

Iterative Data-Driven Controller Tuning with Actuator Constraints and Reduced Sensitivity

Mircea-Bogdan Radac¹ and Radu-Emil Precup²
Politehnica University of Timisoara, RO-300223 Timisoara, Romania

Emil M. Petriu³
University of Ottawa, Ottawa, ON, K1N 6N5 Canada

and
Stefan Preitl⁴
Politehnica University of Timisoara, RO-300223 Timisoara, Romania

This paper proposes a novel iterative data-driven algorithm (IDDA) for the data-driven tuning of controllers for nonlinear systems. The IDDA uses an experiment-based solving of the optimization problems for nonlinear processes with linear controllers accounting for actuator constraints in terms of a quadratic penalty function approach. A Neural Network-based identification provides the gradient information used in the search algorithm for controller tuning and ensures a reduced sensitivity with respect to the controller parameters. A case study dealing with the data-driven controller tuning for the angular position control of a nonlinear aerodynamic system is included to validate the new IDDA.

Nomenclature

F_v (N)	=	amplitude of oscillation
g (m/s ²)	=	gravitational acceleration
I_v (kg m ²)	=	moment of inertia of the rotor
J_v (kg m ²)	=	moment of inertia of the beam
k_v (N m s)	=	vertical angular momentum
l_m (m)	=	effective arm of the aerodynamic force F_v

¹ Assistant Lecturer, Department of Automation and Applied Informatics, Bd. V. Parvan 2, mircea.radac@aut.upt.ro.

² Corresponding author, Professor, Department of Automation and Applied Informatics, Bd. V. Parvan 2, radu.precup@aut.upt.ro.

³ Professor, School of Electrical Engineering and Computer Science, 800 King Edward, petriu@uottawa.ca.

⁴ Professor, Department of Automation and Applied Informatics, Bd. V. Parvan 2, stefan.preitl@aut.upt.ro.

M (N m) = active torque

M_{v1} (N m) = return torque due to the gravitational forces

M_r (N m) = reactive friction torque

U_v (%) = Pulse Width Modulated duty cycle for the DC motor

α_v (rad) = pitch angle of the beam

ω_v (rad/s) = angular speed of the rotor

Ω_v (rad/s) = angular velocity of the beam

I. Introduction

DATA-DRIVEN optimization for controller tuning has the advantage of requiring no *a priori* model information about the controlled processes, or only a reduced amount of information obtained from process measurements [1]–[10]. The easy interpretable performance indices are aggregated in cost functions (c.f.'s). The minimization of the c.f.'s in the framework of constrained optimization problems can fulfill different objectives such as reference trajectory tracking (including model reference tracking), control signal penalty, disturbance rejection, etc. These data-driven optimization techniques compensate for the impact of the model errors and uncertainties.

The main data-driven technique used for the iterative data-driven controller tuning by an experiment-based update of controller parameters is the Iterative Feedback Tuning (IFT) [1]. IFT requires only a few experiments conducted on the real-world control systems (CS) in order to estimate the c.f. gradients used for the iterative solving of the optimization problems. However, this tuning does not guarantee robust stability or performance because it lacks the quantitative knowledge about the controlled process.

Other popular data-driven techniques used for mechanical, aerospace and several complex applications are Simultaneous Perturbation Stochastic Approximation (SPSA) [2], Reinforcement Learning (RL) [11], approximate dynamic programming [12], model-free adaptive control [10], [13], [14], and data-driven monitoring [15]. These techniques also use Neural Networks (NNs) with supervised and unsupervised learning [16]–[20]. Predictive, adaptive fuzzy control [21]–[24], low-cost fuzzy control [25], [26] can also be used in this context by several fuzzy model transformations [25], [27]–[30].

The data-driven techniques are associated with appropriately defined optimization problems. Such optimization problems involve adaptive and predictive control [31], [32], learning, planning and optimal control [33], [34], and nature inspired optimization methods [35]–[38]. Defining the various c.f.'s involved in these applications may be

challenging as performance improvement should be targeted using few experiments and little information on the controlled process.

Building upon recent results on data-driven techniques for the Iterative Learning Control (ILC) [8], the main contribution of this paper is the development of a new iterative data-driven algorithm (IDDA) for optimal controller tuning accounting for the operational constraints on the control signal. Other contributions are the NN-based identification for gradient estimation supported by a novel ILC-based mechanism for NN training and the data-driven tuning that offers a reduced sensitivity with respect to the controller parameters. The proposed algorithm is general and applicable to nonlinear systems. It uses an experiment-based quadratic penalty function approach. The main advantages of IDDA are

- being based on learning from experimental data from real-world CS, it can compensate for process model nonlinearities and parametric uncertainties.
- it requires a small number of experiments, which results in cost-effective implementations.

Our supervised learning approach used for the CS exploits only the nonlinear structure of the problem and no specific process model. It represents a tradeoff between the exploration of the environment, obtained by running input-output closed-loop CS trajectories, and the use of approximate environment models, which are merely approximations of the real-world CS. A similar, but not identical technique, using a RL approach is presented in [39]; this approach suggests gradient-based control policy improvements based on first principles models of the process whereas in our approach this is carried out using ultra-local models of the CS obtained via NNs. Another gradient learning approach based on first principles models of the reference input is suggested in [40], where an open-loop control strategy is used for motion primitives that allow for acrobatic flips for quadcopters.

The paper is organized as follows: Section II gives the formulation of the iterative controller tuning problem for nonlinear processes in the framework of optimal control with focus on the state-of-the-art, and Section III treats the gradient estimation needed in the search algorithm using nonlinear models identified by NNs. Some aspects concerning the sensitivity of the CS with respect to the controller parameters are discussed. A model-free mechanism for dealing with constraints in the optimization problem using quadratic penalty functions is offered. Section IV summarizes the IDDA algorithm. Section V discusses the application of the controller tuning approach to a representative case study which deals with the angular position control of a nonlinear aerodynamic system. The conclusions are then presented in Section VI.

II. Problem Statement

We will consider a Single Input-Single Output (SISO) discrete-time control system (CS) described by the nonlinear process and controller equations:

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

where y is the process output, u is the control signal, r is the reference input, v is the zero-mean stochastic disturbance at the output accounting for a large class of disturbances, and $\boldsymbol{\rho}$, $\boldsymbol{\rho} \in \mathbf{R}^{n_p}$, is the parameter vector of the controller. Due to the nonlinear functions P and C the model (1) belongs to the class of nonlinear autoregressive exogenous (NARX) models.

As shown in [41], several assumptions are made in relation with this CS model (1). The closed-loop CS is stable and the nonlinear operators P , C are smooth functions of their arguments. The nominal CS trajectory is denoted as $\{r_n(k), u_n(k), y_n(k)\}$, $k = 0 \dots N$, where N is the length of the experiment. The notations for the changes around the nominal trajectories are $\delta r(k) = r(k) - r_n(k)$ for the reference input, $\delta u(k) = u(k) - u_n(k)$ for the control signal, and $\delta y(k) = y(k) - y_n(k)$ for the process output.

One major objective in the iterative controller tuning is to search for the controller parameters that solve the following optimization problem starting with the initial solution $\boldsymbol{\rho}_0$:

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

subject to system dynamics (1) and to operational constraints, where D_s is the stability domain of the parameter vectors $\boldsymbol{\rho}$ which ensure a stable CS [42]. The constraints can usually be formulated as inequality constraints imposed to $u(k)$ and $y(k)$, and to their rates with respect to time, $\Delta u(k) = u(k) - u(k-1)$ and $\Delta y(k) = y(k) - y(k-1)$. These constraints can be imposed to other variables as well [43]–[46], and they depend on the specific controlled processes that can include aerospace applications such as unmanned aerial vehicles and helicopters [47], [48]. These constraints imposed on $u(k)$ and $\Delta u(k)$, related to the actuators, will be used in the controller tuning algorithms having beneficial effects on the overall CS performance.

The formulation of the c.f. in (2) targets the trajectory tracking of the desired system output y^d while the control effort is penalized by the weighting parameter $\lambda \geq 0$, and the expectation $E\{\dots\}$ is taken with respect to the stochastic disturbance v . The optimization problem (2) in the unconstrained case is usually solved employing the recursive stochastic search algorithm:

$$\boldsymbol{\rho}_{j+1} = \boldsymbol{\rho}_j - \gamma_j \mathbf{R}_j^{-1} \text{est} \left\{ \left. \frac{\partial J}{\partial \boldsymbol{\rho}} \right|_{\boldsymbol{\rho}=\boldsymbol{\rho}_j} \right\}, \quad (3)$$

where the search information is provided by the estimation of the c.f. gradient J with respect to the controller parameters. This can be done by using, for example, the second order information provided by the Gauss-Newton approximation of the Hessian matrix \mathbf{R}_j , or simply by using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) update algorithm as an estimate for the Hessian. The subscript j , $j \in \mathbf{N}$, indicates the current iteration number, and γ_j , $\gamma_j > 0$, indicates the step size.

The main feature of IFT [1] is that the gradient information is extracted using special experiments conducted on the closed-loop CS. These experiments avoid the need for a process model but, at the same time, they 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 the matrix \mathbf{R}_j . Although the linearity is assumed, a nonlinear-based procedure is also feasible according to [41]. The gradients can be estimated, as shown in [49], not by finite difference approximations with perturbed $\boldsymbol{\rho}$ but by using modified reference trajectories for small changes $\delta r(k)$, $\delta u(k)$ and $\delta y(k)$ around the nominal trajectories.

