Constrained Data-Driven Model-Free ILC-based Reference Input Tuning Algorithm

Mircea-Bogdan Radac[1], Radu-Emil Precup[1], Emil M. Petriu[2]

[1]Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: mircea.radac@upt.ro, radu.precup@upt.ro

[2]School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward, Ottawa, Ontario, Canada, K1N 6N5
E-mail: petriu@eecs.uottawa.ca

Abstract: This paper proposes a data-driven Iterative Reference Input Tuning (IRIT) algorithm that solves a reference trajectory tracking problem viewed as an optimization problem subjected to control signal saturation constraints and to control signal rate constraints. The IRIT algorithm incorporates an experiment-based stochastic search algorithm formulated in an Iterative Learning Control (ILC) framework in order to combine the advantages of model-free data-driven control and of ILC. The reference input vector's dimensionality is reduced by a linear parameterization. Two neural networks (NNs) trained in an ILC framework are employed to ensure a small number of experiments in the gradient estimation. The IRIT algorithm is validated by two case studies concerning the position control of a nonlinear aerodynamic system. The results prove that the IRIT algorithm offers the significant control system performance improvement by few iterations and experiments conducted on the real-world process. The paper successfully merges the use of ILC in both model-free reference input tuning and NN training.

Keywords: constraints; Iterative Reference Input Tuning algorithm; linear parameterization; mechatronics; neural networks

1 Introduction

The reference trajectory tracking problem can be considered as a reference input design of an initial Control System (CS) with a priori tuned feedback controllers for stability and disturbance rejection. Therefore, the reference trajectory tracking is defined as an open-loop optimal control problem. The data-driven solving of this Optimization Problem (OP) can be carried out in the Iterative Learning Control (ILC) framework, where the sequence of reference input signal samples is updated at each iteration. In this setting, the reference input tuning is regarded as

the optimization variable and the solution to the OP is based on a gradient search algorithm. The gradient information is obtained experimentally without using any knowledge on the process.

In the context of the above features, the proposed approach ensures the data-driven model-free iterative tuning. Hence, our approach shares some similarities with other related approaches to iterative and adaptive model-free control, with several advantages versus the model-based controller tuning [1, 2].

This paper applies ILC to both reference input tuning and neural network (NN) training. The ILC-based solving of optimal control problems is formulated in [3] and [4], time and frequency domain convergence analyses are conducted in [5], the stochastic approximation is treated in [6], and the output tracking is discussed in [7]. The affine constraints are handled in [8] by the transformation of ILC problems with quadratic objective functions (o.f.s) into convex quadratic programs. The system impulse response is estimated in [9] using input/output measurements from previous iterations and next used in a norm-optimal ILC structure that accounts for actuator limitations by means of linear inequality constraints. A learning approach that gives the parameters of motion primitives for achieving flips for quadrocopters is proposed in [10], but it makes use of approximate simple models of the process. Similar formulations with reinforcement learning for policy search using approximate models and signed derivative are given in [11]. NNs applied to ILC in a model-based approach are also reported in [12].

Our recent results given in [13] and [14] are focused on an experiment-based approach to the reference trajectory tracking that takes into account control signal saturation constraints, it employs an Interior Point Barrier (IPB) algorithm, and both simulated and experimental case studies are included. We have applied ILC in [15] to the reference input tuning subject to control signal saturation constraints and control signal rate constraints using max-type quadratic penalty functions, and the results have been validated on a simulated case study related to a nonlinear aerodynamic system. The ILC-based training of NNs has been proposed in [16] in order to reduce the number of experiments of IFT with operational constraints for nonlinear systems, and tested by means of an experimental case study. The IFT with operational constraints applied to data-driven controllers tuned for a reduced sensitivity has been suggested in [17]; an NN identification mechanism has provided the gradient information used in the search algorithm, a perturbation-based approach has been involved in the estimation of second-order derivatives, and the results have been validated by a simulated case study and compared with SPSA and with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update algorithm.

This paper is built upon these results, and the main contribution with respect to the state-of-the-art is an experiment-based Iterative Reference Input Tuning (IRIT) algorithm that solves the constrained reference trajectory tracking. This is advantageous because: the dimensionality of the reference input vector is reduced

by a linear parameterization that enables cost-effective controller designs and implementations, the NN-based identification mechanism applied to the nonlinear CS leads to a simple, effective and general IRIT algorithm with a reduced number of experiments, the involvement of ILC in IRIT and NN training makes our approach a special case of supervised learning according to the relationships discussed in [14]. This strong involvement determines our IRIT-based CSs to benefit of the advantages of ILC highlighted in a mechatronics application. The proposed solution is model-free as opposed to the model-based solutions for constrained ILC presented in [8], [18], [19].

The paper is organized as follows. The next section presents the formulation of the problem that concerns the reference trajectory tracking problem solved in the data-driven optimal ILC framework. Section 3 deals with the model-free estimation of o.f.'s gradient. Section 4 proposes the model-free constrained optimal control problem and gives the formulation of the IRIT algorithm. Section 5 motivates the use of the NN-based approach in gradient estimation. Section 6 validates the IRIT algorithm by two simulated case studies that deal with the angular position control of a nonlinear aerodynamic system. The results and their discussion convincingly validate the new IRIT algorithm. The conclusions are highlighted in Section 7.

2 Data-driven Approach to Reference Trajectory Tracking

2.1 Problem Formulation

The CS is characterized by the discrete time Linear Time-Invariant (LTI) Single Input-Single Output (SISO) model

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

where k is the discrete time argument, $y(k)$ is the process output sequence, $r(k)$ is the reference input sequence, $v(k)$ is the zero-mean stationary and bounded stochastic disturbance input sequence acting on the process output and accounting for various types of load or measurement disturbances, $S(\boldsymbol{\rho}, q^{-1})$ is the sensitivity function, $T(\boldsymbol{\rho}, q^{-1})$ is the complementary sensitivity function

$$S(\boldsymbol{\rho}, q^{-1}) = 1/[1 + P(q^{-1})C(\boldsymbol{\rho}, q^{-1})], T(\boldsymbol{\rho}, q^{-1}) = 1 - S(\boldsymbol{\rho}, q^{-1}), \tag{2}$$

$P(q^{-1})$ is the process transfer function (t.f.), $C(\boldsymbol{\rho}, q^{-1})$ is the controller t.f., which is 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 certain equations for the sake of simplicity.

An ILC framework to describe the reference trajectory tracking problem is introduced using the lifted form (or super vector) representation. For a relative degree n of the closed-loop CS t.f. $T(\boldsymbol{\rho}, q^{-1})$, the lifted form representation for an N samples experiment length and the matrices in the deterministic case are

$$\begin{aligned}&\mathbf{Y} = \mathbf{T}\,\mathbf{R} + \mathbf{Y}_0,\\ &\mathbf{Y} = [y(n) \quad y(n+1) \quad ... \quad y(N-1)]^T, \mathbf{R} = [r(0) \quad r(1) \quad ... \quad r(N-n-1)]^T,\\ &\mathbf{Y}_0 = [y_{10} \quad y_{20} \quad ... \quad y_{(N-n)0}]^T, \mathbf{T} = \begin{bmatrix} t_1 & 0 & ... & 0 \\ t_2 & t_1 & ... & 0 \\ ... & ... & ... & ... \\ t_{N-n} & t_{N-n-1} & ... & t_1 \end{bmatrix},\end{aligned} \tag{3}$$

where $\mathbf{R}$ is the reference input vector that contains the reference input sequence over the time interval $0 \le k \le N-n-1$, $\mathbf{Y}$ is the controlled output vector, t_i is the i^{th} impulse response coefficient of $T(\boldsymbol{\rho}, q^{-1})$, $\mathbf{T}$ 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 T indicates matrix transposition. Zero initial conditions are assumed without loss of generality, and the tracking error vector $\mathbf{E}$

$$\mathbf{E} = \mathbf{Y} - \mathbf{Y}^d = \mathbf{T}\,\mathbf{R} - \mathbf{Y}^d, \tag{4}$$

where $\mathbf{Y}^d$ is the desired reference trajectory vector generated from the desired process output $y^d(k)$. Equation (4) shows that knowledge on $\mathbf{T}$ would provide the optimal solution which makes the tracking error zero, i.e., $\mathbf{R} = \mathbf{T}^{-1}\mathbf{Y}^d$. However, $\mathbf{T}$ can be ill-conditioned, and this matrix is always subject to measurement errors; therefore, $\mathbf{T}^{-1}$ cannot be used. A solution to the iterative estimation of $\mathbf{T}$ in an ILC framework is given in [9].

