http://dx.doi.org/10.1049/iet-cta.2014.0187

Data-based two-degree-of-freedom iterative control approach to constrained non-linear systems

Mircea-Bogdan Radac [a], Radu-Emil Precup [a *]

[a] *Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania*

Abstract

This paper proposes a data-based model-free approach to reference trajectory tracking in two-degree-of-freedom (2-DOF) nonlinear control system structures. This model-free control approach tunes both the feedback controller parameters and the reference input sequence accounting for control saturation and control rate constraints. The controller is iteratively tuned in a nonlinear framework that employs a gradient descent search approach. The model-free gradient estimates are obtained by a perturbation-based approach. The reference input tuning is carried out in a linear framework using an Iterative Learning Control-based approach, and it also includes a model-free gradient search algorithm where the gradient estimates are obtained by a similar perturbation-based approach. The number of real-world experiments is significantly reduced by the use of simulated models identified as neural networks. A digitally simulated case study concerning the angular position control of a nonlinear aerodynamic twin-rotor system shows that our approach can effectively improve the control system performance.

Keywords: Control signal rate constraints; Control signal saturation constraints; Iterative control; Model-free control approach; Neural networks; Stochastic search algorithm

[*] Corresponding author. tel.: +40 256 40 3229; fax: +40 256 40 3214.
E-mail address: radu.precup@aut.upt.ro (R.-E. Precup).

http://dx.doi.org/10.1049/iet-cta.2014.0187

1. Introduction

Data-based control design targets the control system (CS) performance improvement using optimization approaches where no a priori model information on the process is available or little such information is used. The performance specifications in complex industrial process applications are translated into easily interpretable performance indices that are usually specified in the time domain (for example, the rise time, the overshoot, the settling time), and they are aggregated in general integral-type or sum-type objective functions. The minimization of these objective functions in the framework of constrained optimization problems can fulfil different objectives such as reference trajectory tracking (including model reference tracking), control signal (c.s.) penalty, disturbance rejection, etc.

The reference trajectory tracking problem can be formulated as a dual data-based iterative optimization problem (OP) with respect to both the feedback controller parameters and the reference input. The main data-based techniques that carry out the iterative experiment-based update of controller parameters are Iterative Feedback Tuning (IFT) [1], Correlation-based Tuning [2], Frequency Domain Tuning [3], Iterative Regression Tuning [4], and Simultaneous Perturbation Stochastic Approximation [5], [6]. Other model-free control approaches are data-driven predictive control [7], [8], Model-free Control [9], data-based or data-driven Model-free Adaptive Control [10], [11], unfalsified control [12], and adaptive online IFT [13]. The most popular non-iterative technique is Virtual Reference Feedback Tuning (VRFT) [14], [15]. These techniques use various approaches to ensure model-free controller tuning. However, the tuning to achieve reference trajectory tracking does not guarantee robust stability or robust performance. Some recent data-based control approaches ensure robust stability/performance while still keeping the model-free property; these approaches try to avoid the direct process identification and to infer the results from data or from easy-to-obtain non-parametric CS models such as the frequency response functions [3]. The data-based control can be integrated with other data-based approaches for modelling, process monitoring and fault diagnosis [16].

On the other hand, as shown in our recent papers [17] and [18], the reference trajectory tracking can also be considered as a reference input design over an initial CS a priori tuned controllers in order to solve stability and disturbance rejection issues. Therefore, the reference trajectory tracking is defined as an open-loop optimal control problem. An Iterative Learning Control (ILC) framework [19], [20] can be used with this respect. Such approaches to ILC-based solving of optimal control problems are formulated in [20], and the stochastic approximation is treated in [21]. The analysis of the current literature highlights that the reference tracking belongs to the optimization issues in data-based control which are thoroughly discussed in [22]. The affine constraints are handled in [23] by the transformation of ILC problems with quadratic o.f.s into convex quadratic programs. The system's impulse response is estimated in [24] using input/output measurements and next used in a norm-optimal ILC structure that accounts for linear inequality actuator constraints. A learning approach for the parameters of motion primitives to achieve flips for quadrocopters is given in [25] using simple approximate models. Reinforcement learning formulations for policy search using approximate models and signed derivatives are presented in [26].

This paper offers a novel two-degree-of-freedom (2-DOF) iterative data-based model-free control approach to reference trajectory tracking problems. The optimal tuning of the controller parameters ensures iterative control as it uses an IFT approach whereas the optimal tuning of the reference input sequence is tackled using ILC. Both tunings address the c.s. saturation and c.s. rate constraints and they can be viewed in a general iteratively solved supervised learning approach.

This paper proposes the following new contributions with respect to the previous approaches given in [18] and [27], and with respect to the state-of-the-art on iterative data-based learning, with focus on as few as possible learning iterations for significant performance improvement using as little as possible information on the process:

- A new 2-DOF iterative data-based model-free solution to the reference trajectory tracking problem is offered, in which both the feedback controller and the reference input are tuned.
- A mechanism to deal with c.s. saturation and c.s. rate constraints using a quadratic penalty function approach is proposed.
- The reduction of the number of real-world experiments in the computation of the gradient of the objective function (o.f.) is achieved. This results from a neural network (NN) simulation-based approach where the models identified as NNs are valid only in the vicinity of the nominal trajectories at the current iteration.
- A convincing case study on a nonlinear aerodynamic twin-rotor system to illustrate the effectiveness of our approach is provided.

Our iterative approach is attractive and advantageous with respect to the state-of-the-art because:

- It significantly improves the CS performance and it also compensates for the poor process modelling (including uncertainties), identification and complexity.
- The computations are carried out offline. Therefore, they do not require excessive real-time processing demands.

The paper is structured as follows. The reference trajectory tracking problem is formulated in Section 2 as an OP. The controller tuning problem and the reference input tuning problems are presented in Subsections 2.1 and 2.2. The digitally simulated case study given in Section 3 illustrates the application of our approach, and discussions are included. The conclusions are highlighted in Section 4.

2. Problem formulation

The Single Input-Single Output (SISO) discrete-time CS is described by the nonlinear process and controller equations:

$$\begin{aligned} y(r,\boldsymbol{\rho},k) &= P(y(k-1),...,y(k-n_{y1}),u(k-1),...,u(k-n_{u1}))+v(k), \\ u(r,\boldsymbol{\rho},k) &= C(\boldsymbol{\rho},u(k-1),...,u(k-n_{u2}),y(k),...,y(k-n_{y2}),r(k),...,r(k-n_r)), \end{aligned} \tag{1}$$

where k is the discrete time argument, $y(r,\boldsymbol{\rho},k)$ is the process output sequence, $r(k)$ is the reference input sequence, $v(k)$ is the zero-mean stationary and bounded stochastic disturbance input and can model a large class of load and noise measurement disturbances, $\boldsymbol{\rho}$, $\boldsymbol{\rho} \in \Re^{n_\rho}$, is the parameter vector of the controller, and $\Re$ is the set of real numbers. The nonlinear functions P and C make the model (1) belong to the class of nonlinear autoregressive exogenous (NARX) models treated in [28].

The assumptions related to (1) are

- The closed-loop CS is stable.
- P and C are smooth functions of their arguments.
- The nominal trajectory of the CS is $\{r_n(k), u_n(k), y_n(k)\}$, $k=0...N$, where N is the experiment length.

A typical objective in iterative CS performance improvement is to solve an OP defined as a reference trajectory tracking problem, starting with the initial controller parameters $\boldsymbol{\rho}_0$ and with the initial reference input $r_0(k)$:

$$\begin{aligned} r^*, \boldsymbol{\rho}^* &= \arg \min_{r,\boldsymbol{\rho} \in D_S} J(r,\boldsymbol{\rho}), \\ J(r,\boldsymbol{\rho}) &= \frac{1}{2N} E\{\sum_{k=0}^{N} [(y(r,\boldsymbol{\rho},k)-y^d(k))^2 + \lambda u^2(r,\boldsymbol{\rho},k)]\}, \end{aligned} \tag{2}$$

subject to system dynamics (1) and to operational constraints, where D_S is the stability domain of those parameter vectors $\boldsymbol{\rho}$ that ensure a stable CS [29], [30], and several stability conditions can be involved [31]–[34]. The constraints are usually formulated as inequalities imposed to $u(k)$ and $y(k)$, and to their rates with respect to time, $\Delta u(k) = u(k) - u(k-1)$ and $\Delta y(k) = y(k) - y(k-1)$, and they depend on specific applications [35]–[37]. The expression of J in (2) targets the trajectory tracking of the desired system output y^d, the c.s. is penalized by the weighting parameter $\lambda \geq 0$, and the expectation $E\{...\}$ is taken with respect to v.

Equations (1) and (2) show that the o.f. is influenced by both the controller parameters and the reference input. Thus, our new approach focuses on the combined tuning of controller parameters and of reference input sequence to achieve the same control objective, namely the reference trajectory tracking specified in the o.f. (2). Our approach considers successive controller and reference input tunings using a closed-loop parameterized control policy and an open-loop unparameterized one.

