w_k the component orthogonal to ζ. For $G(k) = \bar{G}$, we have

$$\zeta^\top y_{k+1} = \zeta^\top (z_k + w_k + \bar{G} v_k) = \zeta^\top z_k = \zeta^\top y_k.$$

This means that, if $G(k) = \bar{G}$, the component of y_k along ζ does not decrease (hence, y_k cannot be driven to zero), no matter how v_k is taken.

Sufficiency. A constructive proof is given in Section 3.1. $\square$

Remark 7. As mentioned above, considering the uncertain system (10) is not equivalent to addressing Problem 1, for which the condition of Theorem 6 is sufficient only. Our approach is conservative: non-singularity of the whole polytope would not be necessary in principle, but we need to adopt a robust approach because we do not know which is the actual

g and whether the actual $\partial g/\partial u$ is non-singular. Robust non-singularity is crucial for the proposed method, which can be regarded as a robust Gauss-Newton scheme. Note that, for large systems, checking robust non-singularity of the matrix polytope is computationally hard (Gurvits & Olshevsky 2009).

Corollary 8. Assume that the polytope $\mathcal{G}$ is robustly non-singular and let $G_1, G_2, \ldots, G_r$ be its vertices, according to (6). The control law v is achieved by computing on-line the *unique* solution of a convex optimisation problem:

$$v = \Phi(y) \doteq \arg\min_{v\in\mathbb{R}^m} \max_{i\in\{1,\ldots,r\}} \|y + G_i v\|. \qquad (11)$$

Proof. See Section 3.1. □

3.1 Theorem 6: a constructive sufficiency proof

In this subsection, we always assume that the polytope $\mathcal{G}$ is robustly non-singular.

We denote by $\|\cdot\|$ the Euclidean norm $\|y\|^2 = y^\top y$ and, as done by Blanchini et al. (2015, 2016) in the continuous-time case, we consider the Lyapunov-like positive definite function

$$V(y) = \frac{1}{2} y^\top y.$$

Our goal is to find, for any y, the control $v = \Phi(y)$ such that

$$\max_{G\in\mathcal{G}} \|y + G\Phi(y)\| \le \lambda \|y\|, \qquad (12)$$

for some contraction factor $\lambda < 1$. This problem can be formulated as a min-max game: given y, find v such that

$$\mu^+ = \min_{v\in\mathbb{R}^m} \max_{G\in\mathcal{G}} \|y + Gv\| \qquad (13)$$

satisfies $\mu^+ < \lambda\|y\|$, with $\lambda < 1$. However, here we cannot rely on a saddle point result. Indeed, if we consider the reverse game, in which G "plays first",

$$\mu^- = \max_{G\in\mathcal{G}} \min_{v\in\mathbb{R}^m} \|y + Gv\|,$$

we have the strict inequality $\mu^- < \mu^+$. In fact, μ^+ is in general positive, while $\mu^- = 0$ (under robust non-singularity assumptions, once the maximizing G has been chosen, the minimiser can be chosen as $v = -G^{-1}y$).

The sufficiency statement of Theorem 6 can be proved constructively, by proposing a suitable control strategy. Consider the set

$$\mathcal{V}(\lambda, y) = \{v : \ \|y + Gv\| \le \lambda\|y\|, \ \text{for all } G \in \mathcal{G}\}. \qquad (14)$$

Lemma 9. The set $\mathcal{V}(\lambda, y)$ in (14) can be equivalently represented as

$$\mathcal{V}(\lambda, y) = \{v : \ \|y + G_i v\| \le \lambda\|y\|, \ i = 1, \ldots, r\}. \qquad (15)$$