The control objective is expressed as the following OP that involves the expected normalized norm of the tracking error:

$$\begin{aligned}&\mathbf{R}^* = \arg\min_{\mathbf{R}} J(\mathbf{R}) = E\{\frac{1}{N} \| \mathbf{E}(\mathbf{R}) \|_2^2\}\\ &\text{subject to system dynamics (1) and to some operational constraints,}\\ &J(\mathbf{R}) = E\{\frac{1}{N}(\mathbf{T}\,\mathbf{R} - \underbrace{\mathbf{Y}^d}_{\mathbf{M}})^T(\mathbf{T}\,\mathbf{R} - \mathbf{Y}^d)\} = E\{\frac{1}{N}(\mathbf{R}^T\,\mathbf{Q}\,\mathbf{R} + 2\mathbf{q}\,\mathbf{R} + \alpha)\},\end{aligned} \tag{5}$$

the deterministic formulation of the o.f. $J(\mathbf{R})$ is quadratic with respect to $\mathbf{R}$, where $\mathbf{Q} = \mathbf{T}^T\mathbf{T}$ is a positive semi-definite matrix, $\mathbf{q} = \mathbf{M}^T\,\mathbf{T}$, and $\alpha = \mathbf{M}^T\mathbf{M}$. A gradient descent approach to iteratively solve (5) is

$$\mathbf{R}_{j+1} = \mathbf{R}_j - \gamma_j \tilde{\mathbf{H}}_{\mathbf{R}}^{-1} est\{\frac{\partial J}{\partial \mathbf{R}}\Big|_{\mathbf{R}=\mathbf{R}_j}\}, \tag{6}$$

where the subscript j is the iteration or trial index, $est\{\frac{\partial J}{\partial \mathbf{R}}\Big|_{\mathbf{R}=\mathbf{R}_j}\}$ is the estimate of the gradient of the o.f. with respect to the reference input vector samples, $\tilde{\mathbf{H}}_{\mathbf{R}}^{-1}$ is a Gauss-Newton approximation of the Hessian of the o.f., typically given by a BFGS update algorithm, and γ_j is the step size of the update law (6). When no model information is used for the choice of γ_j in order to guarantee the convergence of the search algorithm [3-9], a small enough value of the step size will usually ensure the convergence. This renders our approach a truly model-free one.

The stochastic convergence of ILC algorithms treated in [5, 6] is related to two imposed stochastic convergence conditions: the estimated o.f.'s gradient is unbiased, and the step size sequence $\{\gamma_j\}_{j\geq 0}$ converges to zero but not too fast. Constant values of the step size can be set in practical experiments, where the theoretical convergence is not targeted and few iterations are aimed. The deterministic formulation of the OP (5) will be employed in the next sections.

2.2 Reducing the Dimensionality of the Reference Input Vector

Using the reference input vector tuning as in [13, 14], the dimension of the search space is usually high, of about hundreds of samples of the reference input signal to be optimized. A linear transformation is considered in order to reduce the reference input vector dimension. A common linear parameterization can be a polynomial fit of a certain order, a Fourier fit or a Gaussian fit, all of them linear in the parameters. For an h_r degree polynomial for which

$$r(k)=\sum_{i=0}^{h_r} k^i\theta_i,\ 0\leq k\leq N-n-1, \tag{7}$$

the reference input vector is expressed according to the linear transformation

$$\begin{aligned}&\mathbf{R}=\boldsymbol{\Gamma}\,\boldsymbol{\theta},\\ &\boldsymbol{\Gamma}=[\Gamma_{ij}]_{i=1\ldots N-n,\,j=1\ldots h_r+1},\Gamma_{i1}=1,\\ &\boldsymbol{\theta}=[\theta_0\quad\ldots\quad\theta_{h_r}]^T.\end{aligned} \tag{8}$$

The OP given in (5) is also quadratic in $\boldsymbol{\theta}$ by the virtue of the linear transformation.

This reduction of the OP dimension may be useful for several reasons. The convergence to a local minimum of the o.f. can be accelerated; in addition, the ill-conditioning of the BFGS update algorithm can be avoided. The idea of reducing the dimension of the learning space in an ILC formulation is also treated in several

approaches in [20-23]. These approaches range from decomposing the reference signal using different types of basis functions, to down sampling the reference signals. However, this problem handling is considered in a model-based context and not in a model-free one as in our case.

3 Model-free Estimation of the Gradient

Using (8) in (5), the o.f. will be quadratic with respect to $\boldsymbol{\theta}$. Hence

$$J(\boldsymbol{\theta}) = \frac{1}{N}(\boldsymbol{\theta}^T \boldsymbol{\Gamma}^T \mathbf{Q} \boldsymbol{\Gamma} \boldsymbol{\theta} + 2\mathbf{q} \boldsymbol{\Gamma} \boldsymbol{\theta} + \alpha) \cdot \tag{9}$$

A gradient search is performed to find the minimum of this function. The analytic solution is not desired because it depends on the matrices **Q** and **q** that depend on the unknown **T**. The gradient search using a Gauss-Newton approximation of the Hessian of the new o.f. $J(\boldsymbol{\theta})$ is

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j - \gamma_j \tilde{\mathbf{H}}_{\boldsymbol{\theta}}^{-1} \left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} \cdot \tag{10}$$

A model-free approach to the gradient estimation is given in [15] and reformulated here. From (9), using the matrix derivation rules and the fact that $\boldsymbol{\Gamma}^T \mathbf{Q} \boldsymbol{\Gamma}$ is symmetric by the virtue of $\mathbf{Q} = \mathbf{T}^T \mathbf{T}$ being symmetric, the gradient of $J(\boldsymbol{\theta})$ with respect to the parameter vector $\boldsymbol{\theta}$ will be

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} = \frac{2}{N}(\boldsymbol{\Gamma}^T \mathbf{T}^T \mathbf{T} \boldsymbol{\Gamma} \boldsymbol{\theta} + \boldsymbol{\Gamma}^T \mathbf{T}^T \mathbf{M}) = \frac{2}{N} \boldsymbol{\Gamma}^T \mathbf{T}^T (\mathbf{T} \boldsymbol{\Gamma} \boldsymbol{\theta} + \mathbf{M}) \cdot \tag{11}$$

But $\mathbf{T} \boldsymbol{\Gamma} \boldsymbol{\theta} + \mathbf{M} = \mathbf{E}$, and the gradient of $J(\boldsymbol{\theta})$ in the deterministic case at each iteration *j* is finally expressed as

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} = \frac{2}{N} \boldsymbol{\Gamma}^T \mathbf{T}^T \mathbf{E}_j \cdot \tag{12}$$

In (12), $\frac{2}{N} \mathbf{T}^T \mathbf{E}_j$ is actually the gradient of $J(\mathbf{R})$ from (5) with respect to **R**. Therefore, using (8)

$$\frac{\partial J(\mathbf{R}(\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} = \frac{2}{N} \frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial J(\mathbf{R})}{\partial \mathbf{R}} = \frac{2}{N} \boldsymbol{\Gamma}^T \frac{\partial J(\mathbf{R})}{\partial \mathbf{R}} \cdot \tag{13}$$

Equation (13) can be interpreted as the chain derivation rule of the function $J(\mathbf{R}(\boldsymbol{\theta}))$ with respect to $\boldsymbol{\theta}$, and it will be used later in the paper. Equation (12) suggests that the gradient information can be obtained either by an experimentally measured **T** or by using a special gradient experiment at each iteration.

The second approach is preferred in our case and it is next presented. Different solutions to the feed-forward optimal control design problem using finite-differences approximations of the gradient by experiments with perturbed parameters are presented in [24, 25].