2.1. Controller tuning

The reference input r is considered to be fixed within one experiment trial. The usual approach to solve the OP (2) in the unconstrained case is to employ the recursive stochastic search algorithm

$$\boldsymbol{\rho}_{j+1} = \boldsymbol{\rho}_j - \gamma_j \mathbf{H}_j^{-1} est\left\{ \left.\frac{\partial J}{\partial \boldsymbol{\rho}}\right|_{\boldsymbol{\rho}=\boldsymbol{\rho}_j} \right\}, \tag{3}$$

with the search information provided by the estimate of the gradient of the o.f. J with respect to the controller parameters and using, for example, second-order information as a Gauss-Newton approximation of the Hessian $\mathbf{H}_j$ of the o.f. The subscript j, $j \in \mathbf{Z}$, $j \geq 0$, indicates the current iteration number, and $\gamma_j > 0$ is the step size [1].

The stochastic convergence of IFT algorithms is treated in [1]. Two stochastic convergence conditions are imposed, namely the estimated o.f. gradient is unbiased, and the step size sequence $\{\gamma_j\}_{j \geq 0}$ converges to zero. The second condition is fulfilled for the choice of γ_j

$$\sum_{j=0}^{\infty} \gamma_j = \infty, \ \sum_{j=0}^{\infty} \gamma_j^2 < \infty, \ \gamma_j > 0 \ \forall \, j \geq 0 \cdot \tag{4}$$

The main feature of IFT [1] provides gradient information from special experiments conducted on the closed-loop CS. These experiments avoid the process model, and they also require special operating regimes that are different from the nominal ones. The experiments generate the gradients of y and u with respect to the controller parameters, namely $\partial y / \partial \boldsymbol{\rho}$ and $\partial u / \partial \boldsymbol{\rho}$, which are next used to compute both the gradient of J and $\mathbf{H}_j$. Although the linearity is assumed, the nonlinear-based procedure is also feasible [38] because the gradients can be estimated not by finite difference approximations for modifications of $\boldsymbol{\rho}$, but by modified reference trajectories for small changes in the vicinity of the nominal trajectories, $\delta r(k) = r(k) - r_n(k)$, $\delta u(k) = u(k) - u_n(k)$ and $\delta y(k) = y(k) - y_n(k)$. The procedure used in [28] is based on the identification of linear time-varying models by a least squares criterion with forgetting factor which is different from our NN-based approach.

The NNs will be used here as convenient universal approximators (with prescribed accuracy) to provide the gradient information needed in the search algorithm. With this regard, the nonlinear map from r to y and the nonlinear map from r to u are identified using data collected in the normal experiment in which the o.f. is evaluated. Let these two maps be

$$\begin{aligned} y(k) &= M_{ry}(y(k-1),...y(k-n_y), r(k-1),...r(k-n_{ry})), \\ u(k) &= M_{ru}(u(k-1),...u(k-n_u), r(k-1),...r(k-n_{ru})). \end{aligned} \quad (5)$$

The variables $\partial y / \partial \rho_h$ and $\partial u / \partial \rho_h$ are next estimated by finite difference approximations

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial \rho_h} &= \frac{\bar{y}(k, r_n + \mu_h \delta r_h) - \bar{y}(k, r_n)}{\mu_h \delta \rho_h}, \\ \frac{\partial \hat{u}(k)}{\partial \rho_h} &= \frac{\bar{u}(k, r_n + \mu_h \delta r_h) - \bar{u}(k, r_n)}{\mu_h \delta \rho_h}, h = 1...n_\rho, k = 0...N, \end{aligned} \quad (6)$$

where $\delta \rho_h = 1$ is considered, and the numerators are equivalent to carrying out two simulations, i.e., one with nominal controller parameter vector $\boldsymbol{\rho}$ and another one with h^{th} controller parameter varied with the term $\mu_h \delta \rho_h$. The scalars μ_h are chosen to account for only small changes around the nominal reference input trajectory $\{r_n(k)\}$ where the analysis holds. The variables $\bar{y}$ and $\bar{u}$ are obtained by filtering the nominal and the perturbed reference trajectories through the nonlinear maps M_{ry} and M_{ru}, respectively.

The advantages of our approach are:

- It can be applied to linear and nonlinear systems, and avoids the controller parameters perturbation-based approach for gradient estimation, hence the iterative controller parameters update is carried out when a descent direction is computed.
- By perturbing only the reference trajectory at each iteration rather than perturbing the controller parameters, the closed-loop stable operation of the CS is preserved in the vicinity of the current iteration trajectory.

http://dx.doi.org/10.1049/iet-cta.2014.0187

- Our approach avoids direct process knowledge because it uses simulated trajectories in terms of closed-loop CS models. Simple NN architectures can be trained because these models usually exhibit low order behaviours. Moreover, these models are obtained in the vicinity of the nominal trajectories and are not valid in a wide operating range and we are not concerned with experiment design for identification purposes.

The numerical differentiation issues in noisy environments are mitigated because the obtained trajectories are not affected by the noisy data involved in NN training. A double approximation involved by the linearization around the nominal trajectories and the NN-based approach is employed. The approach is efficient for small approximation errors.

The OP that ensures the reference trajectory tracking with c.s. constraints and with c.s. rate constraints is

$$\begin{aligned}
&\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho}\in D_S} J(\boldsymbol{\rho}),\ J(\boldsymbol{\rho}) = \frac{1}{2N}\sum_{k=1}^{N}[r(k) - y(k,\boldsymbol{\rho})]^2,\\
&\text{subject to } u_{\min}(k) \le u(k,\boldsymbol{\rho}) \le u_{\max}(k),\\
&\Delta u_{\min}(k) \le \Delta u(k,\boldsymbol{\rho}) \le \Delta u_{\max}(k),\ k = 1...N.
\end{aligned} \tag{7}$$

The constrained OP is transformed into an unconstrained OP using penalty functions. We propose the following augmented o.f. which accounts for inequality constraints on the c.s. saturation and on the c.s. rate:

$$\begin{aligned}
&\widetilde{J}_{p_j}(\boldsymbol{\rho}) = J(\boldsymbol{\rho}) + p_j\phi(\boldsymbol{\rho}),\\
&\phi(\boldsymbol{\rho}) = \frac{1}{2}\sum_{m=1}^{c}\{[\max\{0, -q_m(\boldsymbol{\rho})\}]^2\},\\
&\mathbf{q}(\boldsymbol{\rho}) = [u_{\max}(1) - u(1,\boldsymbol{\rho}) \quad ... \quad u_{\max}(N) - u(N,\boldsymbol{\rho}) \quad u(1,\boldsymbol{\rho}) - u_{\min}(1) \quad ... \quad u(N,\boldsymbol{\rho}) - u_{\min}(N)\\
&\Delta u_{\max}(1) - \Delta u(1,\boldsymbol{\rho}) \quad ... \quad \Delta u_{\max}(N) - \Delta u(N,\boldsymbol{\rho}) \quad \Delta u(1,\boldsymbol{\rho}) - \Delta u_{\min}(1) \quad ... \quad \Delta u(N,\boldsymbol{\rho}) - \Delta u_{\min}(N)]^T \in \Re^{4N},
\end{aligned} \tag{8}$$

where the positive and strictly increasing sequence of penalty parameters $\{p_j\}_{j\ge 0}$, $p_j \to \infty$, guarantees that the minimum of the sequence of augmented o.f.s $\{\widetilde{J}_{p_j}(\boldsymbol{\rho})\}_{j\ge 0}$ will converge to the solution to the constrained OP (7), and m, $m = 1...c$, is the constraint index, $q_m(\boldsymbol{\rho}) > 0$ is the m^{th} constraint. The OP that minimizes $\widetilde{J}_{p_j}(\boldsymbol{\rho})$ in (8) is solved using a stochastic approximation algorithm which uses the experimentally obtained gradient of $\widetilde{J}_{p_j}(\boldsymbol{\rho})$.

The quadratic penalty function $\phi(\boldsymbol{\rho})$ is defined in (8) on the basis of the maximum function which is non-differentiable only at zero. Given that $\phi(\boldsymbol{\rho})$ is Lipschitz and non-differentiable at a set of points of zero Lebesgue measure, the algorithm visits the zero-measure set with probability zero when a normal distribution for the noise is assumed [39]. Therefore, using

$$\frac{\partial [\max\{0, -q_m(\boldsymbol{\rho})\}]^2}{\partial \rho_h} = -2\max\{0, -q_m(\boldsymbol{\rho})\}\frac{\partial q_m(\boldsymbol{\rho})}{\partial \rho_h}, \tag{9}$$

the expression of the gradient of $\tilde{J}_{p_j}(\boldsymbol{\rho})$ at the current iteration j with respect to the parameter ρ_h is

$$\frac{\partial \tilde{J}_{p_j}(\boldsymbol{\rho})}{\partial \rho_h} = \frac{\partial J(\boldsymbol{\rho})}{\partial \rho_h} - p_j \sum_{m=1}^{c} \{\max\{0, -q_m(\boldsymbol{\rho})\}\frac{\partial q_m(\boldsymbol{\rho})}{\partial \rho_h}\}\cdot \tag{10}$$

The first term in (10) corresponding to the gradient of the original o.f. requires knowing the gradient $\partial y(k)/\partial \boldsymbol{\rho}$, and the second term in (10) requires the gradients of $u(k)$ and $\Delta u(k)$ with respect to $\boldsymbol{\rho}$. These variables are estimated using (6). The derivative of the c.s. rate with respect to the parameter vector $\boldsymbol{\rho}$ is estimated using the finite differences approximation approach for the sampling period δt:

$$\frac{\partial \Delta \hat{u}(k)}{\partial \rho_h} = \frac{1}{\delta t}[\frac{\partial \hat{u}(k)}{\partial \rho_h} - \frac{\partial \hat{u}(k-1)}{\partial \rho_h}],\ h = 1 \ldots n_\rho,\ k = 1 \ldots N \cdot \tag{11}$$