Proof. If v is in the set (14), the inequality has to be satisfied for any G: then, also the inequalities in (15), which are a finite subset, must be satisfied. Conversely, assume that v is in the set (15). Then we can show that $\|y + Gv\| \le \lambda\|y\|$ for all $G \in \mathcal{G}$. In fact, in view of (6),

$$\|y + Gv\| = \left\| y + \sum_{i=1}^r \alpha_i \, G_i v \right\| = \left\| \sum_{i=1}^r \alpha_i (y + G_i v) \right\|$$
$$\le \sum_{i=1}^r \alpha_i \ \|y + G_i v\| \le \sum_{i=1}^r \alpha_i \, \lambda \, \|y\| = \lambda \, \|y\|,$$

because the norm is convex and the nonnegative coefficients α_i sum up to 1. □

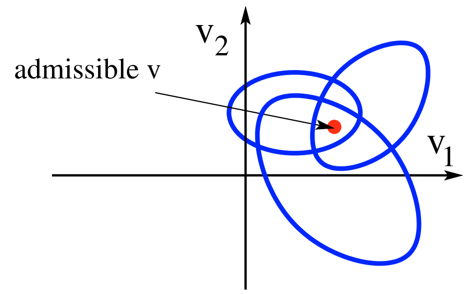

Fig. 1. $\mathcal{V}(\lambda, y)$ as the intersection of ellipsoids.

For any y, the set $\mathcal{V}(\lambda, y)$ in (15) is the intersection of ellipsoids (cf. Fig. 1) defined by $\|y + G_i v\|^2 \le \lambda^2\|y\|^2$ and centered in $v_i = G_i^{-1}y$. Hence, $G_i v_i = y$ and $\|G_i v_i + G_i v\|^2 \le \lambda^2\|y\|^2$.

The set $\mathcal{V}(\lambda, y)$ is such that

$$\mathcal{V}(\lambda_1, y) \subset \mathcal{V}(\lambda_2, y) \quad \text{if} \quad \lambda_1 < \lambda_2$$

and is always non-empty if $\lambda \ge 1$, since it includes the value $v = 0$. Clearly, however, we are interested in values $\lambda < 1$, and possibly in the smallest one, which we denote as

$$\lambda^*(y) = \min\{\lambda : \mathcal{V}(\lambda, y) \ne \emptyset\}. \qquad (16)$$

Lemma 10. The value $\lambda^*(y)$ in (16) corresponds to the set $\mathcal{V}(\lambda^*(y), y)$, which is the smallest non-empty set of the family $\mathcal{V}(\lambda, y)$ and includes a single point.

Proof. By definition, $\lambda = \lambda^*(y)$ is the smallest value for which $\mathcal{V}(\lambda, y)$ is not empty. Therefore, we just need to prove that $\mathcal{V}(\lambda^*(y), y)$ is a singleton. Being the intersection of a finite number of ellipsoids, the set $\mathcal{V}(\lambda, y)$ is strictly convex: for $v_1, v_2 \in \mathcal{V}(\lambda, y)$ and $v_1 \ne v_2$, all the points of the segment $\beta v_1 + (1-\beta)v_2$, with $0 < \beta < 1$, are in the interior of $\mathcal{V}(\lambda, y)$. Then, as long as there are distinct points in $\mathcal{V}(\lambda, y)$, an interior point $\tilde{v}$ exists, hence we can choose a smaller value $\lambda' < \lambda$ such that $\mathcal{V}(\lambda', y)$ includes $\tilde{v}$. □

The set $\mathcal{V}(\lambda, y)$ is the intersection of ellipsoids, each having a non-zero center $v_i = G_i^{-1}y$ as long as $y \ne 0$. In view of Lemma 10, the set corresponding to $\lambda^*(y)$ is a singleton, as shown in Fig. 2: $\mathcal{V}(\lambda^*(y), y) = v^*(y)$. Then $v^*(y)$ is on the boundary of some (say, of the first p) ellipsoids: the active constraints are

$$q_i(v) \doteq \|y + G_i v\|^2 = \lambda^*(y)^2\|y\|^2, \quad i = 1, \ldots, p. \qquad (17)$$

Lemma 11. Given $y \ne 0$, let $i = 1, \ldots, p$ index the constraints that are active at v^* and consider the corresponding gradients computed at v^*: $\nabla q_i(v^*)$, $i = 1, \ldots, p$, where $q_i(v)$ are the quadratic functions defined in (17). Then there exist $\gamma_i \ge 0$, $i = 1, \ldots, p$, with $\sum_{i=1}^p \gamma_i = 1$, such that

$$\sum_{i=1}^p \gamma_i \nabla q_i(v^*) = 0. \qquad (18)$$

The proof, reported in the appendix, is based on the following fact. Given y, our problem is equivalent to finding

$$\lambda^*(y)^2 = \min_{v\in\mathbb{R}^m} \left[\max_{i\in\{1,\ldots,r\}} \frac{\|y + G_i v\|^2}{\|y\|^2} \right] = \min_{v\in\mathbb{R}^m} \phi(v),$$

where ϕ is a convex function. Condition (18) is equivalent to $0 \in \partial\phi(v^*)$, where $\partial\phi(v)$ is the subdifferential of $\phi(v)$.