The advantage of this approach is twofold. First, the closed-loop CS is not changed for the special purpose of obtaining the gradient estimate and the CS stability is not affected. Second, the experiments are carried out in the close vicinity of the nominal trajectories. The motivation for this is presented as follows using elements from [41]. Let the deviations around the nominal trajectories be expressed as the first order Taylor series expansion

$$\begin{aligned}
 u(k) &= u_n(k) + \sum_{i=1}^{n_{u2}} \left. \frac{\partial C}{\partial u(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta u(k-i) + \sum_{i=0}^{n_{y2}} \left. \frac{\partial C}{\partial y(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta y(k-i) \\
 &+ \sum_{i=0}^{n_r} \left. \frac{\partial C}{\partial r(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta r(k-i) + \sum_{h=1}^{n_p} \left. \frac{\partial C}{\partial \rho_h} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta \rho_h + h.o.t., \\
 y(k) &= y_n(k) + \sum_{i=1}^{n_{y1}} \left. \frac{\partial P}{\partial u(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k)\}} \delta u(k-i) + \sum_{i=0}^{n_{y1}} \left. \frac{\partial P}{\partial y(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k)\}} \delta y(k-i) + h.o.t.
 \end{aligned} \tag{4}$$

It is assumed that a change in $\{u_n(k)\}$ is generated around the nominal trajectories $\{r_n(k), u_n(k), y_n(k)\}$, $k = 0 \dots N$, first by modifying only the controller parameter ρ_h , $h = 1 \dots n_p$, and then by changing the nominal reference trajectory $\{r_n(k)\}$. If there exists a reference input change $\{\delta r(k)\}$ that generates the same amount of change in $\{u_n(k)\}$, then the same amount of change in $\{y_n(k)\}$ occurs, namely $\{\delta y(k)\}$. The higher order terms referred to as *h.o.t.* in (4) are considered negligible. Imposing the equality to the last two terms in the expression of $u(k)$ in (4) leads to

$$\left. \frac{\partial C}{\partial \rho_h} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta \rho_h = \sum_{i=0}^{n_r} \left. \frac{\partial C}{\partial r(k-i)} \right|_{\{r_n(k), u_n(k), y_n(k), \mathbf{p}\}} \delta r(k-i), \tag{5}$$

from which the sequence $\{\delta r(k)\}$ can be obtained. For a linear time invariant controller it follows that $\delta r(k) = S(k) \delta \rho_h$, where $S(k)$ is a signal which depends on the nominal trajectories and on the nominal controller parameters [49].

These considerations suggest that conducting experiments with different perturbed reference inputs around the nominal trajectory leads to the same effect in the change of $\{u_n(k)\}$ and $\{y_n(k)\}$ as when a single controller parameter is modified. The gradients of $u(k)$ and $y(k)$ with respect to the change in \mathbf{p} can then be estimated as shown in [41] and [49].

While this paper considers only the case of linear controllers, an extension to nonlinear controllers is also possible. We focus on the linear controllers because many CSs actually use the well understood PI or PID controllers for a large variety of applications [50]–[53].

Two issues have been addressed in the literature: (i) the number of gradient experiments, which can be prohibitive when number of parameters is increasing [41], [54], and (ii) the constrained approach [55], [56]. This

paper will show that the nonlinear tuning accounting for operational constraints gives good results, and it is also efficient as it requires a relatively small number of iterations and experiments.

III. Data-driven Estimation of Gradients Using Neural Networks

A. Gradient Estimation Using Neural Networks

Artificial NNs, which are well known universal approximators for dynamic systems, [41], [57], are well suited for gradient estimation. Each time the gradient information is needed, the nonlinear reference input/process output (r/y) map and the nonlinear reference input/control signal (r/u) map can be both identified using data collected during a normal experiment in which the c.f. is evaluated. Let these two maps r/y and r/u be

$$y(k) = M_{ry}(y(k-1), \dots, y(k-n_y), r(k-1), \dots, r(k-n_{ry})) \quad (6)$$

and

$$u(k) = M_{ru}(u(k-1), \dots, u(k-n_u), r(k-1), \dots, r(k-n_{ru})), \quad (7)$$

respectively.

The variables $\partial y / \partial \rho_h$ and $\partial u / \partial \rho_h$ can be estimated by finite difference approximations as

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial \rho_h} &= \frac{\bar{y}(k, r_n + \mu_h \delta r_h) - \bar{y}(k, r_n)}{\mu_h \delta \rho_h}, \\ \frac{\partial \hat{u}(k)}{\partial \rho_h} &= \frac{\bar{u}(k, r_n + \mu_h \delta r_h) - \bar{u}(k, r_n)}{\mu_h \delta \rho_h}, \quad h = 1 \dots n_\rho, k = 0 \dots N, \end{aligned} \quad (8)$$

where $\delta \rho_h = 1$ is considered, and the numerators are equivalent to carrying out two simulations, namely one with the nominal controller parameter vector ρ and another one with the h^{th} controller parameter perturbed with $\mu_h \delta \rho_h$. 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 perturbed trajectories with respect to $\{r_n(k)\}$ are calculated using (4) and (5). The variables \bar{y} and \bar{u} are obtained by filtering the nominal and the perturbed reference trajectories through the nonlinear functions M_{ry} and M_{ru} , respectively.

Equations (6) to (8) indicate that the gradients with respect to the controller parameters changes are obtained by changing the reference trajectory. This approach is applicable to both linear and nonlinear systems. The closed-loop

operation of the CS is also an advantage. It should be mentioned that if estimation error terms, such as the bias, occur in the identified models, they are canceled by the difference operation in (8).

This approach offers three notable advantages. First, as the simulation with a perturbed reference is conducted in the vicinity of the nominal trajectory for which the NN is trained, this allows using simple NN architectures with few neurons. While the resulting models will only be valid near the nominal trajectories, and not in a wide operating range, they will nevertheless help reducing the time needed for the experiment. Second, the numerical differentiation issues that occur in noisy environments will be mitigated because the obtained trajectories are not affected by the noisy data involved in NN training. Third, the closed-loop system has a typical low-pass filter behavior that also simplifies the identification.

B. Neural Network Gradient Estimation Using Iterative Learning Control

We are using a simple feed-forward NN architecture consisting of one hidden layer with hyperbolic tangent activation functions and a single output layer neuron with a linear activation function. The corresponding input-output map of this NN is

$$\hat{y}(k+1) = \mathbf{W}^T(k) \boldsymbol{\sigma}(\mathbf{V}(k), \mathbf{x}(k)), \quad (9)$$

where $\mathbf{W}^T = [w_0 \ w_1 \ \dots \ w_n] \in \mathbf{R}^{H+1}$ is the vector of the output layer weights, $\boldsymbol{\sigma}^T = [1 \ \sigma_1(\mathbf{V}_1^T \mathbf{x}) \ \dots \ \sigma_H(\mathbf{V}_H^T \mathbf{x})]$ is the vector of the hidden layer neurons outputs having hyperbolic activation tangent activation functions $\sigma_m(x) = \tanh(x)$, $m = 1 \dots H$, and the superscript T indicates the matrix transposition. The first term in $\boldsymbol{\sigma}$ corresponds to the bias of the output neuron. Each hidden layer neuron is parameterized by its vector of weights $(\mathbf{V}^m)^T = [v_m^0 \ v_m^1 \ \dots \ v_m^{nu}] \in \mathbf{R}^{nu+1}$, $m = 1 \dots H$, which multiplies the input vector $\mathbf{x}^T = [x_0 \ x_1 \ \dots \ x_{nu}]$. Each vector \mathbf{V}^m includes the weight v_m^0 of the bias of m^{th} neuron. Here $nu + 1$ is the number of inputs to the network, and H is the number of hidden layer neurons. The time domain index is $k = 0 \dots N$.

This NN is further treated as an iterative multiple-input and multiple-output (MIMO) nonlinear dynamic system:

$$\begin{aligned} \mathbf{W}_{j+1} &= \mathbf{W}_j + \mathbf{u}_j^w, \\ \mathbf{V}_{j+1}^i &= \mathbf{V}_j^i + \mathbf{u}_j^{v^i}, i = 1 \dots H, \\ \mathbf{Y}_j(k+1) &= \mathbf{W}_j^T \boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k)), k = 0 \dots N, \end{aligned} \quad (10)$$

where

$$\begin{aligned}
 \mathbf{u}_j^w &= [\mathbf{u}_j^{w0} \quad \dots \quad \mathbf{u}_j^{wH}]^T \in \mathbf{R}^{H+1}, \\
 \mathbf{u}_j^{v^i} &= [\mathbf{u}_j^{v^i0} \quad \dots \quad \mathbf{u}_j^{v^imu}]^T \in \mathbf{R}^{nu+1}, \\
 \mathbf{Y}_j &= [y_j(1) \quad \dots \quad y_j(N+1)]^T \in \mathbf{R}^{N+1}, \\
 \mathbf{X}_j &= [\mathbf{x}_j^T(0) \quad \dots \quad \mathbf{x}_j^T(N)]^T \in \mathbf{R}^{(N+1)(nu+1)},
 \end{aligned} \tag{11}$$

where j is the iteration index, $\mathbf{u}_j^w, \mathbf{u}_j^{v^i}$ are the input vectors, and the previously defined weight vectors $\mathbf{W}_j, \mathbf{V}_j^i$ are considered as the state vectors of the dynamic system. The \mathbf{X}_j vector can be regarded as a time-series disturbance which is the same at each iteration, but it can be also regarded as a time-varying parameter vector of the nonlinear system (10). The vector \mathbf{Y}_j is the output of the nonlinear dynamic system (10).