The successive updates (10) for the parameterized reference input trajectory are performed in the vicinity of the current iteration reference input trajectory. The linearity assumption and operation can therefore be justified in this case. As we see from (12), the gradient vector is obtained experimentally driving the closed-loop CS in non-nominal operating regimes because the current iteration error $\mathbf{E}_j$ is used as a reference input in the gradient estimation scheme according to [15]. For a linear system this does not affect the quality of the gradient information although it may affect the nominal operation of the CS. In order to allow for near-nominal experimenting regimes to be used with linear systems and to further extend the applicability of the IRIT algorithm to nonlinear systems a perturbation-based approach is proposed to obtain the gradient information near the nominal trajectory. This idea stems from [26], and it is a modified version of the algorithm used in [15]. The model-free gradient estimation algorithm consists of the following steps:

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

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

$$\begin{aligned} &rev(\mathbf{E}_j) = rev([e_j(0) \quad ... \quad e_j(N-n-1)]^T) = [e_j(N-n-1) \quad ... \quad e_j(0)]^T, \\ &e_j(k) = y(n+k) - y^d(n+k), 0 \le k \le N-n-1. \end{aligned} \tag{14}$$

Step C. Apply $\mathbf{\Gamma\theta}_j + \mu \times rev(\mathbf{E}_j)$ as a reference input to the CS and obtain the output vector $\mathbf{Y}_G = \mathbf{T}(\mathbf{\Gamma\theta}_j + \mu \times rev(\mathbf{E}_j))$ where the subscript G stands for "gradient". The scalar coefficient μ is chosen such that the perturbed term $\mu \times rev(\mathbf{E}_j)$ represents only a small deviation around the nominal reference input trajectory $\mathbf{R}_j = \mathbf{\Gamma\theta}_j$.

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

$$\mathbf{\Gamma}^T\mathbf{T}^T\mathbf{E}_j = \frac{1}{\mu}\mathbf{\Gamma}^T rev(\mathbf{Y}_G - \mathbf{Y}_j), \tag{15}$$

and use (15) in (12) to get the gradient $\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j}$.

The choice of the parameter μ can be done automatically such that the nominal reference input is not perturbed too much in amplitude.

4 Dealing with Control Signal Saturation and Control Signal Rate Constraints

The operational constraints regarding the saturation of actuators, the saturation of the control signal rate or the bounds on the state variables of the process are very important in many real-world CS applications. Different numerical algorithms can be employed in model-based approaches to solve the OP (5) for such systems. However, a model-free approach is presented as follows.

The lifted form representations allow the expression of a particular form of the OP that can be of interest. Assuming the deterministic case, let $\mathbf{S}_{ur} \in \Re^{(N-m)\times(N-m)}$ be the lifted map that corresponds to the t.f. $S_{ur}(q^{-1}) = C(q^{-1})S(q^{-1})$, where $\Re$ is the set of real numbers. Using the notation m for the relative degree of $S_{ur}(q^{-1})$, $m \le n$, the lifted form representations are [15]

$$\begin{aligned}
&\mathbf{U} = [u(m) \quad u(m+1) \quad ... \quad u(N-1)]^T, \mathbf{R} = [r(0) \quad r(1) \quad ... \quad r(N-m-1)]^T, \\
&\mathbf{U} = \begin{bmatrix} s_1 & 0 & ... & 0 \\ s_2 & s_1 & ... & 0 \\ ... & ... & ... & ... \\ s_{N-m} & s_{N-m-1} & ... & s_1 \end{bmatrix} \cdot \mathbf{R} = \mathbf{S}_{ur} \cdot \mathbf{R}, \Delta\mathbf{U} = [\Delta u(1) \quad \Delta u(2) \quad ... \quad \Delta u(N-n)]^T \\
&= [u(m) - 0 \;\; u(m+1) - u(m) \;\; ... \;\; u(m+N-n-1) - u(m+N-n-2)]^T \\
&= [s_1 r(0) s_2 r(0) + s_1 r(1) - s_1 r(0) \;\; ... \;\; s_{N-n} r(0) + ... + s_1 r(N-n-1) - s_{N-n-1} r(0) \\
&- ... - s_1 r(N-n-2)]^T \\
&= \begin{bmatrix} s_1 & s_2 & s_3 & ... & s_{N-n} \\ 0 & s_1 & s_2 & ... & s_{N-n-1} \\ 0 & 0 & s_1 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & s_1 \end{bmatrix}^T \cdot \mathbf{R} - \begin{bmatrix} 0 & s_1 & s_2 & ... & s_{N-n-1} \\ 0 & 0 & s_1 & ... & s_{N-n-2} \\ 0 & 0 & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \cdot \mathbf{R} = \mathbf{S}_{\Delta ur} \cdot \mathbf{R},
\end{aligned} \tag{16}$$

where $\mathbf{R} \in \Re^{(N-m)\times 1}$ is a vector of greater length than in (3), 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 considered such that $\mathbf{S}_{ur} \in \Re^{(N-n)\times(N-n)}$ because we need the same $\mathbf{R}$ of size $N-n$ to be tuned, and this in turn will allow only $N-n$ (out of $N-m$) constraints imposed to $\mathbf{U}$, and the affine constraints $\mathbf{U}_{\min} \le \mathbf{U}(\mathbf{R}) \le \mathbf{U}_{\max}$ and $\Delta\mathbf{U}_{\min} \le \Delta\mathbf{U}(\mathbf{R}) \le \Delta\mathbf{U}_{\max}$ are imposed to $\mathbf{R}$.

The OP, which ensures the reference trajectory tracking with control signal constraints and with control signal rate constraints is expressed as

$$\begin{aligned}
&\boldsymbol{\theta}^* = \arg\min_{\mathbf{R}} \frac{1}{N}(\boldsymbol{\theta}^T \boldsymbol{\Gamma}^T \mathbf{Q}\boldsymbol{\Gamma}\boldsymbol{\theta} + 2\mathbf{q}\boldsymbol{\Gamma}\boldsymbol{\theta} + \alpha),\\
&\text{subject to dynamics (1), and to } \tilde{\mathbf{S}}\boldsymbol{\Gamma}\boldsymbol{\theta} \le \tilde{\mathbf{U}} \text{ and } \tilde{\mathbf{S}}_\Delta \boldsymbol{\Gamma}\boldsymbol{\theta} \le \Delta\tilde{\mathbf{U}}, \text{ where}\\
&\tilde{\mathbf{S}} = [\mathbf{S}_{ur}^T \quad -\mathbf{S}_{ur}^T]^T \in \Re^{2(N-n)\times(N-n)}, \tilde{\mathbf{S}}_\Delta = [\mathbf{S}_{\Delta ur}^T \quad -\mathbf{S}_{\Delta ur}^T]^T \in \Re^{2(N-n)\times(N-n)}\\
&\tilde{\mathbf{U}} = [\mathbf{U}_{\max}^T \quad -\mathbf{U}_{\min}^T]^T \in \Re^{2(N-n)\times 1}, \Delta\tilde{\mathbf{U}} = [\Delta\mathbf{U}_{\max}^T \quad -\Delta\mathbf{U}_{\min}^T]^T \in \Re^{2(N-n)\times 1},\\
&\mathbf{U}_{\min} = [u_{\min}^1 \quad u_{\min}^2 \quad \ldots \quad u_{\min}^{N-n}]^T, \mathbf{U}_{\max} = [u_{\max}^1 \quad u_{\max}^2 \quad \ldots \quad u_{\max}^{N-n}]^T.
\end{aligned} \tag{17}$$

A solver for this type of problems in the deterministic case is the IPB algorithm [8, 14]. As we have shown in [14] for inequality constraints concerning only the control signal saturation, the constrained OP is transformed into an unconstrained OP by the use of the penalty functions. The logarithmic barrier penalty function grows unbounded as the constraints are close to being violated and in the stochastic framework this is always the case. A solution to overcome this problem is given in [27, 28], but with quadratic penalty functions. We propose the following augmented o.f. that accounts for inequality constraints concerning the control signal saturation and the control signal rate:

$$\tilde{J}_{p_j}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + p_j[\underbrace{\frac{1}{2}\sum_{h=1}^{c}\{[\max\{0,-(\underbrace{\tilde{u}_h - \tilde{\mathbf{s}}_h^T\boldsymbol{\Gamma}\boldsymbol{\theta}}_{q_h(\boldsymbol{\theta})})\}]^2\}}_{\phi(\boldsymbol{\theta})} + \underbrace{\frac{1}{2}\sum_{h=1}^{c}\{[\max\{0,-q_h(\boldsymbol{\Gamma}\boldsymbol{\theta})\}]^2\}}_{\Delta\phi(\boldsymbol{\theta})}], \tag{18}$$

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 $\{\tilde{J}_{p_j}(\boldsymbol{\theta})\}_{j\ge 0}$ will converge to the solution to the constrained OP (17), h, $h = 1...c$, is the constraint index, $q_h(\boldsymbol{\theta}) > 0$ is h^{th} constraint, $\tilde{u}_h$ is h^{th} element of $\tilde{\mathbf{U}}$, and $\tilde{\mathbf{s}}_h^T$ is h^{th} row of $\tilde{\mathbf{S}}$. The OP (17) is solved using a stochastic approximation algorithm that makes use of the experimentally obtained gradient of $\tilde{J}_{p_j}(\boldsymbol{\theta})$. For practical applications, where stochastic convergence is not targeted and a few number of iterations is desired, the penalty parameters can be chosen as $p_j = p = \text{const}$.