Since, for $\lambda \ge 1$, $\mathcal{V}(\lambda, y)$ is non-empty, then it must be that $\lambda^*(y) \le 1$. We just need to prove that, whenever $\mathcal{G}$ is robustly non-singular, $\lambda^*(y) < 1$.

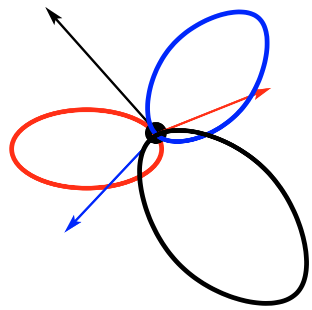

Fig. 2. The intersection of ellipsoids $\mathscr{V}(\lambda^*(y), y)$ is a singleton.

Lemma 12. Robust non-singularity of $\mathscr{G}$ implies $\lambda^*(y) < 1$ for all $y \neq 0$.

Proof. By contradiction, assume that, for some $y \neq 0$, $\lambda^*(y) = 1$. According to Lemma 10, this means that the set $\mathscr{V}(1, y)$ includes a single point: the origin $v = 0$. Consider the gradients of the quadratic functions $q_i(v)$ computed at $v = 0$: $\nabla q_i(0)$. In view of Lemma 11, there exists a null positive convex combination: $\sum_{i=1}^p \gamma_i \nabla q_i(0) = 0$. The gradient of $q_i(v) = \|y + G_i v\|^2$ is

$$\nabla q_i(v) = 2y^\top G_i + 2v^\top G_i^\top G_i.$$

Evaluating $\nabla q_i(v)$ at $v = 0$ and taking a combination with positive numbers γ_i summing up to 1 leads to

$$0 = \sum_{i=1}^p \gamma_i (2y^\top G_i) = 2y^\top \sum_{i=1}^p \gamma_i G_i. \quad (19)$$

Then matrix $G = \sum_{i=1}^p \gamma_i G_i$ is singular, but $G \in \mathscr{G}$: a contradiction. □

So far we have seen that, for every y, there exists a choice of v,

$$v = \Phi(y) \doteq \arg\min_{v \in \mathbb{R}^m} \max_{i \in \{1,\dots,r\}} \frac{\|y + G_i v\|}{\|y\|} = \lambda^*(y) < 1, \quad (20)$$

ensuring a contraction (since $\lambda^*(y) < 1$). This guarantees that the sequence $y_{k+1} = y_k + G(k)v_k$ generated by (10), hence also the original sequence $y_{k+1} = g(u_k + v_k)$ in (9), are decreasing in norm. However, this is not sufficient to assure convergence.

The last step to finally prove the sufficiency statement of Theorem 6 is given by the following lemma, whose proof relies on a Krasowskii-type argument.

Lemma 13. If the control law (20) is applied, the sequence y_k generated by (10) converges to zero: $y_k \to 0$.

Proof. The sequence is decreasing in norm, hence

$$\|y_k\| \to \mu$$

from above, where $\mu \geq 0$. If $\mu = 0$, the proof is over. Assume by contradiction that $\mu > 0$: the sequence y_k is bounded, since

$$\mu \leq \|y_k\| \leq \|y_0\|,$$

and, therefore, it must have an accumulation point $\bar{y}$. Then, there exists a subsequence y'_k that converges to $\bar{y}$, with $\|\bar{y}\| = \mu$. For point $\bar{y}$, there must exist $\bar{v} = \Phi(\bar{y})$ and $\bar{\lambda} < 1$ such that

$$\|\bar{y} + G\bar{v}\| \leq \bar{\lambda}\|\bar{y}\|.$$

Since the points y'_k of the subsequence get arbitrarily close to $\bar{y}$,

$$\|y'_k + G\bar{v}\| \leq \|y'_k - \bar{y} + \bar{y} + G\bar{v}\| \leq$$
$$\|y'_k - \bar{y}\| + \|\bar{y} + G\bar{v}\| \leq \|y'_k - \bar{y}\| + \bar{\lambda}\mu,$$