2.2. Reference input tuning

The controller parameters are considered to be fixed, and the reference input sequence is a vector variable in the OP (2). In addition, a linear approximation of the nonlinear model (1) is considered, and let the CS (1) be described by the discrete-time Linear Time-Invariant SISO model:

$$y(\boldsymbol{\rho}, r, k) = T(\boldsymbol{\rho}, q^{-1})r(k) + S(\boldsymbol{\rho}, q^{-1})v(k), \tag{12}$$

where the input and output variables are defined as in (1), $S(\boldsymbol{\rho}, q^{-1})$ is the sensitivity function, $T(\boldsymbol{\rho}, q^{-1})$ is the complementary sensitivity function

$$\begin{aligned} &S(\boldsymbol{\rho}, q^{-1}) = 1/[1 + P(q^{-1})C(\boldsymbol{\rho}, q^{-1})], \\ &T(\boldsymbol{\rho}, q^{-1}) = 1 - S(\boldsymbol{\rho}, q^{-1}), \end{aligned} \tag{13}$$

$P(q^{-1})$ is the process transfer function (t.f.), $C(\boldsymbol{\rho}, q^{-1})$ is the controller t.f. parameterized by the parameter vector $\boldsymbol{\rho}$ that contains the tuning parameters of the controller, and q^{-1} is the one step delay operator. The parameter vector $\boldsymbol{\rho}$ will be omitted as follows in some equations for the sake of simplicity.

For a relative degree n of the closed-loop CS t.f. $T(q^{-1})$, the lifted form representation for an N samples experiment length in the deterministic case is

$$\mathbf{Y} = \mathbf{T}\,\mathbf{R} + \mathbf{Y}_0, \tag{14}$$

with the matrices

http://dx.doi.org/10.1049/iet-cta.2014.0187

$$\begin{aligned}
&\mathbf{Y}=[y(n) \quad y(n+1) \quad \dots \quad y(N-1)]^T, \\
&\mathbf{R}=[r(0) \quad r(1) \quad \dots \quad r(N-n-1)]^T, \\
&\mathbf{Y}_0=[y_{10} \quad y_{20} \quad \cdots \quad y_{(N-n)0}]^T, \\
&\mathbf{T}=\begin{bmatrix} t_1 & 0 & \dots & 0 \\ t_2 & t_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ t_{N-n} & t_{N-n-1} & \dots & t_1 \end{bmatrix},
\end{aligned} \tag{15}$$

R is the reference input vector which contains the reference input sequence over the time interval $0 \le k \le N-n-1$, **Y** is the process output vector, t_i is i^{th} impulse response coefficient of $T(q^{-1})$, **T** is a lower-triangular Toeplitz matrix, $\mathbf{Y}_0$ is the free response of the CS due to nonzero initial conditions and trial-repetitive disturbances, and the superscript T indicates matrix transposition. Zero initial conditions are assumed without loss of generality, and the tracking error vector **E** is $\mathbf{E}=\mathbf{Y}-\mathbf{Y}^d=\mathbf{T}\,\mathbf{R}-\mathbf{Y}^d$, where $\mathbf{Y}^d$ is the reference trajectory vector generated from $y^d(k)$. Knowledge on **T** would provide the optimal solution which makes the tracking error zero, i.e., $\mathbf{R}=\mathbf{T}^{-1}\mathbf{Y}^d$. However, **T** can be ill-conditioned and it is always subjected to measurement errors; therefore $\mathbf{T}^{-1}$ cannot be used. A solution to the iterative estimation of **T** in an ILC framework is given in [24]. The control objective is to minimize the expected normalized norm of the tracking error:

$$\begin{aligned}
&\mathbf{R}^*=\arg\min_{\mathbf{R}} J(\mathbf{R})=E\{\frac{1}{N}(\mathbf{T}\,\mathbf{R}\underbrace{-\mathbf{Y}^d}_{\mathbf{M}})^T(\mathbf{T}\,\mathbf{R}-\mathbf{Y}^d)\}=E\{\frac{1}{N}(\mathbf{R}^T\,\mathbf{Q}\,\mathbf{R}+\mathbf{q}\,\mathbf{R}+\alpha)\} \\
&\text{subject to system dynamics (1)} \\
&\text{and to some operational constraints,}
\end{aligned} \tag{16}$$

where $\mathbf{Q}=\mathbf{T}^T\mathbf{T}$ is a positive semi-definite matrix, $\mathbf{q}=2\,\mathbf{M}^T\,\mathbf{T}$, and $\alpha=\mathbf{M}^T\mathbf{M}$. A gradient descent approach to iteratively solve (16) is

$$\mathbf{R}_{j+1}=\mathbf{R}_j-\gamma_j\tilde{\mathbf{H}}^{-1}est\{\left.\frac{\partial J}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}\}, \tag{17}$$

where j is the iteration or trial index, $est\{\left.\frac{\partial J}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}\}$ is the estimate of the gradient of the o.f. with respect to the reference input vector samples, $\tilde{\mathbf{H}}^{-1}$ is a Gauss-Newton approximation of the Hessian of the o.f., typically given by a Broyden-Fletcher-Goldfarb-Shanno (BFGS) update, and γ_j is the step size. The stochastic convergence of ILC algorithms is treated in [21], and it requires the same properties of γ_j as in (4).

http://dx.doi.org/10.1049/iet-cta.2014.0187

The o.f. in (16) is quadratic with respect to **R**, and the gradient of the o.f. in the deterministic case at each iteration *j* is

$$\left.\frac{\partial J}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j} = \frac{2}{N}\mathbf{T}^T\mathbf{E}_j \cdot \tag{18}$$

Equation (18) suggests that the gradient information can be obtained either by an experimentally measured **T** or by using a special gradient experiment (g.e.) at each iteration. The second solution is preferred in the model-free approach.

We propose an experimental approach to extract the gradient information from the experiments conducted in the vicinity of the nominal trajectories. This is a perturbation-based approach inspired by [38], and it is a modified version of the algorithm given in [18]. The algorithm that gives $\left.\frac{\partial J}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}$ is formulated as follows:

Step A. Record the tracking error at the current iteration in the vector $\mathbf{E}_j$.

Step B. Define the reversed vector $rev(\mathbf{E}_j)$

$$rev(\mathbf{E}_j) = rev([e_j^t(0) \quad \dots \quad e_j^t(N-n-1)]^T) = [e_j^t(N-n-1) \quad \dots \quad e_j^t(0)]^T . \cdot \tag{19}$$

Step C. Apply $\mathbf{R}_j + \mu \times rev(\mathbf{E}_j)$ as a reference input and obtain the output vector $\mathbf{Y}_G = \mathbf{T}(\mathbf{R}_j + \mu \times rev(\mathbf{E}_j))$, where the subscript *G* indicates the g.e. The scalar parameter μ is chosen such that the term $\mu \times rev(\mathbf{E}_j)$ represents a small deviation around the nominal $\mathbf{R}_j$.

Step D. Since $\mathbf{Y}_j = \mathbf{T}\,\mathbf{R}_j$ is known from the nominal experiment, obtain $\mathbf{T}^T\mathbf{E}_j$ as

$$\mathbf{T}^T\mathbf{E}_j = \frac{1}{\mu} rev(\mathbf{Y}_G - \mathbf{Y}_j), \tag{20}$$

and apply (18) to get the gradient $\left.\frac{\partial J}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}$.

Automatic choice of μ ensures that the nominal reference input is perturbed in an acceptable manner and the linearity is preserved.

Operational constraints are next addressed. Let $\mathbf{S}_{ur} \in \Re^{(N-m)\times(N-m)}$ be the lifted map that corresponds to the t.f. $S_{ur}(q^{-1}) = C(q^{-1})S(q^{-1})$. Using the notation *m* for the relative degree of $S_{ur}(q^{-1})$, $m \le n$, the lifted form representations are

$$\mathbf{U}=[u(m) \quad u(m+1) \quad ... \quad u(N-1)]^T,$$
$$\mathbf{R}=[r(0) \quad r(1) \quad ... \quad r(N-m-1)]^T, \tag{21}$$
$$\mathbf{S}_{ur}=\begin{bmatrix} s_1 & 0 & ... & 0 \\ s_2 & s_1 & ... & 0 \\ ... & ... & ... & ... \\ s_{N-m} & s_{N-m-1} & ... & s_1 \end{bmatrix}.$$

The expression of the c.s. is $\mathbf{U}=\mathbf{S}_{ur}\mathbf{R}$, where $\mathbf{R}\in\Re^{(N-m)\times 1}$ is a vector of greater length than in (15), for which $\mathbf{R}\in\Re^{(N-n)\times 1}$. Therefore, a truncation of $\mathbf{S}_{ur}$ corresponding to the leading principal minor of size $N-n$ is applied such that $\mathbf{S}_{ur}\in\Re^{(N-n)\times(N-n)}$. This truncation ensures that the same **R** of size $N-n$ is tuned, and this also allows only $N-n$ (out of $N-m$) constraints imposed to **U**. So even though we could benefit from the dimensionality of the map $\mathbf{S}_{ur}$, we choose only the appropriate size in order to tune the initial **R** in (15). The c.s. vector is next expressed as

$$\mathbf{U}(\mathbf{R})=\mathbf{S}_{ur}\mathbf{R}, \tag{22}$$

where $\mathbf{S}_{ur}\mathbf{R}\in\Re^{(N-n)\times(N-n)}$, and the constraints hold for $2(N-n)$ lower and upper bounds.