The quadratic penalty functions $\phi(\boldsymbol{\theta})$ and $\Delta\phi(\boldsymbol{\theta})$ in (18) corresponding to the control saturation and control rate constraints use the *max* function, which in this case is non-differentiable only at zero. Given that $\phi(\boldsymbol{\theta})$ and $\Delta\phi(\boldsymbol{\theta})$ are 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 [27]. Therefore

$$\frac{\partial[\max\{0,-q_h(\boldsymbol{\theta})\}]^2}{\partial r(i)} = -2\max\{0,-q_h(\boldsymbol{\theta})\}\cdot\frac{\partial q_h(\boldsymbol{\theta})}{\partial r(i)}. \tag{19}$$

Using the gradient of $\phi(\mathbf{R})$ with respect to $\mathbf{R}$ given in [15], the linear transformation (8) and the chain derivation rule with respect to vectors lead to the expression of the gradient of $\phi(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\frac{\partial \phi(\mathbf{R}=\boldsymbol{\Gamma}\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{\Gamma}^T \mathbf{S}_{ur}^T \boldsymbol{\zeta}(\boldsymbol{\theta}). \tag{20}$$

The gradient of $\Delta\phi(\mathbf{R}=\boldsymbol{\Gamma}\boldsymbol{\theta})$ in (18) with respect to the $N-n$ elements of $\mathbf{R}$ (from 0 to $N-n-1$) is given in [15], and the linear transformation (8), next the chain derivation rule with respect to vectors lead to the expression of the gradient of $\Delta\phi(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\frac{\partial \Delta\phi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{\Gamma}^T (\mathbf{M}_1 - \mathbf{M}_2) \Delta\boldsymbol{\zeta}(\boldsymbol{\theta}) \cdot \tag{21}$$

Using (19), (20) and (21), the expression of the gradient of the o.f. (18) at the current iteration *j* is

$$\left.\frac{\partial \tilde{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} = \frac{2}{N} \boldsymbol{\Gamma}^{\mathrm{T}} \mathbf{T}^T \mathbf{E}_j + p_j \{\boldsymbol{\Gamma}^{\mathrm{T}} \mathbf{S}_{ur}^T [\underbrace{\boldsymbol{\zeta}(\boldsymbol{\theta}_j) + \Delta\boldsymbol{\zeta}(\boldsymbol{\theta}_j) - \Delta\bar{\boldsymbol{\zeta}}(\boldsymbol{\theta}_j)}_{\boldsymbol{\psi}(\boldsymbol{\theta}_j)}]\}, \tag{22}$$

where $\boldsymbol{\theta} \in \Re^{(h+1)\times 1}$, $\boldsymbol{\zeta}(\boldsymbol{\theta}_j)$ is considered in [15] as $\boldsymbol{\zeta}(\boldsymbol{\theta})$ is seen as a function of $\boldsymbol{\theta}$ via the transformation (8), $\Delta\boldsymbol{\zeta}(\boldsymbol{\theta}_j)$ is considered in [16] as $\Delta\boldsymbol{\zeta}(\boldsymbol{\theta})$ is seen as a function of $\boldsymbol{\theta}$ via the transformation (8), and $\Delta\bar{\boldsymbol{\zeta}}(\boldsymbol{\theta}_j)$ is a one step ahead vector of dimension $N-n$:

$$\Delta\bar{\boldsymbol{\zeta}}(\boldsymbol{\theta}_j) = [\Delta\boldsymbol{\zeta}(\boldsymbol{\theta},2) \quad ... \quad \Delta\boldsymbol{\zeta}(\boldsymbol{\theta},N-n) \quad 0]^T. \tag{23}$$

As shown in [16], the matrix term in the expression of $\frac{\partial \Delta\phi(\mathbf{R})}{\partial \mathbf{R}}$ and the structure of the matrices justifies the use of a single gradient experiment, with $rev(\boldsymbol{\zeta}+\Delta\boldsymbol{\zeta}-\Delta\bar{\boldsymbol{\zeta}}) = rev(\boldsymbol{\psi}(\boldsymbol{\theta}_j))$ injected as the reference input to the CS, taking advantage of the dimensionality of the map $\mathbf{S}_{ur}^T$. The same approach will be used as in the gradient estimation algorithm given in Section 3 in order to constrain the evolution of the dynamic system in the vicinity of the nominal trajectory.

The feasibility is not preserved during the tuning since the constraints are weighted in the o.f. only when they are violated. The feasibility is not a problem because our approach allows the initialization of a solution that is not initially feasible. But this causes the nonlinear behaviour when the constraints are active and therefore it is not recommended. However, in the long-term run, as the sequence $\{p_j\}_{j\geq 0}$ increases, the gradient due to the constraints that are violated is decisive, and the reference trajectory tracking objective is neglected with the expense of fulfilling OP's constraints. The constraints are active and they vary only subjected to the random effects of the noise affecting the closed-loop system.

For non-minimum-phase systems, the iterative reference input update (6) or (10) may lead to unbounded growth of the reference input's amplitude because this update will try to compensate for the non-minimum-phase character of the system response. In terms of the analytical solution to the reference trajectory tracking problem $\mathbf{R}=\mathbf{T}^{-1}\mathbf{Y}^{d}$, this corresponds to filtering $\mathbf{Y}^{d}$ through the inverse of an unstable map $\mathbf{T}$. We propose three solutions to this problem, briefly outlined as follows. The first solution requires that the desired trajectory should have a non-minimum-phase character. The second solution uses a regularization factor in the definition of the original o.f. given in (5), for example as the weighted norm of the reference input $\lambda \| \mathbf{R} \|_2^2$, where $\lambda > 0$ is a scalar weight. This will balance the o.f. and the growth of the amplitude of the reference input will be limited. The third solution is based on the fact that the introduction of constraints on the control signal and on the control signal rate will indirectly limit the amplitude of the reference input as an unbounded reference input will generate an unbounded control input signal. Therefore, out approach indirectly solves the reference trajectory tracking problem for non-minimum-phase systems by taking into account the control signal constraints.

Our IRIT algorithm consists of the following steps:

Step S1. Start with the initial guess of $\mathbf{R}$. Calculate the regressor $\mathbf{\Gamma}$, perform data normalization on the regressor, and fit the initial $\boldsymbol{\theta}$ using a least squares algorithm. Choose the upper and lower bounds for the control signal, the upper and lower bounds for the control signal rate and generate the desired reference trajectory vector $\mathbf{Y}^{d}$. Choose the tolerances tol_N for stopping the stochastic search algorithm. Choose the sequence $\{p_j\}_{j\geq 0}$ and γ_0. Set the iteration index for $\boldsymbol{\theta}$ and $\{p_j\}_{j\geq 0}$ to $j=0$.

Step S2. Conduct the normal experiment with the current $\boldsymbol{\theta}_j$. Evaluate the o.f. $\tilde{J}(\boldsymbol{\theta}_j)$ with $\boldsymbol{\theta}_j=\boldsymbol{\theta}$ in (17), record the current tracking error $\mathbf{E}_j$, and compute the vector variables $\zeta, \Delta\zeta, \Delta\bar{\zeta}$ as shown in [15].

Step S3. Conduct the first gradient experiment according to the approach given in Section 3 to find the first term in the gradient of the augmented o.f. given in (22), namely $\frac{2}{N}\mathbf{\Gamma}^{\mathrm{T}}\mathbf{T}^{T}\mathbf{E}_j$.