hence $\|y'_k + G\bar{v}\| < \mu$ for k large enough (because $y'_k \to \bar{y}$ and $\bar{\lambda} < 1$). This means that, at some point of the original sequence y_k, the control $\bar{v}$ produces a norm smaller than μ. Since the control $\Phi(y'_k)$ minimises the norm, we get to the inequality

$$\|y'_k + G\Phi(y'_k)\| \leq \|y'_k + G\bar{v}\| < \mu,$$

which is a contradiction. Then, $\mu = 0$ and $y_k \to 0$. □

Remark 14. **(Computing the Tuning Law).** Given the polytope $\mathscr{G}$ and the current value $y = y_k$, the computation of the tuning law $v_k = v = \Phi(y)$ boils down to the on-line solution of a convex minimisation problem with a linear functional ξ (which corresponds to λ^2) and quadratic constraints $\|y + G_i v\|^2 \leq \xi\|y\|^2$, $i = 1,\dots,r$. Once the optimal ξ^* has been found, $v = v^*(y)$ is the unique value such that $\|y + G_i v\|^2 \leq \xi^*\|y\|^2$ for all $i = 1,\dots,r$.

4. THE NON-SQUARE CASE

If there are more inputs than outputs, Theorem 6 needs to be reformulated as follows.

Theorem 15. Assume $p < m$. Given the discrete-time system (10), an algorithm $v = \Phi(y)$ that drives y_k to 0 for any sequence $G(k) \in \mathscr{G}$ exists if and only if every matrix in the polytope $\mathscr{G}$ has full row rank. The control strategy is

$$v = \Phi(y) \doteq \arg\min_{v \in \mathbb{R}^m} \max_{i \in \{1,\dots,r\}} \|y + G_i v\|. \quad (21)$$

The necessity proof still holds: in fact, if G does not have full row rank, it has a nontrivial left kernel.

However, the set $\mathscr{V}(\lambda, y)$ is no longer the intersection of ellipsoids: it is the intersection of “cylinders”, therefore the solution can be non-unique and Lemma 10 is not true.

The sufficiency proof still can be constructed relying on Lemma 11, which holds for $m > p$ as well, as reported in the appendix. Hence, proceeding along the lines of the previous section, we can show that, if $\lambda^*(y) = 1$, then we get condition (19), which implies that there is a matrix $G \in \mathscr{G}$ that does not have full row rank.

5. APPLICATION EXAMPLE

The proposed approach can be very useful for tackling several engineering problems that are traditionally solved via trial and error. In this section, we consider an example of application. A heat exchanger (cf. Fig. 3), whose purpose is cooling a fluid, needs to be designed by choosing the *fluid flow* q and the *cooler surface* S, in order to produce a *desired positive temperature drop* $\Delta T = T_1 - T_2 > 0$, given the inlet temperature T_1, and a *desired exchanged heat* h. The external temperature T_0 is assumed to be constant and considerably smaller than T_1 and T_2.

We assume that a software is available for computing ΔT and h corresponding to given values of S and q. Typically, a reliable software based on finite elements can require hours of computation. Trial and error can be carried out by gridding the surface-flow plane and looking for the proper point. Clearly, no continuous-time approach (Blanchini et al. 2015, 2016) can be considered to tackle this problem, since iteratively running a software is naturally a discrete sequence of events.

Our goal is to run the software a limited (small) number of times, guided by our algorithm.

Very roughly, the temperature drop $\Delta T = g_1(S,q)$ can be seen as an increasing function of the surface S and as a decreasing function of the flow q: symbolically, $g_1(+,-)$. The exchanged heat $h = g_2(S,q)$ can be seen as an increasing function of both the surface S and the flow q: symbolically, $g_2(+,+)$.