An ILC-based supervised learning (training) algorithm is further used to minimize the tracking error $\mathbf{E}_j = \mathbf{Y}_j - \mathbf{Y}_d$ between the actual NN output and a desired output $\mathbf{Y}_d = [y_d(1) \quad \dots \quad y_d(N+1)]^T \in \mathbf{R}^{N+1}$, where $y_d(1) \dots y_d(N+1)$ are the desired process outputs at the time moments $1 \dots N+1$. The input at each iteration can be derived in the framework of norm-optimal ILC as the solution to the optimization problem

$$(\mathbf{u}_j^{w*}, \mathbf{u}_j^{v^i*}) = \arg \min_{\mathbf{u}_j^w, \mathbf{u}_j^{v^i}} \left\| \mathbf{E}_{j+1}^T \mathbf{R} \mathbf{E}_{j+1} + \mathbf{U}_j^T \mathbf{Q} \mathbf{U}_j \right\|_2^2, \tag{12}$$

where $\mathbf{U}_j = [(\mathbf{u}_j^w)^T \quad (\mathbf{u}_j^{v^1})^T \quad \dots \quad (\mathbf{u}_j^{v^H})^T]^T \in \mathbf{R}^{H+1+H(nu+1)}$ is the stacked vector of inputs, $\mathbf{R} = \mathbf{R}^T \succ 0$ and $\mathbf{Q} = \mathbf{Q}^T \succ 0$ are symmetric positive definite diagonal matrices, $\mathbf{E}_{j+1} = \mathbf{Y}_{j+1} - \mathbf{Y}_d$ is the tracking error for the iteration $j+1$, and $\|\bullet\|_2$ is the general notation for the Euclidean norm of the vector \bullet . The penalty \mathbf{Q} on the vector \mathbf{U}_j is used in order to prevent over-fitting.

A nonlinear least squares method is applied in order to obtain the analytical solution to the optimization problem (12). The linearization of $y_{j+1}(k+1) = \mathbf{W}_{j+1}^T \boldsymbol{\sigma}(\mathbf{V}_{j+1}^i, \mathbf{x}(k)), k=0 \dots N$, is carried out around $\mathbf{W}_j, \mathbf{V}_j^i$ for small variations of $\mathbf{u}_j^w, \mathbf{u}_j^{v^i}$ by considering the output as a nonlinear function of the weight vectors $y_{j+1}(k+1) = f(\mathbf{W}_{j+1}, \mathbf{V}_{j+1}^i, \mathbf{x}(k)), k=0 \dots N$, and the input vector $\mathbf{x}(k)$ as a parameter vector. The Taylor series expansion yields

$$y_{j+1}(k+1) = \mathbf{W}_j^T \boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k)) + [1 \quad \tanh(\mathbf{V}_j^{1T} \mathbf{x}(k)) \quad \dots \quad \tanh(\mathbf{V}_j^{HT} \mathbf{x}(k))] \mathbf{u}_j^w + w_j^1 \frac{4}{(e^{\mathbf{V}_j^{1T} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{1T} \mathbf{x}(k)})^2} \mathbf{x}^T(k) \mathbf{u}_j^{v^1} + \dots + w_j^H \frac{4}{(e^{\mathbf{V}_j^{HT} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{HT} \mathbf{x}(k)})^2} \mathbf{x}^T(k) \mathbf{u}_j^{v^H} + h.o.t. \quad (13)$$

Neglecting the higher order terms in (13) we obtain:

$$y_{j+1}(k+1) = y_j(k+1) + \boldsymbol{\sigma}_j^T(\mathbf{x}(k)) \mathbf{u}_j^w + w_j^1 g_1(k) \mathbf{x}^T(k) \mathbf{u}_j^{v^1} + \dots + w_j^H g_H(k) \mathbf{x}^T(k) \mathbf{u}_j^{v^H}. \quad (14)$$

where $y_j(k+1) = \mathbf{W}_j^T \boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k))$, $g_i(k) = 4 / (e^{\mathbf{V}_j^{iT} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{iT} \mathbf{x}(k)})^2$, and

$$\boldsymbol{\sigma}_j(k) = [1 \quad \tanh(\mathbf{V}_j^{1T} \mathbf{x}(k)) \quad \dots \quad \tanh(\mathbf{V}_j^{HT} \mathbf{x}(k))]^T.$$

Stacking the $N+1$ outputs over the time argument k we then obtain

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \boldsymbol{\Psi}_j \mathbf{U}_j, \boldsymbol{\Psi}_j \in \mathbf{R}^{(N+1) \times (H+1+H(nu+1))},$$

$$\boldsymbol{\Psi}_j = \begin{bmatrix} \boldsymbol{\sigma}_j^T(\mathbf{x}(0)) & w_j^1 g_1(0) \mathbf{x}^T(0) & \dots & w_j^H g_H(0) \mathbf{x}^T(0) \\ \boldsymbol{\sigma}_j^T(\mathbf{x}(1)) & w_j^1 g_1(1) \mathbf{x}^T(1) & \dots & w_j^H g_H(1) \mathbf{x}^T(1) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\sigma}_j^T(\mathbf{x}(N)) & w_j^1 g_1(N) \mathbf{x}^T(N) & \dots & w_j^H g_H(N) \mathbf{x}^T(N) \end{bmatrix}. \quad (15)$$

As $\mathbf{E}_{j+1} = \mathbf{Y}_{j+1} - \mathbf{Y}_d = \mathbf{Y}_j + \boldsymbol{\Psi}_j \mathbf{U}_j - \mathbf{Y}_d = \mathbf{E}_j + \boldsymbol{\Psi}_j \mathbf{U}_j$, the quadratic optimization problem (12) can be rewritten

as

$$\mathbf{U}_j^* = \arg \min_{\mathbf{U}_j} \left\| \mathbf{U}_j^T \mathbf{X} \mathbf{U}_j + 2 \mathbf{Z} \mathbf{U}_j + \mathbf{E}_j^T \mathbf{R} \mathbf{E}_j \right\|_2^2, \quad (16)$$

$$\mathbf{X} = \boldsymbol{\Psi}_j^T \mathbf{R} \boldsymbol{\Psi}_j + \mathbf{Q}, \quad \mathbf{Z} = \mathbf{E}_j^T \mathbf{R} \boldsymbol{\Psi}_j.$$

Using the matrix derivative with respect to vectors and noting that \mathbf{X} is symmetric due to the fact that \mathbf{R} and \mathbf{Q} are symmetric, we obtain the analytic solution of the optimization problem (16):

$$\mathbf{U}_j^* = -(\mathbf{X}^T)^{-1} \mathbf{Z}^T = -(\boldsymbol{\Psi}_j^T \mathbf{R} \boldsymbol{\Psi}_j + \mathbf{Q})^{-1} \boldsymbol{\Psi}_j^T \mathbf{R} \mathbf{E}_j = -\mathbf{K}_j \mathbf{E}_j. \quad (17)$$

The optimal vector \mathbf{U}_j^* contains the increments of the NN weights. Using the notation

$\mathbf{K}_j = [\mathbf{K}_j^{wT} \quad \mathbf{K}_j^{v^1T} \quad \dots \quad \mathbf{K}_j^{v^HT}]^T$, where $\mathbf{K}_j^w \in \mathbf{R}^{(H+1) \times (N+1)}$, $\mathbf{K}_j^{v^i} \in \mathbf{R}^{(nu+1) \times (N+1)}$, and the optimal inputs taken from

(12), the first two equations in (10) could be written as

$$\begin{aligned}\mathbf{W}_{j+1} &= \mathbf{W}_j - \mathbf{K}_j^w \mathbf{E}_j, \\ \mathbf{V}_{j+1}^i &= \mathbf{V}_j^i - \mathbf{K}_j^v \mathbf{E}_j.\end{aligned}\quad (18)$$

The iterative weight adjustment equations (18) are of ILC type [58], and they depend on the error at the current iteration. The norm-optimal ILC formulation is more general since the c.f. also incorporates the penalty on the weights, and it allows for a degree of freedom in the learning.

C. Generating Second Order Gradient Information

Obtaining the first order gradient information by introducing perturbances in the reference trajectory is valid only for small deviations around the nominal trajectory. Hence, the first order gradient information is accurate only for small amplitude deviations of the controller parameters around the nominal ones.

This idea can be further extended to obtain the second order gradient information. Let $f: \mathbf{R} \rightarrow \mathbf{R}$ be a smooth function of real scalar argument ρ . Assume that a first order model obtained by Taylor series expansion is used to describe the behavior of f in the vicinity of ρ^* :

$$\begin{aligned}f(\rho^* + \Delta\rho^1) &\approx f(\rho^*) + f'(\rho^*) \cdot \delta\rho^1, \\ \delta\rho^1 &= \rho - \rho^*.\end{aligned}\quad (19)$$

The gradient of f with respect to ρ can be accurately estimated at $\rho = \rho^*$ for small deviations $\delta\rho^1$

$$\left. \frac{d\hat{f}}{d\rho} \right|_{\rho=\rho^*} = \hat{f}'(\rho^*) \approx \frac{f(\rho) - f(\rho^*)}{\delta\rho^1}.\quad (20)$$

However, for a larger deviation $\delta\rho^2 > \delta\rho^1$ around the nominal argument, the value $f(\rho = \rho^* + \delta\rho^2)$ cannot be accurately represented by the first order model, requiring the use of a second order model obtained from the Taylor series expansion

$$f(\rho = \rho^* + \delta\rho^2) \approx f(\rho^*) + \hat{f}'(\rho^*) \cdot \delta\rho^2 + \frac{1}{2} \hat{f}''(\rho^*) (\delta\rho^2)^2.\quad (21)$$

This will lead to the following estimate of the second order derivative of f with respect to ρ , at $\rho = \rho^*$:

$$\left. \frac{d^2\hat{f}}{d\rho^2} \right|_{\rho=\rho^*} = \hat{f}''(\rho^*) \approx 2 \frac{f(\rho) - f(\rho^*) - \hat{f}'(\rho^*) \cdot \delta\rho^2}{(\delta\rho^2)^2}.\quad (22)$$

This will be further used to estimate the second order gradients of $y(k)$ and $u(k)$.

In the first step, for each controller parameter ρ_h we will calculate, using (5), the $\{\delta r(k)\}$ deviation from the nominal reference trajectory corresponding to a perturbation amplitude $\mu_h \delta \rho_h$. The scaling factor μ_h is used to ensure small amplitude deviation of the reference trajectory. This will allow to estimate the gradients of the controlled output and of the control signal with respect to any controller parameter, as indicated in (8).