Step S4. Conduct the second gradient experiment in the same way as descried in Section 3. The reversed vector $\boldsymbol{\psi}(\boldsymbol{\theta}_j)$ is used as the reference input applied to the real-world CS to find the second term in the gradient of the augmented o.f. given in (22), namely $\mathbf{S}_{ur}^{T}\boldsymbol{\psi}(\boldsymbol{\theta}_j)$, after which the expression $p_j\{\mathbf{\Gamma}^{T}\mathbf{S}_{ur}^{T}\boldsymbol{\psi}(\boldsymbol{\theta}_j)\}$ is obtained in a straightforward manner because $\mathbf{\Gamma}$ is known.

Step S5. Estimate the gradient using (22), and calculate the next reference input sequence using

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j - \gamma_j est\{\frac{\partial \tilde{J}}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j}\}. \tag{24}$$

Step S6. If the gradient search has converged in terms of the gradient of the augmented o.f. less then a constant, $est\{\frac{\partial \tilde{J}}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j}\} < tol_N$, stop the algorithm. Otherwise, calculate $\mathbf{R}_{j+1} = \boldsymbol{\Gamma}\boldsymbol{\theta}_{j+1}$, set $j = j+1$, and next jump to step *S2*.

5 Neural Network-based Gradient Estimation Mechanism

Each iteration in the algorithm given in the previous section requires a normal experiment with the current parameterized reference input. After the normal experiment, the gradient experiments require running perturbed trajectories in the vicinity of the nominal trajectories. These perturbed trajectories are obtained for perturbed reference inputs with small amplitude signals according to the previous sections and to [13-17]. In order to avoid conducting gradient experiments on the real-world CS, a simulation-based mechanism can be used, with identified models instead of the real-world CS. These models are only valid 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 have scope only within the current iteration.

In order to extend the applicability of this approach to smooth nonlinear systems that can be well approximated by linear systems near some operating points, NN-based models can be used for the identification purposes. Our approach has two advantages. First, the closed-loop CS behaviour is usually of low-pass type; therefore, the models usually have simple dynamics. Second, the numerical differentiation issues which occur in noisy environments will be mitigated by our approach. Linear models could have been used as well for gradient estimation since they can be considered as particular cases of nonlinear ones.

Let the nonlinear dynamic maps from the reference input to the controlled output M_{ry} and from the reference input to the control input M_{ru} are supposed to be characterized by the following nonlinear autoregressive exogenous (NARX) models [16, 17]:

$$
\begin{aligned}
&y(k) = M_{ry}(y(k-1),...,y(k-n_y),r(k-1),...,r(k-n_{ry})),\\
&u(k) = M_{ru}(u(k-1),...,u(k-n_u),r(k-1),...,r(k-n_{ru})).
\end{aligned}
\tag{25}
$$

A more compact representation that takes advantage of the supervector 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}, respectively. Using (15) from the model-free gradient estimation algorithm (given in Section 3) in (22), an estimate of the gradient of the augmented o.f. is expressed as

$$
\begin{aligned}
&est\{\frac{\partial \tilde{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j}\} = \frac{2}{\mu_{\mathbf{Y}}}\boldsymbol{\Gamma}^{\mathrm{T}} rev(\overline{\mathbf{Y}}_{G_j} - \overline{\mathbf{Y}}_j) + p_j\{\frac{1}{\mu_{\mathbf{U}}}\boldsymbol{\Gamma}^{\mathrm{T}} rev(\overline{\mathbf{U}}_{G_j} - \overline{\mathbf{U}}_j)\},\\
&\overline{\mathbf{Y}}_{G_j} = M_{ry}(\mathbf{R}_j + \mu_{\mathbf{Y}} rev(\mathbf{E}_j)), \overline{\mathbf{Y}}_j = M_{ry}(\mathbf{R}_j),\\
&\overline{\mathbf{U}}_{G_j} = M_{ru}(\mathbf{R}_j + \mu_{\mathbf{U}} rev(\boldsymbol{\Psi}_j)), \overline{\mathbf{U}}_j = M_{ru}(\mathbf{R}_j),
\end{aligned}
\tag{26}
$$

where $\boldsymbol{\psi}(\boldsymbol{\theta}_j)$ is defined in (22), and $\mu_{\mathbf{Y}}, \mu_{\mathbf{U}}$ are scaling factors chosen such that the perturbations are of small amplitude with respect to the current iteration reference input. The superposition principle invoked here is expected to work for small amplitude perturbations of the nominal trajectories at the current iteration.

We are using a feed-forward NN architecture that consists of one hidden layer with a hyperbolic tangent activation function and a single linear neuron. The input-output map is [16, 17]

$$
\hat{y}(k+1) = \mathbf{W}^T(k)\boldsymbol{\sigma}(\mathbf{V}(k), \mathbf{x}(k)),
\tag{27}
$$

where $\mathbf{W}^T = [w_0 \quad w_1 \quad ... \quad w_H] \in \mathbf{R}^{H+1}$ is the vector of output layer weights, $\boldsymbol{\sigma}^T = [1 \quad \sigma_1(\mathbf{V}_1^T\mathbf{x}) \quad ... \quad \sigma_H(\mathbf{V}_H^T\mathbf{x})]$ is the vector of hidden layer neurons outputs, with the hyperbolic activation tangent activation functions $\sigma_m(x) = \tanh(x), m = 1...H$, the first term in $\boldsymbol{\sigma}$ corresponds to the bias of the output neuron, and each hidden layer neuron is parameterized by its vector of weights $(\mathbf{V}^m)^T = [v_m^0 \quad v_m^1 \quad ... \quad v_m^{nu}] \in \mathbf{R}^{nu+1}$, $m = 1...H$, which multiplies the input vector $\mathbf{x}^T = [x_0 \quad x_1 \quad ... \quad x_{nu}]$.

Treating the NN as a nonlinear multi input-multi output dynamical system considered in the iteration domain [16, 17]:

$$
\begin{aligned}
&\mathbf{W}_{j+1} = \mathbf{W}_j + \mathbf{u}_j^w,\\
&\mathbf{V}_{j+1}^i = \mathbf{V}_j^i + \mathbf{u}_j^{v^i}, i = 1...H,\\
&\mathbf{Y}_j(k+1) = \mathbf{W}_j^T\boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k)), k = 0...N,
\end{aligned}
\tag{28}
$$

where j is the iteration index, the dynamical system (28) is transformed into a static map from the inputs to the outputs, and the batch training of the NN can be

regarded as a supervised learning approach, that aims the minimization of the tracking error $\mathbf{E}_j = \mathbf{Y}_j - \mathbf{Y}^d$ referred to also as training error.

As shown in [16, 17], the input at each iteration is derived in the framework of norm-optimal ILC as the solution to an OP, is next transformed into another OP by a Taylor series expansion. The optimal vector solution to this OP consists of the increments of the NN weights, expressed as update laws, which actually represent our ILC-based training scheme for NNs. The norm-optimal ILC formulation is more general since the o.f. also includes the regularization term on the weights update, and it offers a degree of freedom in learning.

6 Case Studies and Discussion of the Results

The case studies apply our IRIT algorithm to the controller tuning for a representative mechatronics application, namely the angular positioning of the vertical motion of a twin-rotor aero-dynamical system experimental setup [16]. A rigid beam supports at one end a horizontal rotor which produces vertical motion and at the other end a vertical rotor causing horizontal motion. The horizontal position is considered fixed in this case study. The nonlinear equations that describe the vertical motion are [16]

$$\begin{aligned} &J_v\dot{\Omega}_v = l_m F_v(\omega_v) - \Omega_v k_v + g[(A-B)\cos\alpha_v - C\sin\alpha_v], \\ &\dot{\alpha}_v = \Omega_v, \\ &I_v\dot{\omega}_v = M(U_v) - M_r(\omega_v), \end{aligned} \tag{29}$$

where $U_v(\%) = u$ is the control signal represented by the PWM duty-cycle corresponding to the input voltage range of the DC motor, $-24\,\mathrm{V} \le u \le 24\,\mathrm{V}$, $\omega_v\,(rad/s)$ is the angular speed of the rotor, $\alpha_v\,(rad) = y$ is the process output corresponding to the pitch angle of the beam which supports the main and the tail rotor, $\Omega_v\,(rad/s)$ is the angular velocity of the beam. The expressions of the other parameters and variables related to (29) are given in [16], and the parameter values are also given in [16] as

$$\begin{aligned} &J_v = 0.02421\,kg\,m^2,\ I_v = 4.5\cdot 10^{-5}\ kg\,m^2, k_v = 0.0127\ kg\,m^2/s, \\ &B - A = 0.05\ rad\,kg\,m, l_m = 0.2\,m,\ C = 0.0936\ rad\,kg\,m. \end{aligned} \tag{30}$$

The nonlinear model (50) is not used in the reference input tuning process except for obtaining an initial feedback controller, which can also be obtained by model-free approaches. A discrete-time linear PID controller with the following t.f. is considered:

$$H(q^{-1}) = (0.012 + 0.001q^{-1})/(1-q^{-1}). \tag{31}$$

The reference trajectory is prescribed in terms of the unit step response of a second-order normalized reference model with the t.f. $\omega_n^2/(s^2+2\zeta\omega_n s+\omega_n^2)$ and the parameters $\omega_n = 0.5$ rad/s and $\zeta = 0.7$. The sampling period is $T_s = 0.1$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 $m = 0$.