We denote the partial derivatives (up to the sign) with Greek letters, $\alpha = \partial g_1/\partial S$, $\beta = -\partial g_1/\partial q$, $\gamma = \partial g_2/\partial S$ and $\delta = \partial g_2/\partial q$, and we assume that these positive quantities are bounded as $\alpha \in [\alpha^-,\alpha^+]$, $\beta \in [\beta^-,\beta^+]$, $\gamma \in [\gamma^-,\gamma^+]$ and $\delta \in [\delta^-,\delta^+]$. The Jacobian turns out to be inside the polytope having vertices

$$J \in \begin{bmatrix} \alpha^{\pm} & -\beta^{\pm} \\ \gamma^{\pm} & \delta^{\pm} \end{bmatrix},$$

which is robustly non-singular.

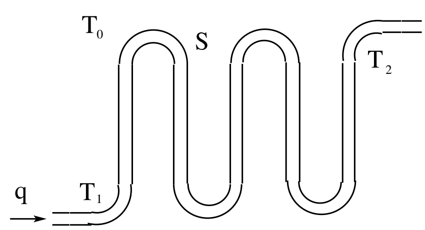

Fig. 3. The heat exchanger problem.

For simulation purposes, we have adopted the classical LMTD model (Log-Mean Temperature Difference model, Kakaç & Liu 2002) for describing the steady-state temperature and exchanged heat balances

$$y_1 = \Delta T - \Delta T_{ref} = (T_1 - T_0)\cdot\left(1 - e^{-\frac{ku_1}{u_2 c\rho}}\right) - \Delta T_{ref}$$

$$y_2 = h - h_{ref} = u_2 c\rho(T_1 - T_0)\cdot\left(1 - e^{-\frac{ku_1}{u_2 c\rho}}\right) - h_{ref}$$

where $u_1 = S$ is the surface, $u_2 = q$ is the flow, $k = 500\mathrm{W/m^2/K}$ is the heat transfer coefficient, $\rho = 10^3\mathrm{Kg/m^3}$ is the fluid's density, $c = 4.186\cdot 10^3\mathrm{J/Kg/K}$ is the fluid's specific heat, $T_1 = 353\mathrm{K}$ is the inlet temperature and $T_0 = 288\mathrm{K}$ is the external temperature. Given the above model (which, of course, is *not known by the algorithm*), we have required a temperature drop $\Delta T_{ref} = 29\mathrm{K}$ and an exchanged heat $h_{ref} = 0.734\mathrm{MW}$. The adopted bounds on the derivatives are

$$\alpha^- = \beta^- = \gamma^- = \delta^- = 0.004$$

and

$$\alpha^+ = 10,\ \ \beta^+ = 20,\ \ \gamma^+ = \delta^+ = 0.02.$$

We have implemented the control strategy (20), solving the sequence of optimisation problems by means of CVX (Grant & Boyd 2014). The transient behaviour of both temperature drop and heat is reported in Fig. 4, while the input evolution (surface and flow) is reported in Fig. 5.

The algorithm requires less than 20 steps to converge to the desired output values.

Finally, it is worth pointing out that, for numerical reasons, it is important to render the variables as similar as possible in magnitude, hence adopting suitably scaled values is highly recommended. In the reported simulations, y_1 was scaled by a factor of 10^{-2} and y_2 by a factor of 10^{-6}, thus obtaining a magnitude of roughly 10^{-1} for both variables (note that the bounds on the derivatives are referred to the scaled variables).

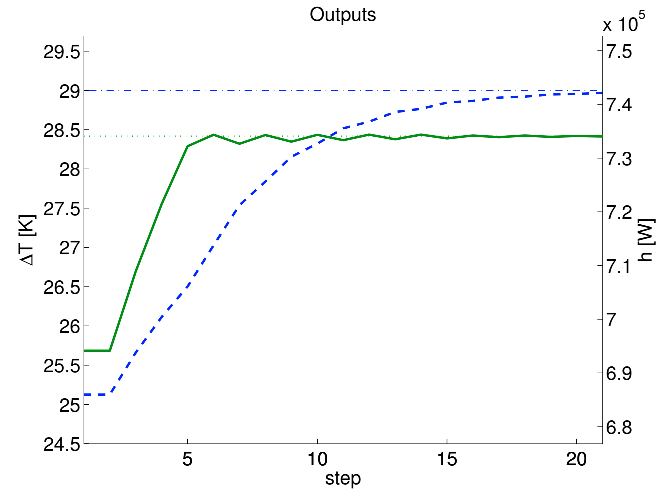

Fig. 4. Evolution of the temperature drop [K] (blue) and heat [W] (green).