Using (21), the c.s. rate sequence $\Delta u(k)$ is expressed in the lifted form

$$\begin{aligned}\Delta\mathbf{U}&=[\Delta u(1) \quad \Delta u(2) \quad ... \quad \Delta u(N-n)]^T\\ &=[u(m)-0 \quad u(m+1)-u(m) \quad ...u(m+N-n-1)-u(m+N-n-2)]^T\\ &=[s_1r(0) \quad s_2r(0)+s_1r(1)-s_1r(0) \quad ... \quad s_{N-n}r(0)+...+s_1r(N-n-1)-s_{N-n-1}r(0)...-s_1r(N-n-2)]^T\\ &=\begin{bmatrix} s_1 & s_2 & s_3 & ... & s_{N-n} \\ 0 & s_1 & s_2 & ... & s_{N-n-1} \\ 0 & 0 & s_1 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & s_1 \end{bmatrix}^T\cdot\mathbf{R}-\begin{bmatrix} 0 & s_1 & s_2 & ... & s_{N-n-1} \\ 0 & 0 & s_1 & ... & s_{N-n-2} \\ 0 & 0 & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T\cdot\mathbf{R}=\mathbf{S}_{\Delta ur}\cdot\mathbf{R}.\end{aligned} \tag{23}$$

As shown in (23), the c.s. rate vector can be expressed as

$$\Delta\mathbf{U}(\mathbf{R})=\mathbf{S}_{\Delta ur}\mathbf{R},\ \mathbf{S}_{\Delta ur}=\begin{bmatrix} s_1 & 0 & 0 & ... & 0 \\ s_2 & s_1 & 0 & ... & 0 \\ s_3 & s_2 & s_1 & ... & ... \\ ... & ... & ... & ... & ... \\ s_{N-n} & s_{N-n-1} & ... & ... & s_1 \end{bmatrix}-\begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ s_1 & 0 & 0 & ... & 0 \\ s_2 & s_1 & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ s_{N-n-1} & s_{N-n-2} & ... & s_1 & 0 \end{bmatrix}, \tag{24}$$

and these constraints also hold for $2(N-n)$ lower and upper bounds. Using the notations for the vectors of lower and upper bounds

http://dx.doi.org/10.1049/iet-cta.2014.0187

$$
\begin{aligned}
\mathbf{U}_{\min} &= [u^1_{\min} \quad u^2_{\min} \quad \dots \quad u^{N-n}_{\min}]^T, \\
\mathbf{U}_{\max} &= [u^1_{\max} \quad u^2_{\max} \quad \dots \quad u^{N-n}_{\max}]^T, \\
\Delta\mathbf{U}_{\min} &= [\Delta u^1_{\min} \quad \Delta u^2_{\min} \quad \dots \quad \Delta u^{N-n}_{\min}]^T, \\
\Delta\mathbf{U}_{\max} &= [\Delta u^1_{\max} \quad \Delta u^2_{\max} \quad \dots \quad \Delta u^{N-n}_{\max}]^T,
\end{aligned} \tag{25}
$$

the inequality constraints are

$$
\begin{aligned}
&\mathbf{U}_{\min} \le \mathbf{U}(\mathbf{R}) \le \mathbf{U}_{\max}, \\
&\Delta\mathbf{U}_{\min} \le \Delta\mathbf{U}(\mathbf{R}) \le \Delta\mathbf{U}_{\max},
\end{aligned} \tag{26}
$$

and the OP which ensures the reference trajectory tracking with c.s. saturation constraints and with c.s. rate constraints is

$$
\begin{aligned}
&\mathbf{R}^* = \arg\min_{\mathbf{R}} \frac{1}{N}(\mathbf{R}^T\,\mathbf{Q}\,\mathbf{R} + \mathbf{q}\,\mathbf{R} + \alpha), \\
&\text{subject to } \widetilde{\mathbf{S}}\,\mathbf{R} \le \widetilde{\mathbf{U}} \text{ and to } \widetilde{\mathbf{S}}_\Delta\,\mathbf{R} \le \Delta\widetilde{\mathbf{U}}, \\
&\widetilde{\mathbf{S}} = [\mathbf{S}^T_{ur} \quad -\mathbf{S}^T_{ur}]^T \in \Re^{2(N-n)\times(N-n)},\ \widetilde{\mathbf{S}}_\Delta = [\mathbf{S}^T_{\Delta ur} \quad -\mathbf{S}^T_{\Delta ur}]^T \in \Re^{2(N-n)\times(N-n)}, \\
&\widetilde{\mathbf{U}} = [\mathbf{U}^T_{\max} \quad -\mathbf{U}^T_{\min}]^T \in \Re^{2(N-n)\times 1},\ \Delta\widetilde{\mathbf{U}} = [\Delta\,\mathbf{U}^T_{\max} \quad -\Delta\mathbf{U}^T_{\min}]^T \in \Re^{2(N-n)\times 1},
\end{aligned} \tag{27}
$$

A solver for this type of problems in the deterministic case is the Interior Point Barrier algorithm [18], [23]. We propose a quadratic penalty approach, with an augmented o.f. which accounts for inequality constraints concerning the c.s. and the c.s. rate:

$$
\begin{aligned}
&\widetilde{J}_{p_j}(\mathbf{R}) = J(\mathbf{R}) + p_j[\phi(\mathbf{R}) + \Delta\phi(\mathbf{R})], \text{with} \\
&\phi(\mathbf{R}) = \frac{1}{2}\sum_{h=1}^{c}\{[\max\{0, -(\underbrace{\widetilde{u}_h - \widetilde{\mathbf{s}}^T_h\mathbf{R}}_{q_h(\mathbf{R})})\}]^2\}, \\
&\Delta\phi(\mathbf{R}) = \frac{1}{2}\sum_{h=1}^{c}\{[\max\{0, -q_h(\mathbf{R})\}]^2\},
\end{aligned} \tag{28}
$$

where $\{p_j\}_{j\ge 0}$, $p_j \to \infty$, guarantees, as in (8), that the minimum of the sequence of augmented o.f.s $\{\widetilde{J}_{p_j}(\mathbf{R})\}_{j\ge 0}$ will converge to the solution to the constrained OP (27), h, $h = 1...c$, is the constraint index, $q_h(\mathbf{R}) > 0$ is h^{th} constraint, $\widetilde{u}_h$ is h^{th} element of $\widetilde{\mathbf{U}}$, and $\widetilde{\mathbf{s}}^T_h$ is h^{th} row of $\widetilde{\mathbf{S}}$. The OP with the o.f. $\widetilde{J}_{p_j}(\mathbf{R})$ given in (28) is solved using a stochastic approximation algorithm which uses the experimentally obtained gradient of $\widetilde{J}_{p_j}(\mathbf{R})$.

The quadratic penalty functions $\phi(\mathbf{R})$ and $\Delta\phi(\mathbf{R})$ in (28) fulfil the same conditions as $\phi(\boldsymbol{\rho})$ defined in (8). Therefore

$$
\frac{\partial[\max\{0, -q_h(\mathbf{R})\}]^2}{\partial r(i)} = -2\max\{0, -q_h(\mathbf{R})\}\frac{\partial q_h(\mathbf{R})}{\partial r(i)},\ i = 0...N-n-1\cdot \tag{29}
$$

http://dx.doi.org/10.1049/iet-cta.2014.0187

The expression of $\phi(\mathbf{R})$ in (28) is

$$\begin{aligned}\phi(\mathbf{R}) &= \frac{1}{2}\{[\max\{0,-(\tilde{u}_1 - s_1 r(0))\}]^2 + [\max\{0,-(\tilde{u}_2 - s_2 r(0) - s_1 r(1))\}]^2 + \ldots \\ &+ [\max\{0,-(\tilde{u}_{N-n} - s_{N-n} r(0) - \ldots - s_1 r(N-n-1))\}]^2 + [\max\{0,-(\tilde{u}_{N-n+1} + s_1 r(0))\}]^2 \\ &+ [\max\{0,-(\tilde{u}_{N-n+2} + s_2 r(0) + s_1 r(1))\}]^2 + \ldots + [\max\{0,-(\tilde{u}_{2(N-n)} + s_{N-n} r(0) + \ldots + s_1 r(N-n-1))\}]^2\}.\end{aligned} \tag{30}$$

The gradient with respect to $r(0)$ is

$$\begin{aligned}\frac{\partial\phi(\mathbf{R})}{r(0)} &= s_1 \max\{0,-(\tilde{u}_1 - s_1 r(0))\} + s_2 \max\{0,-(\tilde{u}_2 - s_2 r(0) - s_1 r(1))\} + \ldots \\ &+ s_{N-n} \max\{0,-(\tilde{u}_{N-n} - s_{N-n} r(0) - \ldots - s_1 r(N-n-1))\}]^2 - s_1 \max\{0,-(\tilde{u}_{N-n+1} + s_1 r(0))\} \\ &- s_2 \max\{0,-(\tilde{u}_{N-n+2} + s_2 r(0) + s_1 r(1))\} - \ldots - s_{N-n} \max\{0,-(\tilde{u}_{2(N-n)} + s_{N-n} r(0) + \ldots + s_1 r(N-n-1))\}\}.\end{aligned} \tag{31}$$