The initial reference input is chosen such that to obtain a CS response that is very different from the targeted reference trajectory. Therefore, the initial reference input is set as a squared signal of amplitude 0.1, and this motion corresponds to a "take-off" manoeuvre followed by a "landing" manoeuvre. The coefficients of the initial polynomial fit obtained via a least squares algorithm of dimension $h_r + 1 = 8$ are grouped in the parameter vector

$$\boldsymbol{\theta}_0 = [0.05\ \ 0.93\ \ -15.62\ \ 101.36\ \ -307.68\ \ 465.59\ \ -342.52\ \ 97.86]^T. \tag{32}$$

The NN architecture used in the identification and subsequently in the gradient estimation consists of one hidden layer with six neurons and one output layer with one neuron. As shown in Section 5, hyperbolic tangent activation functions are employed in the hidden layer, and a linear function is employed as the output neuron activation function. This NN architecture uses the last two outputs and the last two inputs in order to obtain the output prediction. The same simple architecture is used for both M_{ry} and M_{ru}. The inputs of the two NNs are selected as

$$\begin{aligned} \mathbf{x}_{ry}^T(k) &= [1\ \ y(k)\ \ y(k-1)\ \ r(k)\ \ r(k-1)] \text{ for } M_{ry}, \\ \mathbf{x}_{ru}^T(k) &= [1\ \ u(k)\ \ u(k-1)\ \ r(k)\ \ r(k-1)] \text{ for } M_{ru}. \end{aligned} \tag{33}$$

The outputs of the NNs are the closed-loop output and the control signal, respectively.

The training of the two NN architectures is carried out in our ILC framework. Each neuron in the hidden layer has five parameters, i.e., four weights and one bias. The output layer has seven weights including the bias. We trained the weight vectors $\mathbf{W} \in \mathbf{R}^{7\times 1}$ and $\mathbf{V}_i \in \mathbf{R}^{5\times 1}, i = 1...6$. The initial values of the hidden neurons parameters are chosen from a normal distribution centred at zero with variance 1.

The NN-based identification is carried out on the nominal trajectories of the closed-loop for the initial controller parameters presented in the sequel. Only the results concerning the identified map M_{ry} are presented 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 $\mathbf{I}_\zeta$ is the general notation for ζ^{th} order identity matrix. The evolutions of the training error throughout the iterations and of the simulated trajectory before and after training are shown in Figure 1.

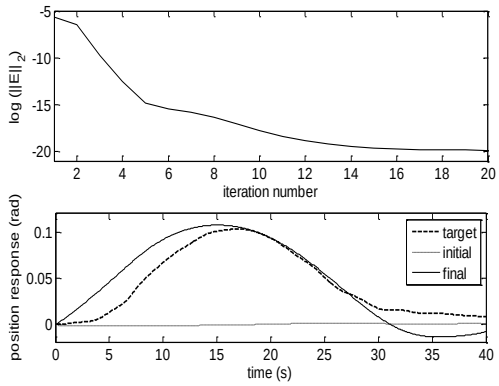

Figure 1

NN training error versus iteration number and controlled output response before and after training

Two simulated case studies are next considered. The first simulated case study deals with the unconstrained optimization, where only the reference input signal is tuned according using our approach in order to ensure the output tracking improvement. A BFGS update was used for the Hessian estimate and the step size was chosen constant equal to 0.12. The final parameter vector that describes the reference input is

$$\boldsymbol{\theta}_{22} = [0.04 \;\; 0.88 \;\; -15.66 \;\; 101.35 \;\; -307.69 \;\; 465.59 \;\; -342.52 \;\; 97.86]^T. \tag{34}$$

Figure 2 gives the initial and final reference input after optimization and the o.f. decrease over the iterations. Only the first five parameters of the parameter vector are changed significantly. The control signal and the final controlled output before and after the optimization are shown in Figure 3 and in Figure 4, respectively. The rise of the reference input on the first 15 s with respect to the initial value and the decrease of the final reference input after 20 s compared to the initial reference input have to be correlated with the output response. This indicates that the reference input is tuned such as to anticipate the low bandwidth of the CS. As an effect, the final controlled output is rising faster under the take-off manoeuvre, and also tracks the reference input more accurately for the landing manoeuvre.

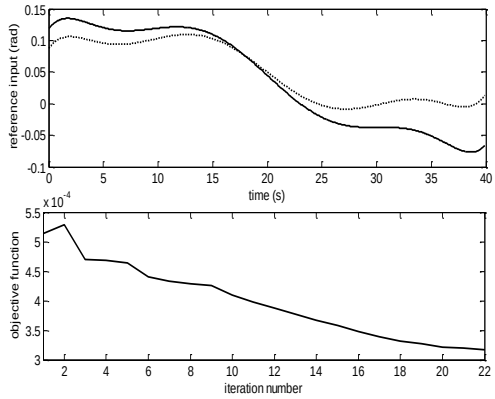

Figure 2

Simulations results expressed as reference input versus time and as o.f. in (5) versus iteration number. The initial reference is dotted and the final reference input is black solid

The second simulated case study corresponding to the optimization with control signal saturation and control signal rate constraints is presented as follows. The two inequality constraints are $-0.05 \le u(k) \le 0.12$ and $-0.01 \le \Delta u(k) \le 0.015$.

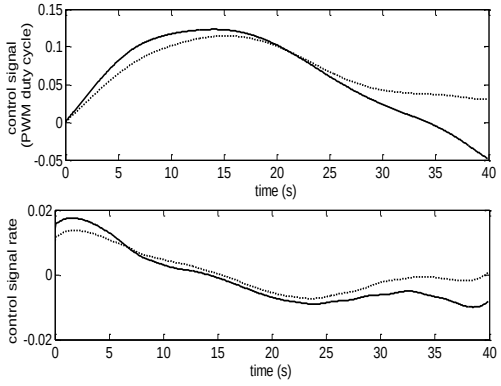

Figure 3

Simulation results for the unconstrained case: control signal responses: initial (dotted) and final (solid)

The algorithm is applied as in the deterministic case as follows. The sequence of penalty parameters in (18) was set to a constant value $p_j = 9$. Two constant values of the step-scaling parameter were used for the gradient descent. When no constraints are violated the step size was set to $\gamma = 0.1$; otherwise, it was set to $\gamma = 0.12$. 400 samples of the reference input are subject to optimization and a total of 796 constraints were used: 798 for control signal saturation and 798 for control signal rate saturation. The final parameter vector that describes the reference input is

$$\boldsymbol{\theta}_{32} = [0.04 \;\; 0.91 \;\; -15.64 \;\; 101.35 \;\; -307.69 \;\; 465.59 \;\; -342.52 \;\; 97.86]^T. \qquad (35)$$

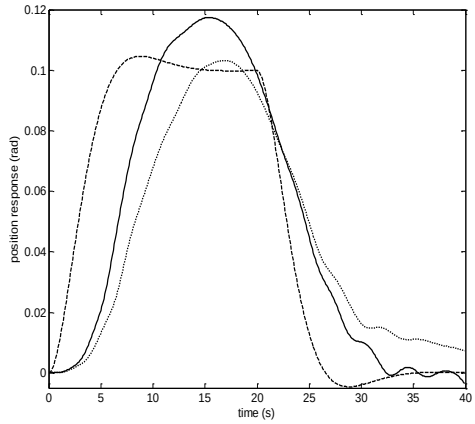

Figure 4

Simulation results for the unconstrained case expressed as position response: initial (black dotted), final response after optimization (black solid) and reference trajectory (black dashed)

The evolution of the reference input during the learning process is presented in Figure 5. The final reference input parameterized by $\boldsymbol{\theta}_{32}$ has higher amplitude for the first 20 sec and forces the take-off motion to respond faster but with higher overshoot. On the other hand, after 20 s, the final reference input drops below the initial reference input trajectory in order to compensate for the slow response in the landing manoeuvre and also to correct the steady-state error.