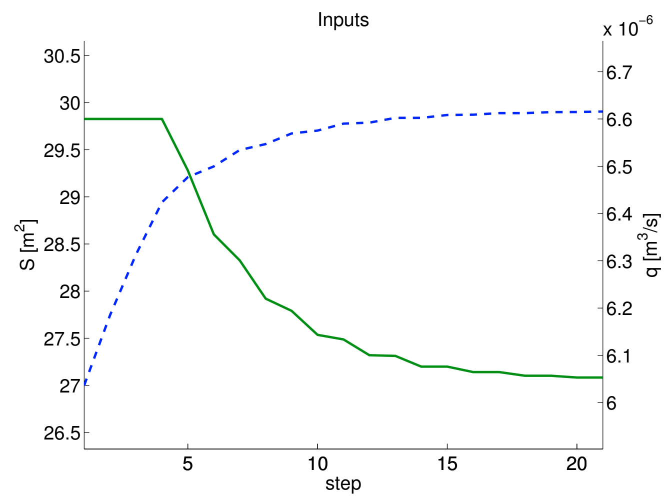

Fig. 5. Evolution of surface [m^2] (blue) and flow [m^3/s] (green).

6. CONCLUDING DISCUSSION

We have considered the problem of *discrete-time plant tuning*: driving the output of a static nonlinear plant to a specific value by selecting the proper input *without the knowledge of the model*, and having at our disposal just a discrete sequence of input trials (as it happens when we iteratively run a software, with different input data at each iteration, so as to get desired output values). Under proper non-singularity assumptions, the problem has been solved by Blanchini et al. (2015, 2016) in the continuous-time case, namely, when the input value can be continuously changed in time. In discrete-time, the saddle point theorem on which the continuous-time scheme relies does no longer apply. However, adopting a different technique, we have shown that the problem can be solved as well, achieving the same results under the same non-singularity assumptions. The technique has been illustrated by proposing an application example, which shows that a software-based design problem can be efficiently handled by our scheme, ensuring convergence to the sought input values after few iterations.

Extensions of the proposed method include the case in which the unknown function is not differentiable, but is Lipschitz only (this is the case, for instance, of piecewise smooth and piecewise linear plants). We also believe that the proposed technique for the discrete-time problem can be extended to implicitly-defined functions, as done for the continuous-time case (Blanchini et al. 2016).

REFERENCES

H.-S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization", *IEEE Trans. on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, pp. 1099-1121, 2007.

K. J. Åström, "Theory and applications of adaptive control — A survey", *Automatica*, vol. 19, no. 5, pp. 471–486, 1983.

H. G. Beyer and B. Sendhoff, "Robust optimization — A comprehensive survey", *Comput. Methods in Appl. Mech. Eng.*, vol. 196, no. 33–34, pp. 3190–3218, 2007.

F. Blanchini, "The gain scheduling and the robust state feedback stabilization problems", *IEEE Trans. Autom. Control*, vol. 45, no. 11, pp. 2061–2070, 2000.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, "Plant tuning: a robust Lyapunov approach", *Proc. 54th IEEE Conf. on Decision and Control*, Osaka, December 15-18, 2015.

F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, "Model-free plant tuning", to appear in *IEEE Trans. Autom. Control* http://ieeexplore.ieee.org/document/7586127/.

F. Blanchini and S. Miani, *Set-theoretic methods in control*. Systems & Control: Foundations & Applications. Second edition. Birkhäuser, Basel, 2015.

F. Blanchini and R. Pesenti, "Min-max control of uncertain multi-inventory systems with multiplicative uncertainties", *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 955–959, 2001.

D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, June 2006.

C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations", *Math. Comp.*, vol. 19, pp. 577–539, 1965.

A. Fradkov, "A scheme of speed gradient and its application in problems of adaptive control", *Autom. and Rem. Contr.*, vol. 40, no. 9, pp. 1333–1342, 1980.

M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, v. 2.1*, http://cvxr.com/cvx, 2014.

L. Gurvits and A. Olshevsky, "On the NP-Hardness of Checking Matrix Polytope Stability and Continuous-Time Switching Stability", in *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 337-341, Feb. 2009.