Using relationships that are similar to (31) for the other components of the reference input vector, the matrix form of the gradient of $\phi(\mathbf{R})$ with respect to $\mathbf{R}$ is

$$\begin{aligned}&\frac{\partial\phi(\mathbf{R})}{\partial\mathbf{R}} = \begin{bmatrix} s_1 & s_2 & \ldots & s_{N-n} \\ 0 & s_1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & s_1 \end{bmatrix} \cdot (\boldsymbol{\varepsilon}_G^1(\mathbf{R}) - \boldsymbol{\varepsilon}_G^2(\mathbf{R})) = \mathbf{S}_{ur}^T \cdot \boldsymbol{\zeta}(\mathbf{R}), \\ &\boldsymbol{\zeta}(\mathbf{R}) = \boldsymbol{\varepsilon}_G^1(\mathbf{R}) - \boldsymbol{\varepsilon}_G^2(\mathbf{R}), \\ &\boldsymbol{\varepsilon}_G^1(\mathbf{R}) = [\max\{0,-q_1(\mathbf{R})\} \quad \ldots \quad \max\{0,-q_{N-n}(\mathbf{R})\}]^T, \\ &\boldsymbol{\varepsilon}_G^2(\mathbf{R}) = [\max\{0,-q_{N-n+1}(\mathbf{R})\} \quad \ldots \quad \max\{0,-q_{2(N-n)}(\mathbf{R})\}]^T.\end{aligned} \tag{32}$$

Using (28), $\Delta\phi(\mathbf{R})$ is expressed as

$$\begin{aligned}\Delta\phi(\mathbf{R}) &= \frac{1}{2}\{(\max\{0,-\Delta u_{\max}^1 + s_1 r(0)\})^2 + (\max\{0,-\Delta u_{\max}^2 + s_2 r(0) + s_1 r(1) - s_1 r(0)\})^2 + \ldots \\ &+ (\max\{0,-\Delta u_{\max}^{N-n} + s_{N-n} r(0) + \ldots s_1 r(N-n-1) - s_{N-n-1} r(0) - \ldots - s_1 r(N-n-2)\})^2 \\ &+ (\max\{0, \Delta u_{\min}^1 - s_1 r(0)\})^2 + (\max\{0, \Delta u_{\min}^2 - s_2 r(0) - s_1 r(1) + s_1 r(0)\})^2 + \ldots \\ &+ (\max\{0, \Delta u_{\min}^{N-n} - s_{N-n} r(0) - \ldots - s_1 r(N-n-1) + s_{N-n-1} r(0) + \ldots + s_1 r(N-n-2)\})^2\}.\end{aligned} \tag{33}$$

Using (24) in (33), the gradient of $\Delta\phi(\mathbf{R})$ with respect $\mathbf{R}$ is

$$\frac{\partial \Delta\phi(\mathbf{R})}{\partial \mathbf{R}} = \begin{bmatrix} s_1 & s_2 - s_1 & s_3 - s_2 & \dots & s_{N-n} - s_{N-n-1} \\ 0 & s_1 & s_2 - s_1 & \dots & s_{N-n-1} - s_{N-n-2} \\ 0 & 0 & s_1 & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_1 \end{bmatrix} \cdot \Delta\zeta(\mathbf{R}),$$

$$\Delta\zeta(\mathbf{R}) = \Delta\boldsymbol{\varepsilon}_G^1(\mathbf{R}) - \Delta\boldsymbol{\varepsilon}_G^2(\mathbf{R}) = [\Delta\zeta(\mathbf{R},1) \quad \dots \quad \Delta\zeta(\mathbf{R}, N-n-1) \quad \Delta\zeta(\mathbf{R}, N-n)]^T,$$

$$\Delta\boldsymbol{\varepsilon}_G^1 = \begin{bmatrix} (\max\{0, -\Delta u_{\max}^1 + s_1 r(0)\}) \\ \\ \dots \\ (\max\{0, -\Delta u_{\max}^{N-n} + s_{N-n} r(0) + \dots + s_1 r(N-n-1) \\ - s_{N-n-1} r(0) - \dots - s_1 r(N-n-2)\}) \end{bmatrix},$$

$$\Delta\boldsymbol{\varepsilon}_G^2 = \begin{bmatrix} (\max\{0, \Delta u_{\min}^1 - s_1 r(0)\}) \\ \\ \dots \\ (\max\{0, \Delta u_{\min}^{N-n} - s_{N-n} r(0) - \dots - s_1 r(N-n-1) \\ + s_{N-n-1} r(0) + \dots + s_1 r(N-n-2)\}) \end{bmatrix}. \tag{34}$$

The gradient of $\Delta\phi(\mathbf{R})$ with respect to $\mathbf{R}$ is transformed as

$$\frac{\partial \Delta\phi(\mathbf{R})}{\partial \mathbf{R}} = \begin{bmatrix} s_1 & s_2 & s_3 & \dots & s_{N-n} \\ 0 & s_1 & s_2 & \dots & s_{N-n-1} \\ 0 & 0 & s_1 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & s_1 \end{bmatrix} \cdot \Delta\zeta(\mathbf{R}) - \begin{bmatrix} 0 & s_1 & s_2 & \dots & s_{N-n-1} \\ 0 & 0 & s_1 & \dots & s_{N-n-2} \\ 0 & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \Delta\zeta(\mathbf{R}) \tag{35}$$

$$= \mathbf{M}_1 \Delta\zeta(\mathbf{R}) - \mathbf{M}_2 \Delta\zeta(\mathbf{R}) = (\mathbf{M}_1 - \mathbf{M}_2)\Delta\zeta(\mathbf{R}).$$

Using (32) and (35), the expression of the gradient of the o.f. $\tilde{J}_{p_j}(\mathbf{R})$ at the current iteration j is

$$\left.\frac{\partial \tilde{J}(\mathbf{R})}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j} = \frac{2}{N}\mathbf{T}^T \mathbf{E}_j + p_j \{\mathbf{S}_{ur}^T \boldsymbol{\psi}(\mathbf{R}_j)\}, \text{ with} \tag{36}$$

$$\boldsymbol{\psi}(\mathbf{R}_j) = \zeta(\mathbf{R}_j) + \Delta\zeta(\mathbf{R}_j) - \Delta\bar{\zeta}(\mathbf{R}_j),$$

and $\Delta\bar{\zeta}(\mathbf{R}_j)$ is the one step ahead vector of dimension $N-n$

$$\Delta\bar{\zeta}(\mathbf{R}_j) = [\Delta\zeta(\mathbf{R},2) \quad \dots \quad \Delta\zeta(\mathbf{R}, N-n) \quad 0]^T. \tag{37}$$

The matrix $\mathbf{M}_1$ in (35) is exactly the map $\mathbf{S}_{ur}^T$, and $\mathbf{M}_2$ consists of the impulse response coefficients of $\mathbf{S}_{ur}^T$. Therefore, the term $(\mathbf{M}_1 - \mathbf{M}_2)\Delta\zeta(\mathbf{R})$ can be obtained in one g.e. described as follows. Since $\mathbf{M}_1 \Delta\bar{\zeta}(\mathbf{R}) = \mathbf{M}_2 \Delta\zeta(\mathbf{R})$, instead of building $\mathbf{M}_2$ from unknown coefficients of $\mathbf{S}_{ur}^T$, we experiment with a slightly modified input, i.e., $\Delta\bar{\zeta}(\mathbf{R}_j)$, to obtain the same effect as that caused by using $\mathbf{M}_2$.

http://dx.doi.org/10.1049/iet-cta.2014.0187

Finally, a single g.e. scheme can be used with the reversed vector

$$rev(\boldsymbol{\zeta}(\mathbf{R}_j) + \Delta\boldsymbol{\zeta}(\mathbf{R}_j) - \Delta\overline{\boldsymbol{\zeta}}(\mathbf{R}_j)) = rev(\boldsymbol{\psi}(\mathbf{R}_j)) \quad (38)$$

injected as a reference input taking advantage of the dimensionality of $\mathbf{S}_{ur}^T$. This single g.e. will provide the gradient with respect to all c.s. saturation and c.s. rate constraints. The same approach is used as in the previously presented four-step algorithm in order to constrain the evolution of CS in the vicinity of the nominal trajectory.

Each iteration in the algorithm requires a normal experiment with the current reference input. After the normal experiment, the g.e.s require running perturbed trajectories in the vicinity of the nominal ones. These perturbed trajectories are obtained for perturbed reference inputs with small amplitude signals. A simulation-based mechanism based on identified models is used to avoid conducting g.e.s on the real-world CS. The identified models are valid only in the vicinity of the current iteration nominal trajectories. No additional experiments are required to collect data in a wide operating range for identification purposes, so these models are used only within the current iteration. In order to extend the applicability of this approach to smooth nonlinear systems, NN-based models as NARX ones are used in the identification, with two advantages:

- The closed-loop CS behaviour is usually of low-pass type, resulting in models with rather simple dynamics.
- The numerical differentiation issues which occur in noisy environments are mitigated by our approach.