Figure 5 also shows the sum of penalty functions $\phi(\boldsymbol{\theta})+\Delta\phi(\boldsymbol{\theta})$ that contributes to the augmented o.f. $\tilde{J}_{p_j}(\boldsymbol{\theta})$ in (18) to be optimized. Since the constraints are violated more, they weight more in the o.f., and they eventually provide a more significant contribution to the gradient of the o.f., thus driving the optimization in the direction of bringing the trajectories within the feasible boundaries. This has a negative impact on the reference tracking criterion. Even with the double approximation involved in the linearity assumption and in the NN-based gradient estimation, the o.f. decreases as shown in Figure 5 and the performance improvements are evident.

The results given in Figure 5 have to be correlated with the control signal responses illustrated in Figure 6. There are several control signal trajectories during the learning process that violate the constraints more heavily. But, our IRIT algorithm brings the trajectories as much as possible close to the boundaries of the feasible region in such situations.

The output trajectory evolution during the learning is presented in Figure 7, where the final output trajectory is closer to the reference trajectory when compared with the initial response. The corresponding final reference input overcomes the difficulty of both the take-off manoeuvre and the landing manoeuvre by anticipating the slow responses.

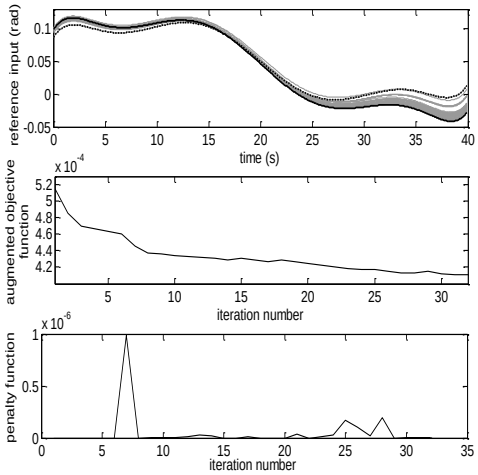

Figure 5

Simulation results expressed as reference input versus time as the learning converges, augmented o.f. (18) versus iteration number and sum of penalty functions. The initial reference is dotted and the final reference input is black solid

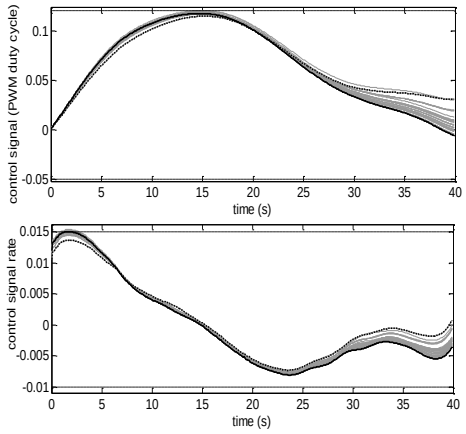

Figure 6

Simulation results for the constrained case expressed as control signal responses: initial (dotted) and final (solid). The constraints are dashed

The advantage of the proposed approach is even more obvious because it can force the controlled output in the vicinity of the reference trajectory; the parameter tuning of the controller can next be carried out using similar model-free iterative techniques in linear or nonlinear formulations such as IFT [17]. Our approach prevents the optimization to get stuck in local minima that are far from the global minimum. Proceeding this way the windsurfing approach is actually solved using our algorithm in a two-degrees-of-freedom tuning setting. Therefore, the CS can be optimized in order to exhibit highly complex manoeuvres.

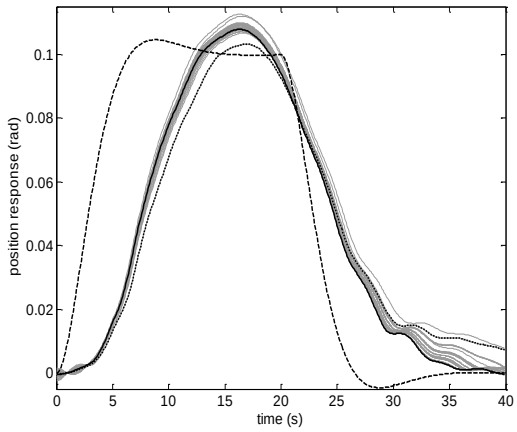

Figure 7

Simulation results for the constrained case expressed as position response: initial (black dotted), final response after optimization (black solid) and reference trajectory (black dashed) and intermediate trajectories (grey)

The optimization of the reference input sequence in the parameterized form leads to less spectacular results then in the case when the non-parameterized reference input is subjected to optimization [13-17]. However, the parameterization effect is that it reduces the many local minima specific to the case when no parameterization is used. It seems that the polynomial approximation employed for the reduction of the reference input dimension has limited potential and other basis functions may be exploited for this purpose such as radial basis functions.

The convergence speed to the solution within few experiments on the real-world process depends on how many constraints are violated at each iteration. This depends on the interplay between the penalty parameter p_j and the step size of the search algorithm.

The discussion presented in this section can be extended because different results will be obtained for other o.f.s and other constraints in the OPs. This depends on the performance specifications and objectives [29-34] for various CS applications [35-40].

The training approach using the ILC framework described in the paper can be extended to other NN architectures. Moreover, it can be employed for the same architecture that is used in this paper, with more than one hidden layer.

Conclusions

This paper has proposed a data-driven algorithm, which solves an optimal control problem in order to ensure the constrained reference trajectory tracking by few experiments conducted on the real-world CS. The new IRIT algorithm has three advantages. First, the closed-loop CS stability is not affected while solving the

trajectory tracking problem. Starting with a given closed-loop controller, the stability is maintained along the iterations of the IRIT algorithm and no additional tests are needed. The change in the controller parameters specific to other tuning techniques including IFT, which usually require attention in order to achieve the bumpless transfer between controllers, is mitigated by our approach. Second, cost-effective controller designs and implementations are achieved because of the linear parameterization that ensures the reduced dimensionality of the reference input vector. Third, our IRIT algorithm is advantageous as it works for smooth nonlinear systems around some operating points

The proposed algorithm can be generalized by considering other data-driven optimization approaches to controller tuning combined with ILC to optimize the reference input sequence. These techniques can yield automated tools for controller design and tuning, with benefits for the CS designers.

Acknowledgement

This work was supported in part by a grant from the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, 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 Program Human Resources Development 2007-2013, and from the NSERC of Canada.

References

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

[2] Z.-S. Hou, Z. Wang: From Model-Based Control to Data-Driven Control: Survey, Classification and Perspective, Information Sciences, Vol. 235, 2013, pp. 3-35

[3] S. Gunnarsson, M. Norrlöf: On The Design of ILC Algorithms Using Optimization, Automatica, Vol. 37, No. 12, 2001, pp. 2011-2016

[4] D. H. Owens, J. Hätönen: Iterative Learning Control – An Optimization Paradigm, Annual Reviews in Control, Vol. 29, No. 1, 2005, pp. 57-70

[5] M. Norrlöf, S. Gunnarsson: Time and Frequency Domain Convergence Properties in Iterative Learning Control, International Journal of Control, Vol. 75, No. 14, 2002, pp. 1114-1126

[6] M. Butcher, A. Karimi, R. Longchamp: Iterative Learning Control Based on Stochastic Approximation, Proceedings of 17th IFAC World Congress, Seoul, Korea, 2008, pp. 1478-1483

[7] H.-F. Chen, H.-T. Fang: Output Tracking for Nonlinear Stochastic Systems by Iterative Learning Control, IEEE Transactions on Automatic Control, Vol. 49, No. 4, 2004, pp. 583-588

[8] S. Mishra, U. Topcu, M. Tomizuka: Optimization-based Constrained Iterative Learning Control, IEEE Transactions on Control Systems Technology, Vol. 19, No. 6, 2011, pp. 1613-1621