In the second step, we will consider the second order model as a more accurate representation of the controlled output's deviation around the nominal trajectory:

$$\begin{aligned} y(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho}) &\approx y_n(k, \boldsymbol{\rho}) + \sum_{h=1}^{n_p} \frac{\partial \hat{y}(k, \boldsymbol{\rho})}{\partial \rho_h} \delta \rho_h + \frac{1}{2} \sum_{h=1}^{n_p} \frac{\partial^2 y(k, \boldsymbol{\rho})}{\partial \rho_h^2} (\delta \rho_h)^2 + \sum_{i,j=1}^{n_p} \frac{\partial^2 y(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j} \delta \rho_i \delta \rho_j, \\ u(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho}) &\approx u_n(k, \boldsymbol{\rho}) + \sum_{h=1}^{n_p} \frac{\partial \hat{u}(k, \boldsymbol{\rho})}{\partial \rho_h} \delta \rho_h + \frac{1}{2} \sum_{h=1}^{n_p} \frac{\partial^2 u(k, \boldsymbol{\rho})}{\partial \rho_h^2} (\delta \rho_h)^2 + \sum_{i,j=1}^{n_p} \frac{\partial^2 u(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j} \delta \rho_i \delta \rho_j, \end{aligned} \quad (23)$$

where $y(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ and $u(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ are obtained experimentally with a nominal reference input affected by a deviation $\{\delta r_h(k)\}$, and the $\partial \hat{y} / \partial \rho_h, \partial \hat{u} / \partial \rho_h, h = 1 \dots n_p$ estimates are computed according to (8). Let us assume at this step that $\{\delta r_h(k)\}$ is obtained from (5) for a certain $\mu_h \delta \rho_h$ and that all the other deviations $\delta \rho_i, i = 1 \dots n_p, i \neq h$ are null. For a scaled amplitude deviation $\{\varepsilon \cdot \delta r_h(k)\}$ with $\varepsilon > 1$ obtained from the deviation $\varepsilon \mu_h \delta \rho_h$, the new perturbed trajectories corresponding to $y(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ and $u(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ can be recorded, and the estimates of

$$\frac{\partial^2 y(k, \boldsymbol{\rho})}{\partial \rho_h^2}, \frac{\partial^2 u(k, \boldsymbol{\rho})}{\partial \rho_h^2}$$

In the third step, the deviations $\{\delta r_{i,j}(k)\}$ are obtained from (5) for the deviations $\varepsilon \mu \delta \rho_i$ and $\varepsilon \mu \delta \rho_j, i, j = 1 \dots n_p, i \neq j$. This is equivalent to simultaneously perturbing two parameters of the controller. The recorded quantities $y(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ and $u(k, \boldsymbol{\rho} + \delta \boldsymbol{\rho})$ obtained with the new reference perturbed by $\{\delta r_{i,j}(k)\}$ are then employed to estimate $\frac{\partial^2 \hat{y}(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j}, \frac{\partial^2 \hat{u}(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j}$ from (23) for the already known first order information obtained in the previous

step.

The NN-based identification allows us to reduce the number of experiments needed in this approach. Second order gradient information estimates can conveniently be obtained via simulation using the closed-loop identified models. This second order gradient information is useful for the calculation of the Hessian matrix \mathbf{R}_j used in the

search algorithm (3), and in the introduction of the sensitivity functions of controlled output and control signal with respect to the controller parameters in the initial c.f. as it will be shown in the next section.

D. Model-Free Controller Tuning for Reduced Sensitivity with Respect to Controller Parameters

The model-based optimization defined in [59] aims to minimize performance indices of the CS's integral squared error and the sensitivity model.

We will consider the generalized c.f.

$$J(\boldsymbol{\rho}) = \frac{1}{2N} E \left\{ \sum_{k=1}^N \left[e^2(k, \boldsymbol{\rho}) + \left[\frac{\partial y(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right]^T \boldsymbol{\Gamma}_y \left[\frac{\partial y(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right] + \left[\frac{\partial u(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right]^T \boldsymbol{\Gamma}_u \left[\frac{\partial u(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right] \right\}, \quad (24)$$

where $\boldsymbol{\Gamma}_y = \boldsymbol{\Gamma}_y^T > 0$ and $\boldsymbol{\Gamma}_u = \boldsymbol{\Gamma}_u^T > 0$ are diagonal positive definite matrices, and

$$\begin{aligned} \left[\frac{\partial y(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right]^T &= \left[\frac{\partial y(k, \boldsymbol{\rho})}{\partial \rho_1} \quad \dots \quad \frac{\partial y(k, \boldsymbol{\rho})}{\partial \rho_{n_p}} \right] \\ \left[\frac{\partial u(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right]^T &= \left[\frac{\partial u(k, \boldsymbol{\rho})}{\partial \rho_1} \quad \dots \quad \frac{\partial u(k, \boldsymbol{\rho})}{\partial \rho_{n_p}} \right] \end{aligned} \quad (25)$$

are the sensitivity functions of the controlled output and of the control signal with respect to the controller parameter vector $\boldsymbol{\rho}$. Using sensitivity functions with respect to the controller parameters allows for a more cautious controller tuning strategy, which results in a more robust CS. While these sensitivity functions cannot be analytically formulated because the process model is unknown, they can be estimated as it will be further shown.

The c.f. (24) can be evaluated for the current iteration using the gradient estimation method. This model-free approach is different to that in the model-based paradigm where these sensitivity functions can be evaluated analytically using appropriate sensitivity models. The sensitivity models for the controller parameters at the current iteration are estimated using the parameters from the previous iteration. The c.f. gradient estimation with respect to $\boldsymbol{\rho}$ is

$$\frac{\partial \hat{J}}{\partial \boldsymbol{\rho}} = \frac{1}{N} \sum_{k=1}^N \left[e(k, \boldsymbol{\rho}) \frac{\partial \hat{y}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} + \left[\frac{\partial^2 \hat{y}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}^2} \right]^T \boldsymbol{\Gamma}_y \left[\frac{\partial \hat{y}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right] + \left[\frac{\partial^2 \hat{u}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}^2} \right]^T \boldsymbol{\Gamma}_u \left[\frac{\partial \hat{u}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right] \right], \quad (26)$$

where

$$\begin{aligned} \left[\frac{\partial^2 \hat{y}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}^2} \right]^{-T} &= \left[\frac{\partial^2 \hat{y}(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j} \right]^{-T} \in \mathbf{R}^{n_p \times n_p}, \\ \left[\frac{\partial^2 \hat{u}(k, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}^2} \right]^{-T} &= \left[\frac{\partial^2 \hat{u}(k, \boldsymbol{\rho})}{\partial \rho_i \partial \rho_j} \right]^{-T} \in \mathbf{R}^{n_p \times n_p}, \quad i, j = 1 \dots n_p. \end{aligned} \quad (27)$$

The first and the second order gradient information are estimated using the approach described in the previous sections. The minimization of the c.f. can be carried out by a search algorithm similar to (3). The degrees of freedom in the design influence the choice of the weighting matrices $\boldsymbol{\Gamma}_y$ and $\boldsymbol{\Gamma}_u$. The gradient of the c.f. can be evaluated using the sensitivity functions with the controller parameters at the current iteration.

E. A Mechanism for Constrained Optimization Using Penalty Functions

The optimization problem in the data-driven tuning context is defined as

$$\begin{aligned} \boldsymbol{\rho}^* &= \arg \min_{\boldsymbol{\rho} \in D_S} J(\boldsymbol{\rho}), \\ \text{subject to } \boldsymbol{\rho} \in D_S &= \{\boldsymbol{\rho} \mid u_{\min}(k) \leq u(k, \boldsymbol{\rho}) \leq u_{\max}(k), \Delta u_{\min}(k) \leq \Delta u(k, \boldsymbol{\rho}) \leq \Delta u_{\max}(k), k = 1 \dots N\}, \end{aligned} \quad (28)$$

where the c.f. J can penalize, for example, the expected mean squared control error $e^2(k, \boldsymbol{\rho}) = [r(k) - y(k, \boldsymbol{\rho})]^2$, where e is the control error. There are many different discrete sum-type c.f.'s that can be selected for the optimal tuning of controllers.

The control signal can be penalized as well and, as shown in Section III.D, weighted sensitivity functions with respect to the controller parameters can be also added to the c.f.

The interior point barrier algorithm provides a solution for this type of problems in the deterministic case. As shown in [60] for inequality constraints concerning only the control signal saturation, the constrained optimization problem is transformed into an unconstrained optimization problem by using the penalty functions. The logarithmic barrier penalty function becomes unbounded as the constraints are closer to being violated, as always is the case in the stochastic framework. A solution to this problem is given in [60], where the quadratic penalty functions are used. We propose the following augmented c.f. which accounts for the inequality constraints with regard the control signal saturation and the control signal rate:

$$\begin{aligned}
 \tilde{J}_{p_j}(\boldsymbol{\rho}) &= J(\boldsymbol{\rho}) + p_j \phi(\boldsymbol{\rho}), \\
 \phi(\boldsymbol{\rho}) &= \frac{1}{2} \sum_{m=1}^c [\max\{0, -q_m(\boldsymbol{\rho})\}]^2, \quad c = 4N, \\
 \mathbf{q}(\boldsymbol{\rho}) &= [q_1(\boldsymbol{\rho}) \quad \dots \quad q_m(\boldsymbol{\rho}) \quad \dots \quad q_c(\boldsymbol{\rho})]^T \in \mathbf{R}^c \\
 &= [u_{\max}(1) - u(1, \boldsymbol{\rho}) \quad \dots \quad u_{\max}(N) - u(N, \boldsymbol{\rho}) \quad u(1, \boldsymbol{\rho}) - u_{\min}(1) \quad \dots \quad u(N, \boldsymbol{\rho}) - u_{\min}(N) \\
 &\quad u_{\max}(1) - \Delta u(1, \boldsymbol{\rho}) \quad \dots \quad \Delta u_{\max}(N) - \Delta u(N, \boldsymbol{\rho}) \quad \Delta u(1, \boldsymbol{\rho}) - \Delta u_{\min}(1) \quad \dots \quad \Delta u(N, \boldsymbol{\rho}) - \Delta u_{\min}(N)]^T \in \mathbf{R}^{4N},
 \end{aligned} \tag{29}$$

where the positive and strictly increasing sequence of penalty parameters $\{p_j\}_{j \geq 0}$, $p_j \rightarrow \infty$, guarantees that the minimum of the sequence of augmented c.f.s $\{\tilde{J}_{p_j}(\boldsymbol{\rho})\}_{j \geq 0}$ will converge to a solution to the constrained optimization problem (28), the subscript m , $m = 1 \dots c$, is the constraint index, $q_m(\boldsymbol{\rho}) > 0$ is the m^{th} constraint.