Given the nonlinear maps (5), a more compact representation that takes advantage of the super vector notation is $\overline{\mathbf{Y}} = M_{ry}(\mathbf{R})$ and $\overline{\mathbf{U}} = M_{ru}(\mathbf{R})$. The current iteration trajectories $\{\mathbf{R}_j, \mathbf{U}_j, \mathbf{Y}_j\}$ from the normal experiment are used to identify M_{ry} and M_{ru}. Using the gradient estimation scheme from (20) in (36), the estimate of $\left.\frac{\partial \tilde{J}(\mathbf{R})}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}$ is

$$\begin{aligned}
&est\{\left.\frac{\partial \tilde{J}(\mathbf{R})}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_j}\} = \frac{2}{\mu_{\mathbf{Y}}} rev(\overline{\mathbf{Y}}_{G_j} - \overline{\mathbf{Y}}_j) + p_j\{\frac{1}{\mu_{\mathbf{U}}} rev(\overline{\mathbf{U}}_{G_j} - \overline{\mathbf{U}}_j)\}, \text{ with} \\
&\overline{\mathbf{Y}}_{G_j} = M_{ry}(\mathbf{R}_j + \mu_{\mathbf{Y}} rev(\mathbf{E}_j)), \overline{\mathbf{Y}}_j = M_{ry}(\mathbf{R}_j), \\
&\overline{\mathbf{U}}_{G_j} = M_{ru}(\mathbf{R}_j + \mu_{\mathbf{U}} rev(\mathbf{\Psi}_j)), \overline{\mathbf{U}}_j = M_{ru}(\mathbf{R}_j),
\end{aligned} \quad (39)$$

where $\mu_{\mathbf{Y}} > 0$ and $\mu_{\mathbf{U}} > 0$ are scaling factors chosen such that the perturbations are only of small amplitude with respect to the current iteration reference input.

http://dx.doi.org/10.1049/iet-cta.2014.0187

3. Case study and discussion of the simulation results

The case study deals with the angular positioning of the vertical motion of a nonlinear aerodynamic twin-rotor system experimental setup [27]. The horizontal position is considered fixed, and the nonlinear equations of the vertical motion are [27]

$$\begin{aligned} &J_v\dot{\Omega}_v = l_m F_v(\omega_v) - \Omega_v k_v + g[(A-B)\cos\alpha_v - C\sin\alpha_v], \\ &\dot{\alpha}_v = \Omega_v, \\ &I_v\dot{\omega}_v = M(U_v) - M_r(\omega_v), \end{aligned} \quad (40)$$

where $U_v\ (\%) = u$ is the c.s., i.e., the PWM duty-cycle of motor's input voltage, $-24\,\text{V} \le u \le 24\,\text{V}$, $\omega_v\ (rad/s)$ is the rotor's angular speed, $\alpha_v\ (rad) = y$ is the process output, i.e., the pitch angle of the beam which supports the two rotors, and $\Omega_v\ (rad/s)$ is the beam's angular velocity. The other parameters and variables are given in [27], and the parameter values are

$$\begin{aligned} &J_v = 0.02421\, kg\, m^2,\ I_v = 4.5\cdot 10^{-5}\ kg\, m^2, \\ &k_v = 0.0127\ kg\, m^2/s,\ B - A = 0.05\ rad\, kg\, m, \\ &l_m = 0.2\, m,\ C = 0.0936\ rad\, kg\, m. \end{aligned} \quad (41)$$

The nonlinear model (40) is not used in the tuning process except for obtaining an initial controller which can also be obtained by model-free approaches as VRFT [14], [15].

A discrete-time linear PI controller with the t.f. $H(q^{-1}) = (0.012 + 0.001q^{-1})/(1-q^{-1})$ is considered initially. The reference trajectory is prescribed as the unit step response the reference model with the t.f.

$$RM(s) = \omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2),\ \text{with}\ \omega_n = 0.5\ \text{rad/s},\ \zeta = 0.7\,. \quad (42)$$

The sampling period is $T_s = 0.1\,\text{s}$ and the length of experiments is of $N = 400$ samples. The relative degree of $T(q^{-1})$ is $n = 1$ and the relative degree of $S_{ur}(q^{-1})$ is $m = 0$.

3.1. Neural network training

The NN architecture used in the identification and in the gradient estimation consists of one hidden layer with six neurons and one output layer with one neuron. Hyperbolic tangent activation functions are employed in the hidden layer, and the output neuron uses a linear activation function. The same architecture is used for both M_{ry} and M_{ru}. The inputs of the two NNs are $\mathbf{x}_{ry}^T(k) = [1\quad y(k)\quad y(k-1)\quad r(k)\quad r(k-1)]$ for M_{ry}

and $\mathbf{x}_{ru}^T(k) = [1 \quad u(k) \quad u(k-1) \quad r(k) \quad r(k-1)]$ for M_{ru}. The outputs of the NNs are $y(k)$ for M_{ry} and $u(k)$ for M_{ru}.

The two NN architectures are trained using the ILC framework with the guidelines given in [27]. Each hidden layer neuron has five parameters, i.e., four weights and one bias. The output layer has seven weights including the bias. We trained the output weight vectors $\mathbf{W} \in \mathbf{R}^{7\times1}$ and hidden units weights $\mathbf{V}_i \in \mathbf{R}^{5\times1}, \ i = 1...6$. All parameters are initialized with a zero mean normal distribution with variance 1.

The NN-based identification is carried out on the nominal trajectories of the closed-loop CS for the initial controller parameters and for the initial reference input presented in the next section. Only the results concerning the identified map M_{ry} are given here. For the norm-optimal ILC problem, the weighting matrices were chosen as $\mathbf{R} = \mathbf{I}_{400}$ and $\mathbf{Q} = 0.0001 \cdot \mathbf{I}_{37}$, where $\mathbf{I}_\zeta$ indicates the ζ^{th} order identity matrix. The evolution of the training error throughout the iterations and the evolutions of the process output before and after training are shown in Fig. 1.

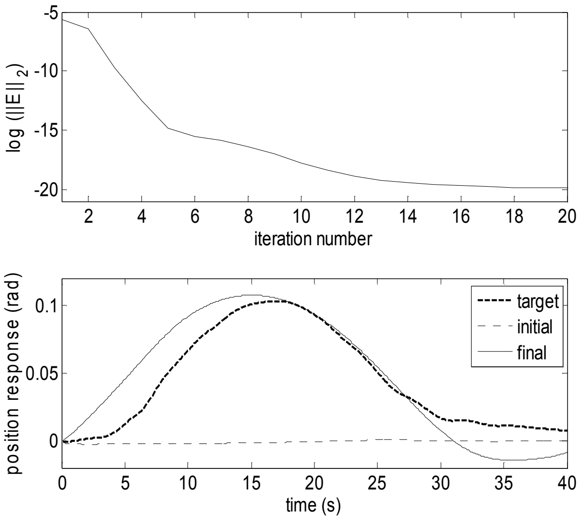

Fig. 1. NN training during the iterations and process output before and after training.

3.2. Controller tuning

With the reference input fixed, initially given in terms of the unit step response of the reference model with the t.f. (42), the controller is now optimally tuned. The tuning aims the minimization of $J(\boldsymbol{\rho})$ in (7) in this first phase. The c.s. saturation and rate constraints are considered as $-0.05 \le u(k) \le 0.18$ and

$-0.02 \le \Delta u(k) \le 0.02$. The sequence of penalty parameters of the augmented o.f. is set to $p_j = 2.5$. When no constraints are violated the search uses a BFGS update for the estimate of the Hessian and a step size of $\gamma = 0.1$; otherwise, the Hessian estimate is the identity matrix and the step size is the same, $\gamma = 0.1$. Several intermediate trajectories of the c.s. rate shown in Fig. 2 violate the upper constraint, but they are next pushed back within the boundaries. This is correlated with the activation of the penalty function in Fig. 3, which drives the tuning to ensure that the constraints are violated. When no constraints are violated, the tuning is driven to minimize the control error outlined in $J(\boldsymbol{\rho})$. The learned output trajectories given in Fig. 4 illustrate the scenario, and the final output is closer to the reference trajectory. The final controller parameters are $\boldsymbol{\rho} = [0.0192 \quad 0.0028]^T$.

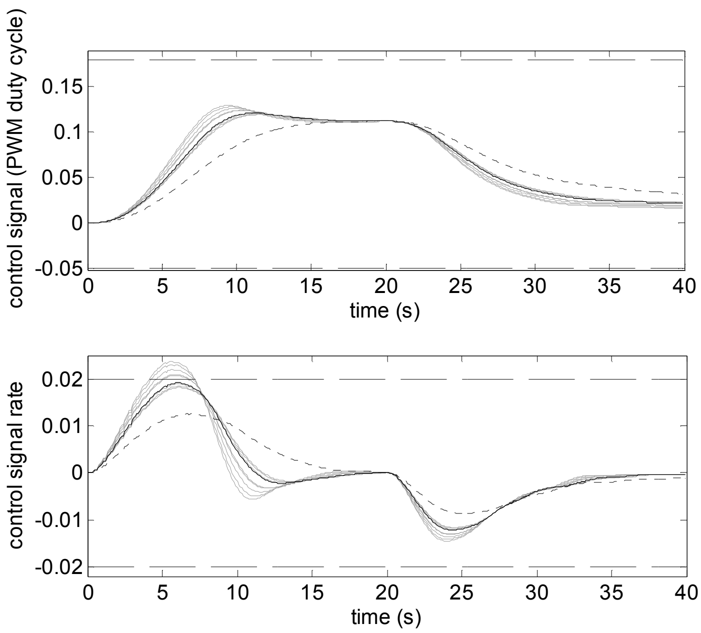

Fig. 2. Simulated responses of c.s. and of c.s. rate: initial trajectories (dotted), intermediate trajectories (grey) and final trajectories (solid black). The constraints are dashed.