[9] P. Janseens, G. Pipeleers, J. Swevers: A Data-Driven Constrained Norm-Optimal Iterative Learning Control Framework for LTI Systems, IEEE Transactions on Control Systems Technology, Vol. 21, No. 2, 2013, pp. 546-551

[10] S. Lupashin, A. Schöllig, M. Sherback, R. D'Andrea: A Simple Learning Strategy for High-Speed Quadrocopter Multi-Flips, Proceedings of 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 2010, pp. 1642-1648

[11] J. Z. Kolter, A. Y. Ng: Policy Search via the Signed Derivative, in Robotics: Science and Systems V, J. Trinkle, Y. Matsuoka, J. A. Castellanos, Eds., The MIT Press, Cambridge, MA, USA, 2009, pp. 1-8

[12] Z. Xiong, J. Zhang: A Batch-to-Batch Iterative Optimal Control Strategy Based on Recurrent Neural Network Models, Journal of Process Control, Vol. 15, No. 1, 2005, pp. 11-21

[13] M.-B. Radac, R.-E. Precup, E. M. Petriu, S. Preitl, C.-A. Dragos: Experiment-Based Approach to Reference Trajectory Tracking, Proceedings of 2012 IEEE Multi-Conference on Systems and Control, Dubrovnik, Croatia, 2012, pp. 470-475

[14] M.-B. Radac, R.-E. Precup, E. M. Petriu, S. Preitl, C.-A. Dragos: Data-Driven Reference Trajectory Tracking Algorithm and Experimental Validation, IEEE Transactions on Industrial Informatics, Vol. 9, No. 4, 2013, pp. 2327-2336

[15] M.-B. Radac, R.-E. Precup, E. M. Petriu: Design and Testing of a Constrained Data-Driven Iterative Reference Input Tuning Algorithm, Proceedings of 13th European Control Conference, Strasbourg, France, 2014, pp. 2034-2039

[16] M.-B. Radac, R.-E. Precup, E. M. Petriu, S. Preitl: Iterative Data-Driven Tuning of Controllers for Nonlinear Systems with Constraints, IEEE Transactions on Industrial Electronics, Vol. 61, No. 11, 2014, pp. 6360-6368

[17] M.-B. Radac, R.-E. Precup, E. M. Petriu, S. Preitl: Iterative Data-Driven Controller Tuning with Actuator Constraints and Reduced Sensitivity, Journal of Aerospace Information Systems, Vol. 11, No. 9, 2014, pp. 551-564

[18] C. T. Freeman, Y. Tan: Iterative Learning Control with Mixed Constraints for Point-to-Point Tracking, IEEE Transactions on Control Systems Technology, Vol. 21, No. 3, 2012, pp. 604-616

[19] M. Volckaert, M. Diehl, J. Swevers: Generalization of Norm Optimal ILC for Nonlinear Systems with Constraints, Mechanical Systems and Signal Processing, Vol. 39, No. 1-2, 2013, pp. 280-296

[20] J. X. Xu, Y. Chen, T. H. Lee, S. Yamamoto: Terminal Iterative Learning Control with an Application to RTPCVD Thickness Control, Automatica, Vol. 35, No. 9, 1999, pp. 1535-1542

[21] J. van de Wijdeven, O. Bosgra: Using Basis Functions in Iterative Learning Control: Analysis and Design Theory, International Journal of Control, Vol. 83, No. 4, 2010, pp. 661-675

[22] G. Pipeleers, K. L. Moore: Reduced-Order Iterative Learning Control and a Design Strategy for Optimal Performance Tradeoffs, IEEE Transactions on Automatic Control, Vol. 57, No. 9, 2012, pp. 2390-2395

[23] J. Boolder, B. Lemmen, S. Koekebakker, T. Oomen, O. Bosgra, M. Steinbuch: Iterative Learning Control with Basis Functions for Media Positioning in Scanning Inkjet Printers, Proceedings of 2012 IEEE Multi-Conference on Systems and Control, Dubrovnik, Croatia, 2012, pp. 1255-1260

[24] M. Heertjes, D. Hennekens, M. Steinbuch: MIMO Feed-Forward Design in Wafer Scanners Using a Gradient Approximation-Based Algorithm, Control Engineering Practice, Vol. 18, No. 5, 2010, pp. 495-506

[25] M. J. C. Ronde, G. A. L. Leenknegt, M. J. G. van de Molengraft, M. Steinbuch: Data-Based Spatial Feedforward for Over-Actuated Motion Systems, Mechatronics, Vol. 24, No. 6, 2014, pp. 679-690

[26] J. Sjöberg, F. De Bruyne, M. Agarwal, B. D. O. Anderson, M. Gevers, F. J. Kraus, N. Linard: Iterative Controller Optimization for Nonlinear Systems, Control Engineering Practice, Vol. 11, No. 9, 2003, pp. 1079-1086

[27] I.-J. Wang, J. C. Spall: Stochastic Optimization with Inequality Constraints Using Simultaneous Perturbations and Penalty Functions, International Journal of Control, Vol. 81, No. 8, 2008, pp. 1232-1238

[28] Y. He, M. C. Fu, S. I. Marcus: Convergence of Simultaneous Perturbation Stochastic Approximation for Nondifferentiable Optimization, IEEE Transactions on Automatic Control, Vol. 48, No. 8, 2003, pp. 1459-1463

[29] S. Preitl, R.-E. Precup, J. Fodor, B. Bede: Iterative Feedback Tuning in Fuzzy Control Systems. Theory and Applications, Acta Polytechnica Hungarica, Vol. 3, No. 3, 2006, pp. 81-96

[30] F.-G. Filip, K. Leiviskä: Large-Scale Complex Systems, in Springer Handbook of Automation, S. Y. Nof, Ed., Springer-Verlag, Berlin, Heidelberg, 2009, pp. 619-638

[31] J. Vaščák, K. Hirota: Integrated Decision-Making System for Robot Soccer, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 15, No. 2, 2011, pp. 156-163

[32] R.-E. Precup, S. Preitl, M.-B. Radac, E. M. Petriu, C.-A. Dragos, J. K. Tar: Experiment-Based Teaching in Advanced Control Engineering, IEEE Transactions on Education, Vol. 54, No. 3, 2011, pp. 345-355

[33] M. Bošnak, D. Matko, S. Blažič: Quadrocopter Hovering Using Position-Estimation Information From Inertial Sensors and a High-Delay Video System, Journal of Intelligent & Robotic Systems, Vol. 67, No. 1, 2012, pp. 43-60

[34] D. Rojas, G. Millan, F. Passold, R. Osorio, C. Cubillos, G. Lefranc: Algorithms for Maps Construction and Localization in a Mobile Robot, Studies in Informatics and Control, Vol. 23, No. 2, 2014, pp. 189-196

[35] R.-E. Precup, S. Preitl: Stability and Sensitivity Analysis of Fuzzy Control Systems. Mechatronics Applications, Acta Polytechnica Hungarica, Vol. 3, No. 1, 2006, pp. 61-76

[36] Z. C. Johanyák: Fuzzy Modeling of Thermoplastic Composites' Melt Volume Rate, Computing and Informatics, Vol. 32, No. 4, 2013, pp. 845-857

[37] K. Lamár, J. Neszveda: Average Probability of Failure of Aperiodically Operated Devices, Acta Polytechnica Hungarica, Vol. 10, No. 8, 2013, pp. 153-167

[38] H.-N. Teodorescu: On the Characteristic Functions of Fuzzy Systems, International Journal of Computers Communications & Control, Vol. 8, No. 3, 2013, pp. 469-476

[39] D. Yazdani, B. Nasiri, R. Azizi, A. Sepas-Moghaddam, M. R. Meybodi: Optimization in Dynamic Environments Utilizing a Novel Method Based on Particle Swarm Optimization, International Journal of Artificial Intelligence, Vol. 11, No. A13, 2013, pp. 170-192

[40] E. Osaba, F. Diaz, E. Onieva, R. Carballedo, A. Perallos: AMCPA: A Population Metaheuristic With Adaptive Crossover Probability and Multi-Crossover Mechanism for Solving Combinatorial Optimization Problems, International Journal of Artificial Intelligence, Vol. 12, No. 2, 2014, pp. 1-23