The optimization problem (29) is solved using a stochastic approximation algorithm which makes use of the experimentally obtained gradient of $\tilde{J}_{p_j}(\boldsymbol{\rho})$.

The quadratic penalty function $\phi(\boldsymbol{\rho})$ in (29) uses the maximum function which in this case is non-differentiable only at zero. Given that $\phi(\boldsymbol{\rho})$ is of a Lipschitz type and non-differentiable at a set of points of zero Lebesgue measure, the algorithm visits the zero-measure set with a probability zero when a normal noise distribution is assumed [60]. Therefore, using

$$\frac{\partial [\max\{0, -q_m(\boldsymbol{\rho})\}]^2}{\partial \rho_h} = -2 \max\{0, -q_m(\boldsymbol{\rho})\} \frac{\partial q_m(\boldsymbol{\rho})}{\partial \rho_h}, \tag{30}$$

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

$$\frac{\partial \tilde{J}_{p_j}(\boldsymbol{\rho})}{\partial \rho_h} = \frac{\partial J(\boldsymbol{\rho})}{\partial \rho_h} - p_j \sum_{m=1}^c \{\max\{0, -q_m(\boldsymbol{\rho})\} \frac{\partial q_m(\boldsymbol{\rho})}{\partial \rho_h}\}. \tag{31}$$

The first term in (31), corresponding to the gradient of the original c.f., requires knowing the gradient $\partial y(k)/\partial \boldsymbol{\rho}$, and the second term in (31) requires the gradients of the control signal with respect to $\boldsymbol{\rho}$ and the gradient of the control signal rate with respect to $\boldsymbol{\rho}$. All these variables can be estimated using the NN-based identification mechanism given in (8). The derivative of the control signal rate with respect to the parameter vector $\boldsymbol{\rho}$ is estimated using the finite differences approximation approach for the sampling period δt

$$\frac{\partial \Delta \hat{u}(k)}{\partial \rho_h} = \frac{1}{\delta t} \left[\frac{\partial \hat{u}(k)}{\partial \rho_h} - \frac{\partial \hat{u}(k-1)}{\partial \rho_h} \right], \quad h = 1 \dots n_p, \quad k = 1 \dots N. \quad (32)$$

The nonlinearity in the penalty function induces gradient estimation errors, such as bias, and also changes the normal distribution of the measured constraints. Under linearity and Gaussian noise assumptions, the constraints $q_m(\mathbf{p}) > 0$ are measured without any bias because the control signal is linearly affected by noise. While there is no problem if the quadratic penalty functions are dominated by 0 or by $-q_m(\mathbf{p})$, estimation errors will occur when the constraints are close to zero. The algorithm will work with a reasonable degree of accuracy even for small bias errors in the estimation.

As the constraints are weighted in the c.f. only when they are violated, our approach allows to initially use solutions that do not respect all the inequality constraints. In the long run, as the length of the sequence $\{p_j\}_{j \geq 0}$ increases, the impact of gradient due to the violated constraints is decisive and the reference trajectory tracking objective is neglected in order to fulfill the constraints in the optimization problem. The measurement noise, which adds a persistent excitation to the signals, is useful for the NN-based identification.

IV. The IDDA Algorithm

The proposed iterative data-driven algorithm (IDDA) consists of the following steps.

Step S1. Starting with an initial value of \mathbf{p} , choose the upper and lower bounds for the control signal, the upper and lower bounds for the control signal rate and the desired control criterion, namely the c.f. $J(\mathbf{p})$. Choose the tolerances tol_N for stopping the stochastic search algorithm. Set the iteration index for \mathbf{p} and $\{p_j\}_{j \geq 0}$ to $j = 0$. Choose the sequence $\{p_j\}_{j \geq 0}$ and the initial step size γ_0 .

Step S2. Conduct the normal experiment with the current \mathbf{p}_j for the nominal reference input. Evaluate the c.f. $\tilde{J}(\mathbf{p}_j)$.

Step S3. Train the models M_{ry}^j and M_{ru}^j using the ILC approach. Calculate the perturbed reference trajectories $\{\Delta r_h(k)\}$ using (5) to be introduced in (8). Use the models M_{ry}^j and M_{ru}^j to estimate $\partial \hat{y}(k) / \partial \mathbf{p}$, $\partial \hat{u}(k) / \partial \mathbf{p}$ and $\partial \Delta \hat{u}(k) / \partial \mathbf{p}$ using (8) and (32). Evaluate the gradient of the c.f. by means of (31).

Step S4. Calculate the next controller parameter vector \mathbf{p}_{j+1}

$$\mathbf{p}_{j+1} = \mathbf{p}_j - \gamma_j \left. \text{est} \left\{ \frac{\partial \tilde{J}}{\partial \mathbf{p}} \right\} \right|_{\mathbf{p}=\mathbf{p}_j} \quad (33)$$

Step S5. If the gradient at the current iteration is below a specified threshold $\left. \text{est} \left\{ \frac{\partial \tilde{J}}{\partial \mathbf{p}} \right\} \right|_{\mathbf{p}=\mathbf{p}_j} < \text{tol}_N$, stop the algorithm.

Otherwise, set $j = j + 1$ and jump to step *S2*.

V. Case Study

The case study deals with the angular positioning on vertical motion of an experimental twin-rotor aerodynamic system [61], with a horizontal rotor which produces vertical motion and a vertical rotor producing horizontal motion (Fig. 1). An adjustable counterweight beam is positioned at the middle of the main beam supporting the two rotors.

The horizontal motion is not allowed in this case study, so the horizontal position is considered fixed. The nonlinear equations describing the vertical motion are [61]

$$\begin{aligned} J_v \dot{\Omega}_v &= l_m F_v(\omega_v) - \Omega_v k_v + \underbrace{g((A-B)\cos\alpha_v - C\sin\alpha_v)}_{M_{v1}}, \\ \dot{\alpha}_v &= \Omega_v, \\ I_v \dot{\omega}_v &= M(U_v) - M_r(\omega_v), \end{aligned} \quad (34)$$

where $U_v = u$ is the control signal, $\alpha_v = y$ is the process output, and M_{v1} depends on the constant parameters A , B and C of the mechanical ensemble and on α_v .

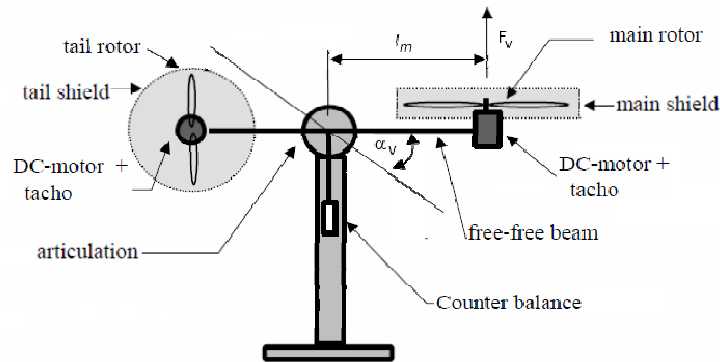


Fig. 1 The twin-rotor aerodynamic experimental setup [61].

The nonlinearity of the system is also due to the static nonlinear mapping of the control signal u to the rotor angular speed and of the rotor angular speed to the aerodynamic force. The parameters in (34) are [61]

$$\begin{aligned} J_v &= 0.02421 \text{ kg m}^2, I_v = 4.5 \cdot 10^{-5} \text{ kg m}^2, k_v = 0.0127 \text{ kg m}^2 / \text{s}, B - A = 0.05 \text{ rad kg m}, \\ l_m &= 0.2 \text{ m}, C = 0.0936 \text{ rad kg m}. \end{aligned} \quad (35)$$

This nonlinear model is used only to obtain an initial controller, but not in the actual tuning process.

A. Neural Network-Based Identification

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 III.B, hyperbolic tangent activation functions are employed in the hidden layer, and a linear function is employed as the output neuron activation function.

This nonlinear autoregressive exogenous (NARX) NN 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 the closed-loop output and the control signal given in (6) and (7), respectively.

The training of the two NARX NN architectures is carried out in the ILC framework using the guidelines given in Section III.B. 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 $\mathbf{V}_i \in \mathbf{R}^{5 \times 1}, i = 1 \dots 6$. The initial values of the hidden neurons parameters are chosen from a normal distribution centered at zero with variance 1. Because of the special structure of the NN which is linear in the output weights vector \mathbf{W} , a least squares initialization of \mathbf{W} was performed.

The NN-based identification is carried out on the nominal trajectories of the closed-loop for the initial controller parameters presented in the next section.

We will restrict our further discussion to the identified map M_{ry} . For an experiment of a duration of 90 s, we used 898 samples for training. For the norm-optimal ILC problem, the weighting matrices were chosen as $\mathbf{R} = \mathbf{I}_{898}$ and $\mathbf{Q} = 0.0005 \cdot \mathbf{I}_{37}$, where \mathbf{I}_ζ is the general notation for the ζ^{th} order identity matrix.

Fig. 2 illustrates the training error and of the output neuron weights throughout the NN training iterations.