http://dx.doi.org/10.1049/iet-cta.2014.0187

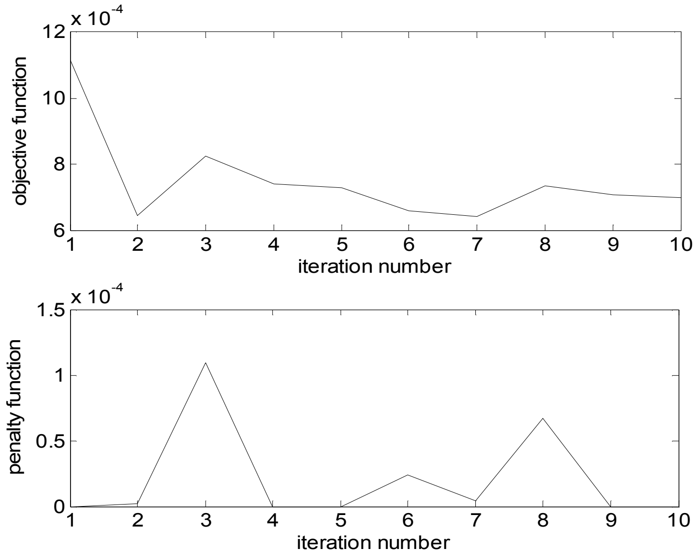

Fig. 3. Augmented objective function and penalty function versus iteration number.

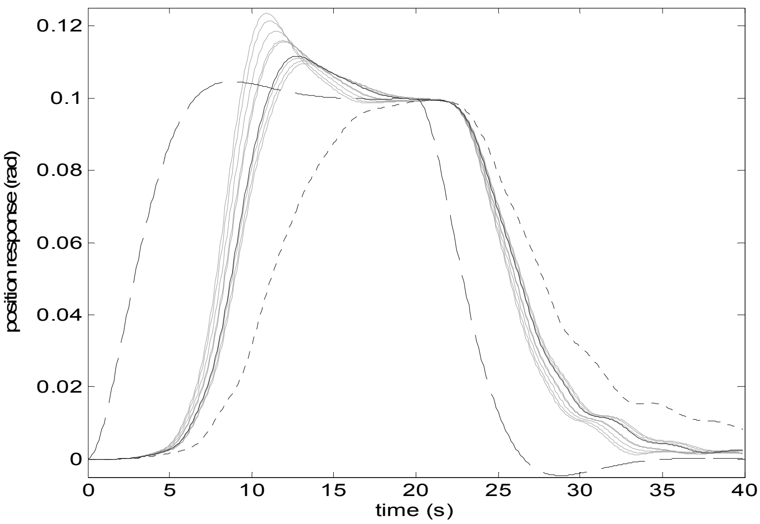

Fig. 4. Simulated responses of process output: initial trajectory (dotted), intermediate trajectories (grey) and final trajectory (solid black). The reference input is dashed.

3.3. Reference input tuning

With the controller parameters tuned and fixed, the reference input sequence is optimized in terms of the OP (16) using the same constraints.

The approach is applied as in the deterministic case as follows. The sequence of penalty parameters in (28) is set to the constant value $p_j = 25$. Two constant values of the step size are used in the gradient descent. When no constraints are violated the step size is $\gamma = 0.2$ and a BFGS Hessian update is used; otherwise, we

http://dx.doi.org/10.1049/iet-cta.2014.0187

set $\gamma = 10$ and the Hessian estimate to the unit matrix. 400 samples of the reference input sequence are subjected to optimization, and a total number of 1596 constraints are used: 798 for c.s. saturation and 798 for c.s. rate saturation.

Fig. 5 gives the evolution of the c.s. and of the c.s. rate during the learning process. Fig. 6 shows the evolution of the reference input during the learning process. The differences from the initial reference input are dramatic. Fig. 6 also shows the penalty function which contributes to the optimized augmented o.f. As the constraints are violated, they weight more in the o.f., and they eventually provide a more significant contribution to the gradient, thus driving the optimization to bring the trajectories within the feasible boundaries. This is done with the cost of the reference tracking criterion. Even with the double approximation involved in the linearity assumption and in the NN-based gradient estimation mechanism, the o.f. decreases as illustrated in Fig. 6, and the performance improvements are evident. The penalty function in Fig. 6 must be correlated with the c.s. rate constraint violation in Fig. 5. The output trajectories presented in Fig. 7 convincingly point out the CS performance improvement.

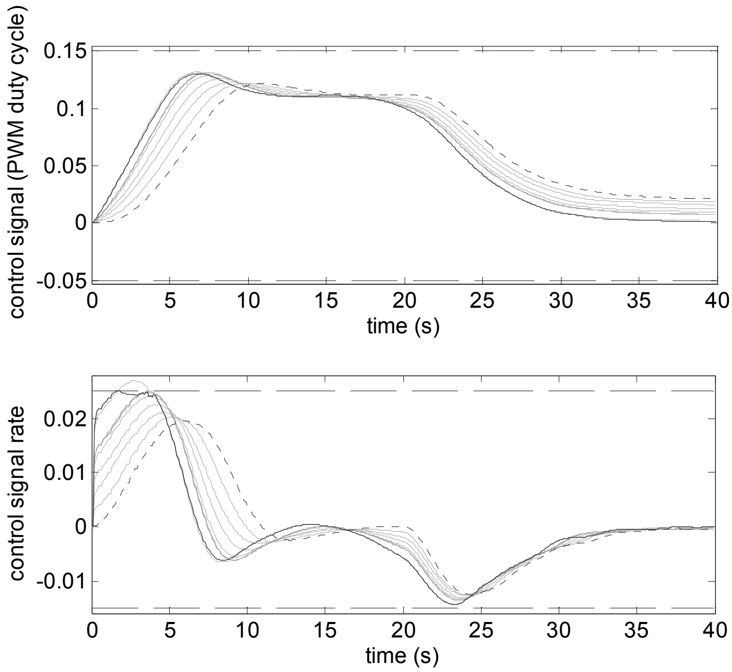

Fig. 5. Simulated responses of c.s. and of c.s. rate: initial trajectories (dotted), intermediate trajectories (grey) and final trajectories (solid black). The constraints are dashed.

http://dx.doi.org/10.1049/iet-cta.2014.0187

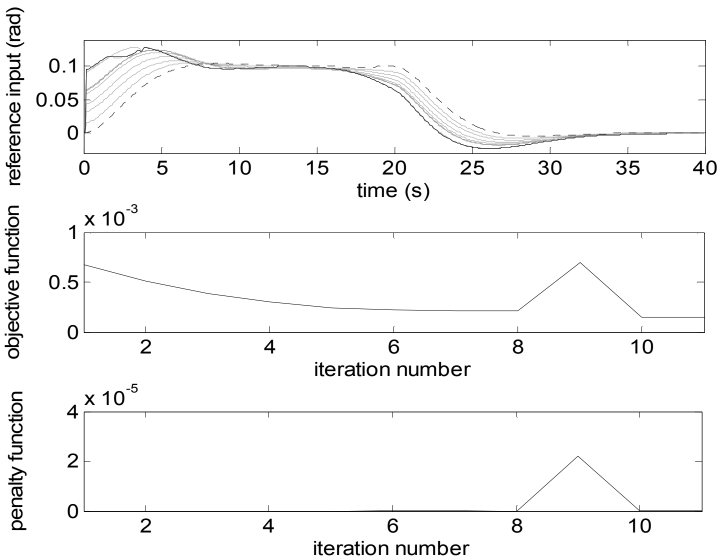

Fig. 6. Simulated reference inputs as initial one (dotted), intermediate ones (grey) and final one (solid black), augmented objective function and penalty function versus iteration number.

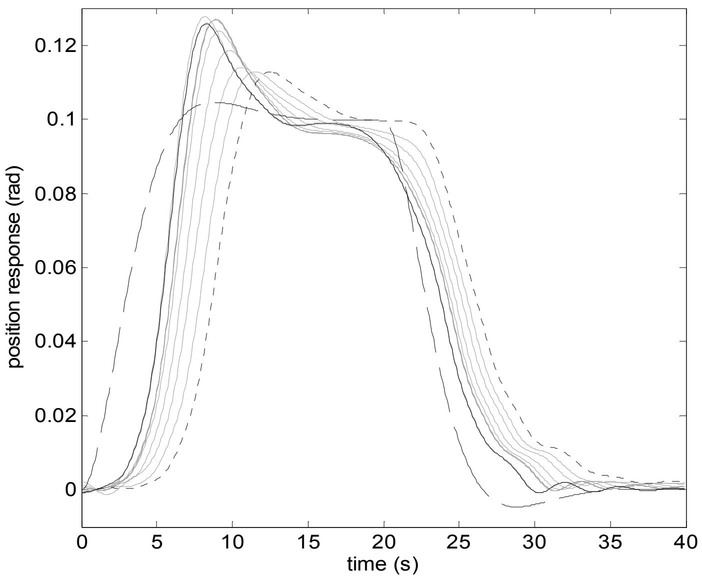

Fig. 7. Simulated responses of process output: initial trajectory (dotted), intermediate trajectories (grey) and final trajectory (solid black). The reference input is dashed.