S. Gutman and G. Leitmann, "Stabilizing control for linear systems with bounded parameter and input uncertainty", Optimization Techniques Modeling and Optimization in the Service of Man Part 2, Lecture Notes in Computer Science, pp. 729-755, 1976.

S. Gutman, "Uncertain dynamical systems – a Lyapunov min-max approach", *IEEE Trans. Autom. Control*, vol. 24, no. 3, pp. 437–443, 1979.

H. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, NJ, 1996.

S. Z. Khong, D. Nešić, Y. Tan, C. Manzie, "Unified frameworks for sampled-data extremum seeking control: global optimisation and multi-unit systems", *Automatica*, vol. 49, no. 9, pp. 2720–2733, 2013.

D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons Inc., New York, 1969.

A. M. Meılakhs, "Design of stable control systems subject to parametric perturbation", *Autom. Rem. Control*, vol. 39, no. 10, pp. 1409–1418, 1979.

D. Nešić, A. Mohammadi, C. Manzie, "A framework for extremum seeking control of systems with parameter uncertainties", *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 435–448, 2013.

B. Ross Barmish, *New Tools for Robustness of Linear Systems*. MacMillan, 1994.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, USA, 1970.

Y. Tan, D. Nešić and I. Mareels, "On non-local stability properties of extremum seeking control", *Automatica*, vol. 42, no. 6, pp. 889–903, 2006.

S. Kakaç and H. Liu, *Heat Exchangers: Selection, Rating, and Thermal Design*, 2nd edition, CRC Press, Boca Raton, Florida, USA, 2002.

Appendix A. PROOF OF LEMMA 11

We prove the statement in the general case $m \geq p$.

We have observed that, for a fixed $y \neq 0$, our problem is equivalent to finding

$$\lambda^*(y)^2 = \min_{v\in\mathbb{R}^m} \left[\max_{i\in\{1,\dots,r\}} \frac{\|y+G_i v\|^2}{\|y\|^2} \right] = \min_{v\in\mathbb{R}^m} \phi(v).$$

In the general case, the minimum is not unique. So, let $\mathscr{V}^*$ be the set of all minimisers of the convex problem. We can prove that, for any $v^* \in \mathscr{V}^*$,

$$\sum_{i=1}^{p} \nabla q_i(v^*)\gamma_i = 0, \quad \sum_{i=1}^{p} \gamma_i = 1, \quad \gamma_i \geq 0, \qquad \text{(A.1)}$$

where $i = 1, 2, \dots, p$ index all active constraints.

Function ϕ is convex; its subdifferential at $\bar{v}$ (Rockafellar 1970) is the set

$$\partial\phi(\bar{v}) \doteq \{z \in \mathbb{R}^m : \, z^\top(v-\bar{v}) \leq \phi(v) - \phi(\bar{v}), \, \forall v\}$$

and $\phi(v)$ attains a minimum at v^* if and only if (Rockafellar 1970)

$$0 \in \partial\phi(v^*).$$

Define functions $\psi_i(v) \doteq \frac{\|y+G_i v\|^2}{\|y\|^2} = \frac{q_i(v)}{\|y\|^2}$ and denote by $\mathscr{A}(v)$ the set that indexes "active functions" (those providing the maximum):

$$\mathscr{A}(v) = \{i : \psi_i(v) = \phi(v)\}.$$

The subdifferential of the max function (see for instance Blanchini & Miani 2015) is a convex cone formed by all the positive convex combinations of the gradients of the active functions, $\nabla\psi_i(v)$ with $i \in \mathscr{A}(v)$. At the optimum, all active functions are such that

$$\psi_i(v) = \lambda^*(y)^2,$$

therefore $\mathscr{A}(v)$ indexes indeed active constraints, cf. (17), and $\mathscr{A}(v) = \{1, \dots, p\}$.

The optimality condition $0 \in \partial\phi(v^*)$ is then equivalent to

$$0 = \sum_{i=1}^{p} \nabla\psi_i(v^*)\gamma_i = \frac{1}{\|y\|^2} \sum_{i=1}^{p} \nabla q_i(v^*)\gamma_i,$$

for some $\gamma_i \geq 0$ such that $\sum_{i=1}^{p} \gamma_i = 1$, which entails condition (A.1).