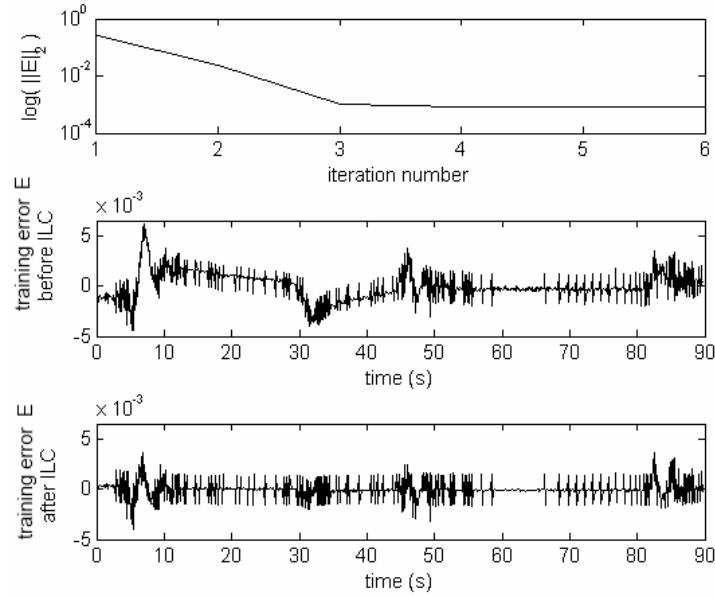


Fig. 2 Neural network training error during iterations.

The training error norm at the first iteration results after the least squares initialization of the output weights vector \mathbf{W} . Results for NNs trained using the ILC framework show a decrease of the training error of about three orders of magnitude in after four iterations, which conclusively demonstrate the efficiency of this ILC-based NN training.

B. Controller Tuning for Tracking Error Minimization and Reduced Sensitivity

This case study considers a discrete-time linear PI controller with the transfer function $H(q^{-1}) = (\rho_1 + \rho_2 q^{-1}) / (1 - q^{-1})$ and with the parameter vector $\boldsymbol{\rho} = [\rho_1 \quad \rho_2]$. The initial parameter vector obtained experimentally is $\boldsymbol{\rho}_0 = [0.012 \quad 0.001]^T$.

The purpose is to compare the results when the sensitivity with respect to the change in controller parameter is taken into account. The two performance indices to be minimized together with the corresponding optimization problems are

$$\begin{aligned}
 J_1(\boldsymbol{\rho}) &= \frac{1}{2N} E \left\{ \sum_{k=1}^N (r(k) - y(k))^2 \right\}, \\
 \boldsymbol{\rho}^* &= \arg \min_{\boldsymbol{\rho} \in D_S} J_1(\boldsymbol{\rho}),
 \end{aligned} \tag{36}$$

and

$$J_2(\boldsymbol{\rho}) = \frac{1}{2N} E \left\{ \sum_{k=1}^N (r(k) - y(k))^2 + \alpha \left(\frac{\partial y(k)}{\partial \rho_1} \right)^2 \right\}, \quad (37)$$

$$\boldsymbol{\rho}^* = \arg \min_{\boldsymbol{\rho} \in D_s} J_2(\boldsymbol{\rho}).$$

The sampling period is set to 0.1 s and the experiment length is set to 90 s. The desired trajectory specified as a reference input to the CS is a step-signal of amplitude 0.1 rad (approximately 5.72°) for 35 s, a ramp-type signal for 20 s, and then no signal (zero) for the remaining 35 s.

The common objective of the two optimization problems is to minimize the reference tracking error. In addition, the objective of the second problem is to reduce the sensitivity of the CS with respect to changes in ρ_1 . The weighting parameter α in (37) was set to $\alpha = 10000$.

For the sake of comparing the performance of the CSs with optimized controller by IDDA applied to (36) and to (37), another sensitivity performance criterion is defined as

$$J = \frac{1}{N} \sum_{k=1}^N (y^{final}(k) - y^{initial}(k))^2, \quad (38)$$

where $y^{initial}(k)$ for each CS is recorded with the optimized controller and $y^{final}(k)$ is obtained after the parameter ρ_1 was disturbed by 1%.

Results illustrating the efficiency of using IDDA to minimize $J_1(\boldsymbol{\rho})$ and $J_2(\boldsymbol{\rho})$ are presented in Fig. 3 and in Fig. 4, respectively. Other details are given in Table 1.

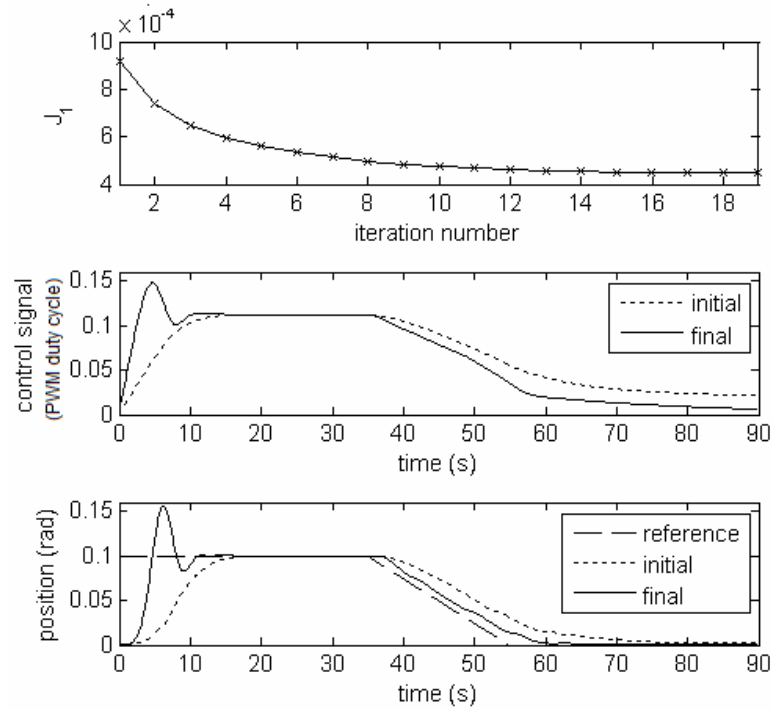


Fig. 3 Results concerning the application of IDDA to the minimization of $J_1(\rho)$ in (36).

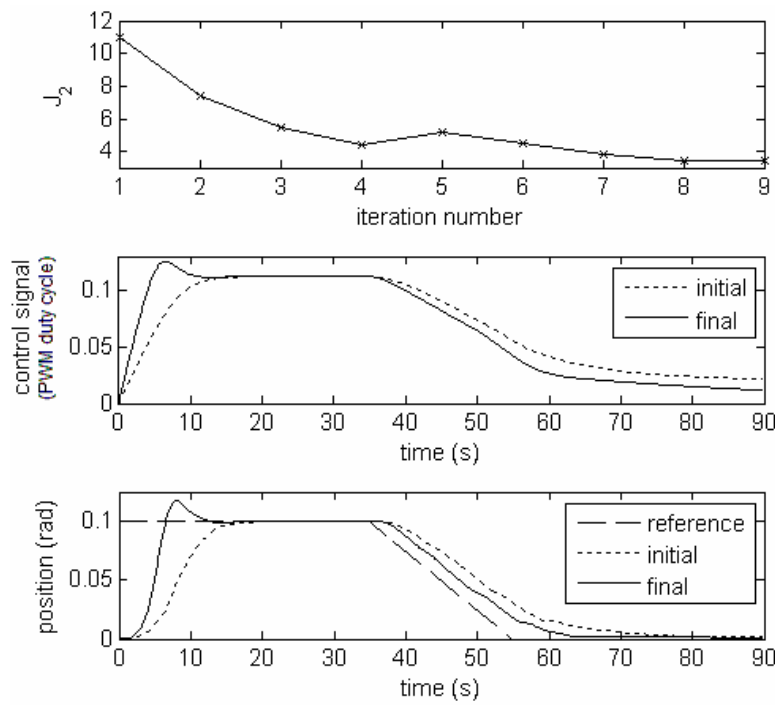


Fig. 4 Results concerning the application of IDDA to the minimization of $J_2(\rho)$ in (37).

Table 1 Comparison of IFT applied to the optimization problems (36) and (37)

	J_1	J_2
Final controller parameters	$\boldsymbol{\rho}_{19} = [0.0254 \quad 0.0147]^T$	$\boldsymbol{\rho}_9 = [0.0184 \quad 0.0076]^T$
Final sensitivity criterion	$J = 5.88 \cdot 10^{-8}$	$J = 3.5 \cdot 10^{-8}$
$\{\gamma_j\}$	$0.9/(j+1)^{0.6}$	$0.6/(j+1)^{0.6}$
\mathbf{R}_i	\mathbf{I}_2	\mathbf{I}_2

The results presented in Table 1 show that the sensitivity with respect to disturbances in the controller is lower for the criterion J_2 . This validates the proposed data-driven approach.

We did not consider the constraints in this case study, since it may be possible that a sensitivity reduction is not achieved due to the violation of constraints. This approach allows for a cautious controller tuning by introducing the sensitivity with respect to the controller parameters in the c.f. This has the equivalent effect of introducing a weighted penalty on the control signal the c.f. Therefore, the proposed tuning method can result in more robust CSs.

There are two additional aspects referring to the minimization of J_1 , which deserve mentioning.

In the first simulation scenario we ran our learning algorithm multiple times starting with the same initial controller parameters $\boldsymbol{\rho}_0$. At this point, only the random NN weight-initialization led to different results in NN training, which in turn led to gradient estimation errors. These estimation errors propagate from one iteration to the next. After 20 runs, the results show a standard deviation of $6.5 \cdot 10^{-9}$ (or 0.0014% of the mean value) for the final value of J_1 , which can be considered as insignificant. The variations of the final controller parameters were also insignificant.