4. Conclusions

This paper shows that:

- The same reference trajectory tracking objective can be addressed either by tuning the controller parameters or by tuning the reference input signal sequence (equivalent to tuning a reference input filter). In this sense the proposed approach can be considered as model-free 2-DOF controller tuning.

- Only the OP structure is exploited in both cases, and it does not use explicit process models. Therefore, an iterative data-based model-free control approach has been offered in this paper.
- A reactive mechanism for dealing with operational constraints has been successfully validated.

The results in the case study attest that the performance improvements are obvious. Our approach adapts well from data, compensating for process nonlinearities and uncertainties. Future research will focus the application of this approach to other convincing processes with experimental validations.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, and by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

References

[1] Hjalmarsson, H.: ‘Iterative feedback tuning - an overview’, Int. J. Adapt. Control Signal Process., 2002, 16, (5), pp. 373–395

[2] Karimi, A., Miskovic, L., and Bonvin, D.: ‘Iterative correlation-based controller tuning’, Int. J. Adapt. Control Signal Process., 2004, 18, (8), pp. 645–664

[3] Kammer, L.C., Bitmead, R.R., and Bartlett, P.L.: ‘Direct iterative tuning via spectral analysis’, Automatica, 2000, 36, (9), pp. 1301–1307

[4] Halmevaara, K., and Hyötyniemi, H.: ‘Data-based parameter optimization of dynamic simulation models’, Proc. 47th Conference on Simulation and Modelling, Helsinki, Finland, 2006, pp. 68–73

[5] Spall, J.C., and Cristion, J.A.: ‘Model-free control of nonlinear stochastic systems with discrete-time measurements’, IEEE Trans. Autom. Control, 1998, 43, (9), pp. 1198–1210

[6] Radac, M.-B., Precup, R.-E., Petriu, E.M., and Preitl, S.: ‘Application of IFT and SPSA to servo system control’, IEEE Trans. Neural Netw., 2011, 22, (12), pp. 2363–2375

[7] Kadali, R., Huang, B., and Rossiter, A.: ‘A data driven subspace approach to predictive controller design’, Control Eng. Pract., 2003, 11, (3), pp. 261–278

[8] Lu, X., Chen, H., Wang, P., and Gao, B.: 'Design of a data-driven predictive controller for start-up process of AMT vehicles', IEEE Trans. Neural Netw., 2011, 22, (11), pp. 2201–2212

[9] Fliess, M., and Join, C.: 'Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?', Proc. 15th IFAC Symposium on System Identification, Saint-Malo, France, 2009, pp. 1531–1550

[10] Hou, Z.-S., and Jin, S.: 'A novel data-driven control approach for a class of discrete-time nonlinear systems', IEEE Trans. Contr. Syst. Technol., 2011, 19, (6), pp. 1549–1558

[11] Hou, Z.-S., and Wang, Z.: 'From model-based control to data-driven control: Survey, classification and perspective', Inf. Sci., 2013, 235, pp. 3–35

[12] Safonov, M.G., and Tsao, T.-C.: 'The unfalsified control concept and learning', IEEE Trans. Automat. Control, 1997, 42, (6), pp. 843–847.

[13] McDaid, A.J., Aw, K.C., Haemmerle, E., and Xie, S.Q.: 'Control of IPMC actuators for microfluidics with adaptive "online" iterative feedback tuning', IEEE/ASME Trans. Mechatronics, 2012, 17, (4), pp. 789–797

[14] Campi, M.C., Lecchini, A., and Savaresi, S.M.: 'Virtual reference feedback tuning: a direct method for the design of feedback controllers', Automatica, 2002, 38, (8), pp. 1337–1346

[15] Formentin, S., Savaresi, S.M., and Del Re, L.: 'Noniterative direct data-driven tuning of multivariable controllers: theory and application', IET Control Theory Appl., 2012, 6, (9), pp. 1250–1257

[16] Yin., S., Ding, S., Xie, X., Luo, H., 'A review on basic data-driven approaches for industrial process monitoring,' IEEE Trans. Ind. Electron., 2014, DOI: 10.1109/TIE.2014.2301773

[17] Radac, M.-B., Precup, R.-E., Petriu, E.M., Preitl, S., and Dragos, C.-A.: 'Experiment-based approach to reference trajectory tracking', Proc. 2012 IEEE International Conference on Control Applications, Dubrovnik, Croatia, 2012, pp. 470–475

[18] Radac, M.-B., Precup, R.-E., Petriu, E.M., Preitl, S., and Dragos, C.-A.: 'Data-driven reference trajectory tracking algorithm and experimental validation', IEEE Trans. Ind. Informat., 2013, 9, (4), pp. 2327–2336

[19] Bristow, D.A., Tharayil, M., and Alleyne, A.G.: 'A survey of iterative learning control,' IEEE Control Syst. Mag., 2006, 26, (3), pp. 96–114

[20] Ahn, H.-S., Chen, Y., and Moore, K.L.: 'Iterative learning control: brief survey and categorization,' IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., 2007, 37, (6), pp. 1109–1121

[21] Butcher, M., Karimi, A., and Longchamp, R.: 'Iterative learning control based on stochastic approximation,' Proc. 17th IFAC World Congress, Seoul, Korea, 2008, pp. 1478–1483

[22] Bazanella, A.S., Campestrini, L., and Eckhard, D.: 'Data-Driven Controller Design: The H_2 Approach' (Springer-Verlag, Berlin, Heidelberg, 2012)

[23] Mishra, S., Topcu, U., and Tomizuka, M.: 'Optimization-based constrained iterative learning control,' IEEE Trans. Contr. Syst. Technol., 2011, 19, (6), pp. 1613–1621

[24] Janseens, P., Pipeleers, G., and Swevers, J.: 'A data-driven constrained norm-optimal iterative learning control framework for LTI systems,' IEEE Trans. Contr. Syst. Technol., 2013, 21, (2), pp. 546–551

[25] Lupashin, S., Schöllig, A., Sherback, M., and D'Andrea, R.: 'A simple learning strategy for high-speed quadrocopter multi-flips,' Proc. 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 1642–1648

[26] Kolter, J.Z., and Ng, A.Y.: 'Policy search via the signed derivative,' in Trinkle, J., Matsuoka, Y., and Castellanos, J.A. (Eds.): 'Robotics: Science and Systems V' (The MIT Press, Cambridge, MA, 2010), 8 pp.

[27] Radac, M.-B., Precup, R.-E., Petriu, E.M., and Preitl, S.: 'Iterative data-driven tuning of controllers for nonlinear systems with constraints,' IEEE Trans. Ind. Electron., 2014, DOI 10.1109/TIE.2014.2300068

[28] Sjöberg, J., Gutman, P.-O., Agarwal, M., and Bax, M.: 'Nonlinear controller tuning based on a sequence of identifications of linearized time-varying models,' Control Eng. Pract., 2009, 17, (2), pp. 311–321

[29] Radac, M.-B., Precup, R.-E., Petriu, E.M., Cerveneak, B.-S., Dragos, C.-A., and Preitl, S.: 'Stable iterative correlation-based tuning algorithm for servo systems,' Proc. 38th Annual Conference of IEEE Industrial Electronics Society, Montreal, QC, Canada, 2012, pp. 2500–2505

[30] Xie, X., Yin, S., Gao, H., Kaynak, O.: 'Asymptotic stability and stabilisation of uncertain delta operator systems with time-varying delays,' IET Control Theory Appl., 2013, 7, (8), pp. 1071–1078

[31] Meng, X., Lam, J., Du, B., and Gao, H. (2010): 'A delay-partitioning approach to the stability analysis of discrete-time systems,' Automatica, 2010, 46, (3), pp. 610–614

[32] Blažič, S., Matko, D., and Škrjanc, I.: 'Adaptive law with a new leakage term, IET Control Theory Appl., 2010, 4, (9), pp. 1533–1542

[33] Lam, H.K., Li, H., and Liu, H.: 'Stability analysis and control synthesis for fuzzy-observer-based controller of nonlinear systems: a fuzzy-model-based control approach,' IET Control Theory Appl., 2013, 7, (5), pp. 663–672

http://dx.doi.org/10.1049/iet-cta.2014.0187

[34] Baranyi, P., Yam, Y., and Varlaki, P.: 'TP Model Transformation in Polytopic Model-Based Control' (Taylor & Francis, Boca Raton, FL, 2013)

[35] Yin, S., Ding, S.X., Sari, A.H.A., and Hao, H.: 'Data-driven monitoring for stochastic systems and its application on batch process,' Int. J. Syst. Sci., 2013, 44, (7), pp. 1366–1376

[36] Gao, H., Zhan, W., Karimi, H.R., Yang, X., and Yin, S.: 'Allocation of actuators and sensors for coupled-adjacent-building vibration attenuation,' IEEE Trans. Ind. Electron., 2013, 60, (12), pp. 5792–5801

[37] Yin, S., Luo, H., and Ding, S.X.: 'Real-time implementation of fault-tolerant control systems with performance optimization,' IEEE Trans. Ind. Electron., 2014, 61, (5), pp. 2402–2411

[38] Sjöberg, J., De Bruyne, F., Agarwal, M., Anderson, B.D.O., Gevers, M., Kraus, F.J., and Linard, N.: 'Iterative controller optimization for nonlinear systems,' Control Eng. Pract., 2003, 11, (9), pp. 1079–1086

[39] Wang, I.-J., and Spall, J.C.: 'Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions,' Int. J. Control, 2008, 81, (8), pp. 1232–1238