In the second simulation scenario we started each time with different controller parameters in order to investigate their impact on the optimization. The controller parameters were perturbed by random normally-distributed values such that the variations for each controller parameter were within a $\pm 3\%$ domain. After 20 runs, the standard deviation for the final value of J_1 of was about $25.27 \cdot 10^{-9}$ (or 0.0056% of the mean value), which were about 4 times larger than in the previous scenario. The standard deviation of the final controller parameter values was also larger, but the mean values were the same as those shown in Table 1 for J_1 .

Our algorithm leads to the same final solution even when starting with different initial points. It also indicates a convex c.f. even if the underlying problem is nonlinear. These effects are to be expected, because the desired response to be tracked is close to the initial responses and we do not require drastic performance modifications. However, if the disturbances on the controller parameters are larger, which results in larger differences between the initial response and the desired response, our algorithm is producing different local minima.

C. Controller Tuning for Constrained Optimization

We consider another optimization problem in which the constraints imposed to the control signal rate are taken into account. As in the previous section, the main goal is to minimize the reference tracking error. The problem is formulated as

$$\begin{aligned}
 J_1(\boldsymbol{\rho}) &= \frac{1}{2N} E\left\{\sum_{k=1}^N (r(k) - y(k))^2\right\}, \\
 \boldsymbol{\rho}^* &= \arg \min_{\boldsymbol{\rho} \in D_S} J_1(\boldsymbol{\rho}), \\
 \text{subject to } \boldsymbol{\rho} \in D_S &= \{\boldsymbol{\rho} \mid \Delta u_{\min}(k) \leq \Delta u(k, \boldsymbol{\rho}) \leq \Delta u_{\max}(k), k = 1 \dots N\},
 \end{aligned} \tag{39}$$

Let's consider the specific case where the control signal rate assumes values within the $-0.02 \leq \Delta u(k) \leq 0.02$ interval, and 900 samples are collected during the experiment, which will result in 1800 inequality constraints. For the sake of comparison, we will use two gradient search algorithms are used for IFT. We use first the steepest descent search algorithm and then the BFGS update algorithm to get a Gauss-Newton approximation of the Hessian matrix \mathbf{R}_i considered in the first step [62]. The initial approximation for \mathbf{R}_i in the BFGS algorithm is an identity matrix which is recursively updated. When one ore more constraints become active/inactive, \mathbf{R}_i resets back to the identity matrix since the c.f. is different due to the penalty function weighted by the factor p_j .

The initial parameter vector of the controller is the same as in the previous section, namely $\boldsymbol{\rho}_0 = [0.012 \quad 0.001]^T$. The sequences $\gamma_j = 0.3/(j+1)^{0.6}$ and $\{p_j\} = 15 \cdot j^{0.5}$ are the same in both cases. The results are presented in Fig. 5.

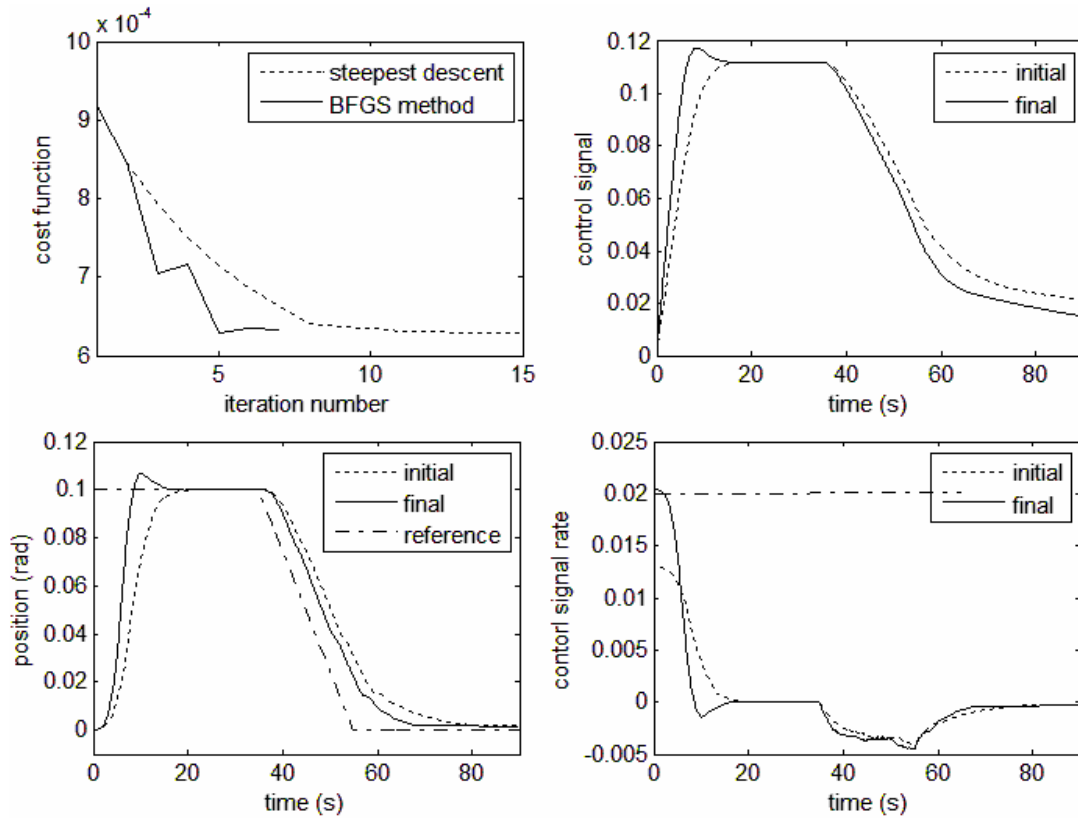


Fig. 5 Results concerning the application of IDDA to the minimization of $J_1(\rho)$ in (39), which deals with the constrained case.

Fig. 5 clearly shows that the proposed approach iteratively adjusts the controller parameters in order to minimize the reference tracking error. The c.f. shown in Fig. 5 is the original one $J_1(\rho)$ from (39), augmented with the penalty function. The constraints imposed to the control signal rate are violated in the steepest descent algorithm at the seventh iteration, after which the tuning deactivates the constraints. The constraints are violated in the BFGS algorithm at the fourth iteration, after which the tuning reduces the control signal rate until it reaches the upper bound. This is achieved in seven iterations compared to fifteen iterations for steepest descent. The small number of iterations considerably reduces the time spent for controller tuning. The final parameter vectors for the controllers, after the tuning is done, are $\rho_{15} = [0.0189 \ 0.0011]^T$ and, respectively, $\rho_7 = [0.0187 \ 0.0017]^T$. The final responses for all the CS signals are almost identical.

D. Discussion of the results and comparisons

The number of iterations is low and comparable with the number of iterations needed for the RL approach reported in [39]. While the learning algorithm for offline gradient information reported in [39] uses approximate first principles process models, our approach does not use any specific process models but NN-based identified models for the closed-loop CS. On the other hand, the line-search employed in [39] requires additional evaluations of the control policy in real environment conditions, whereas we use a BFGS algorithm to reduce the number of real-world experiments. Our approach also addresses the constrained environment, in addition to its advantage of providing a model-free tuning.

As shown in this section, we used a BFGS algorithm in order to compare it with our pure gradient algorithm, because both apply to the same structure of the optimization problem and use the same collected data. It has been shown that the BFGS algorithm outperforms the pure gradient descent algorithm.

While the SPSA algorithm [2] can be used in the optimization, it is different from our algorithm because it does not exploit the problem structure. Given the low dimension of the search space, the gradient estimation mechanism can still propagate the gradient information only from measurements of the augmented c.f. It should be noted that the SPSA algorithm does not outperform our new algorithm, because it requires more experimental evaluations of the c.f., and it scales poorly for a large number of parameters, which will require a prohibitive number of experiments.

Our approach uses identified CS models that are valid only in the vicinity of the nominal trajectories at the current iteration. Since the closed-loop CS typically has a low-pass behavior, we can use simple NN architectures for identification purposes.

When estimating the gradient, the perturbed reference trajectories used as NN inputs are very similar to the nominal reference input, which avoids NN generalization problems and results in low gradient estimation errors. Our identification approach does not need specially designed trajectories to be run on the real-world CS because we do not need closed-loop CS models that are valid in a wide operating range. The NN models used during the iterations are independent of each other, being valid only near the nominal trajectories of their corresponding iterations.

VI. Conclusion

The proposed IDDA, which solves an optimal control problem based on the minimization of the mean squared control error with constraints imposed to the control signal and to the control signal rate, can successfully be used for controller tuning, resulting in a reduced sensitivity of the CS with respect to variations of the controller parameters.

The NN-based identification mechanism reduces the number of experiments conducted on the real-world process compared to the experiment-based gradient estimation approaches by avoiding the gradient estimation from real-world experiments. In addition, the NN training is efficiently performed using an ILC framework by considering the NN as a nonlinear dynamic system in the iteration domain, providing the basis for a systematic convergence analysis and guarantee. Our approach can be extended with an ILC-based reference input tuning in order to achieve more aggressive maneuvers for the considered aerodynamic experimental setup.

Our algorithm offers several advantages. First, other integral-type constraints can be added to the optimization problem without requiring additional experiments and without increasing the complexity; nonlinear controllers can also be tuned. Second, the model needs only to be valid around the nominal trajectory, where the gradients are generated, and not in a wider operating range. Third, the numerical differentiation problems in noisy environments are avoided by the use of the identified CS models. Fourth, the optimization approach using quadratic penalty functions ensures the operation in a noisy environment avoiding the shortcoming of the Interior Point Barrier algorithm. Finally, by avoiding gradient estimation using a controller parameters perturbation approach, the controller is not updated until a gradient search direction which stabilizes the CS is computed.

The tuning capabilities of our algorithm show a good compromise between an exhaustive search of the state space/environment strategy and a model-based strategy by using limited information about the environment and therefore requiring only a reasonable search effort.

A limitation of our IDDA is that it does not guarantee the convergence to the global optimum. A convergence analysis to the global optimum has to be carried out. Future research will be focused on the thorough comparison of our algorithm's performance with those obtained by similar model-free approaches.

Acknowledgments

This work was supported by grants from the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, from the Partnerships in priority areas – PN II program of the Romanian National Authority for Scientific Research ANCS, CNDI – UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-0732, and from the NSERC of Canada.

References

- [1] Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O., “Iterative Feedback Tuning: Theory and Applications,” *IEEE Control Systems Magazine*, Vol. 18, No. 4, 1998, pp. 26–41.
doi: 10.1109/37.710876
- [2] Spall, J. C., and Cristion, J. A., “Model-Free Control of Nonlinear Stochastic Systems with Discrete-Time Measurements,” *IEEE Transactions on Automatic Control*, Vol. 43, No. 9, 1998, pp. 1198–1210.
doi: 10.1109/9.718605
- [3] Shi, G., and Skelton, R. E., “Markov Data-Based LQG Control,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 122, No. 3, 2000, pp. 551–559.
doi: 10.1115/1.1286868
- [4] Campi, M. C., Lecchini, A., and Savaresi, S. M., “Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers,” *Automatica*, Vol. 38, No. 8, 2002, pp. 1337–1346.
doi: 10.1016/S0005-1098(02)00032-8
- [5] Kadali, R., Huang, B., and Rossiter, A., “A Data Driven Subspace Approach to Predictive Controller Design,” *Control Engineering Practice*, Vol. 11, No. 3, 2003, pp. 261–278.
doi: 10.1016/S0967-0661(02)00112-0
- [6] Radac, M.-B., Precup, R.-E., Petriu, E. M., and Preitl, S., “Application of IFT and SPSA to Servo System Control,” *IEEE Transactions on Neural Networks*, Vol. 22, No. 12, 2011, pp. 2363–2375.
doi: 10.1109/TNN.2011.2173804
- [7] Formentin, S., and Karimi, A., “A Data-Driven Approach to Mixed-Sensitivity Control with Application to an Active Suspension System,” *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 4, 2013, pp. 2293–2300.
doi: 10.1109/TII.2012.2220556
- [8] 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 Transactions on Industrial Informatics*, Vol. 9, No. 4, 2013, pp. 2327–2336.
doi: 10.1109/TII.2012.2220973

- [9] Chi, R., Hou, Z., Jin, S., Wang, D., and Hao, J., “A Data-Driven Iterative Feedback Tuning Approach of ALINEA for Freeway Traffic Ramp Metering with PARAMICS Simulations,” *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 4, 2013, pp. 2310–2317.
doi: 10.1109/TII.2013.2238548
- [10] Hou, Z.-S., and Wang, Z., “From Model-Based Control to Data-Driven Control: Survey, Classification and Perspective,” *Information Sciences*, Vol. 235, Jun. 2013, pp. 3–35.
doi: 10.1016/j.ins.2012.07.014
- [11] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
- [12] Werbos, P. J., “Approximate Dynamic Programming for Real-Time Control and Neural Modeling,” *Handbook of Intelligent Control*, edited by D. White and D. Sofge, Van Nostrand, New York, 1992, pp. 493–525.
- [13] Zhu, Y., and Hou, Z.-S., “Controller Compact Form Dynamic Linearization Based Model Free Adaptive Control,” *Proceedings of 51st IEEE Conference on Decision and Control (CDC 2012)*, Maui, HI, USA, 2012, pp. 4817–4822.
doi: 10.1109/CDC.2012.6426479
- [14] Coelho, L. S., Pessôa, M. W., Sumar, R. R., and Coelho, A. A. R., “Model-Free Adaptive Control Design Using Evolutionary-Neural Compensator,” *Expert Systems with Applications*, Vol. 37, No. 1, 2010, pp. 499–508.
doi: 10.1016/j.eswa.2009.05.042
- [15] Iverson, D. L., Martin, R., Schwabacher, M., Spirkovska, L., Taylor, W., Mackey, R., Castle, J. P., and Baskaran, V., “General Purpose Data-Driven Monitoring for Space Operations,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 9, No. 2, 2012, pp. 26–44.
doi: 10.2514/1.54964
- [16] Vaščák, J., and Rutrich, M., “Path Planning in Dynamic Environment Using Fuzzy Cognitive Maps,” *Proceedings of 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2008)*, Herl’any, Slovakia, 2008, pp. 5–9.
doi: 10.1109/SAMI.2008.4469153
- [17] Boussemart, Y., Cummings, M. L., Las Fargeas, J., and Roy, N., “Supervised vs. Unsupervised Learning for Operator State Modeling in Unmanned Vehicle Settings,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 8, No. 3, 2011, pp. 71–85.
doi: 10.2514/1.46767
- [18] Teodorescu, H.-N., “Taylor and Bi-local Piecewise Approximations with Neuro-Fuzzy Systems,” *Studies in Informatics and Control*, Vol. 21, No. 4, 2012, pp. 367–376.
- [19] Chowdhary, G., How, J. P., and Kingravi, H., “Model Reference Adaptive Control Using Nonparametric Adaptive Elements,” *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, USA, 2012, pp. 1–24.

- [20] John, S., and Pedro, J. O., “Neural Network-Based Adaptive Feedback Linearization Control of Antilock Braking System,” *International Journal of Artificial Intelligence*, Vol. 10, No. S13, 2013, pp. 21–40.
- [21] Dovžan, D., and Škrjanc, I., “Predictive Functional Control Based on an Adaptive Fuzzy Model of a Hybrid Semi-Batch Reactor,” *Control Engineering Practice*, Vol. 18, No. 8, 2010, pp. 979–989.
10.1016/j.conengprac.2010.04.004
- [22] Dovžan, D., and Škrjanc, I., “Recursive Fuzzy C-Means Clustering for Recursive Fuzzy Identification of Time-Varying Processes,” *ISA Transactions*, Vol. 50, No. 2, 2011, pp. 159–169.
doi: 10.1016/j.isatra.2011.01.004
- [23] Zdešar, A., Cerman, O., Dovžan, D., Hušek, P., and Škrjanc, I., “Fuzzy Control of a Helio-Crane: Comparison of Two Control Approaches,” *Journal of Intelligent & Robotic Systems*, Vol. 72, No. 3–4, 2013, pp. 497–515.
doi: 10.1007/s10846-012-9796-0
- [24] Blažič, S., and Škrjanc, I., “A Robust Fuzzy Adaptive Control Algorithm for a Class of Nonlinear Systems,” *Adaptive and Natural Computing Algorithms*, edited by M. Tomassini, A. Antonioni, F. Daolio, and P. Buesser, Lecture Notes in Computer Science, Springer-Verlag, 2013, Vol. 7824, pp. 297–306.
doi: 10.1007/978-3-642-37213-1_31
- [25] Baranyi, P., Tikk, D., Yam, Y., and Patton, R. J., “From Differential Equations to PDC Controller Design via Numerical Transformation,” *Computers in Industry*, Vol. 51, No. 3, 2003, pp. 281–297.
doi: 10.1016/S0166-3615(03)00058-7
- [26] Haber, R. E., del Toro, R. M., and Gajate, A., “Optimal Fuzzy Control System Using the Cross-Entropy Method. A Case Study of a Drilling Process,” *Information Sciences*, Vol. 180, No. 14, 2010, pp. 2777 – 2792.
doi: 10.1016/j.ins.2010.03.030
- [27] Angelov, P., and Yager, R., “A New Type of Simplified Fuzzy Rule-Based Systems,” *International Journal of General Systems*, Vol. 41, No. 2, 2012, pp. 163–185.
doi: 10.1080/03081079.2011.634807
- [28] Penedo, F., Haber, R. E., Gajate, A., and del Toro, R. M., “Hybrid Incremental Modeling Based on Least Squares and Fuzzy K-NN for Monitoring Tool Wear in Turning Processes,” *IEEE Transactions on Industrial Informatics*, Vol. 8, No. 4, 2012, pp. 811–818.
doi: 10.1109/TII.2012.2205699
- [29] Filip, F. G., “A Decision-Making Perspective for Designing and Building Information Systems,” *International Journal of Computers, Communications & Control*, Vol. 7, No. 2, 2012, pp. 264–272.

- [30] Angelov, P., and Yager, R., “Density-Based Averaging – A New Operator for Data Fusion,” *Information Systems*, Vol. 222, Feb. 2013, pp. 163–174.
doi: 10.1016/j.ins.2012.08.006
- [31] Škrjanc, I., “Adaptive Supervisory Predictive Control of a Hybrid Fed-Batch Reactor with Slow Actuator,” *Industrial & Engineering Chemistry Research*, Vol. 46, No. 24, 2007, pp. 8050–8057.
doi: 10.1021/ie070123t
- [32] Škrjanc, I., “Self-Adaptive Supervisory Predictive Functional Control of a Hybrid Semi-Batch Reactor with Constraints,” *Chemical Engineering Journal*, Vol. 136, No. 2–3, 2008, pp. 312–319.
doi: 10.1016/j.cej.2007.04.012
- [33] Walsh, T. J., Nouri, A., Li, L., and Littman, M. L., “Learning and Planning in Environments with Delayed Feedback,” *Autonomous Agents and Multi-Agent Systems*, Vol. 18, No. 1, 2009, pp. 83–105.
doi: 10.1007/s10458-008-9056-7
- [34] Françaolin, C. C., Rao, A. V., Duarte, C., and Martel, G., “Optimal Control of a Surface Vehicle to Improve Underwater Vehicle Network Connectivity,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 9, No. 1, 2012, pp. 1–13.
doi: 10.2514/1.1010002
- [35] Johanyák, Z. C., and Papp, O., “A Hybrid Algorithm for Parameter Tuning in Fuzzy Model Identification,” *Acta Polytechnica Hungarica*, Vol. 9, No. 6, 2012, pp. 153–165.
- [36] Johanyák, Z. C., and Papp, O., “Benchmark Based Comparison of Two Fuzzy Rule Base Optimization Methods,” *Applied Computational Intelligence in Engineering and Information Technology*, edited by R.-E. Precup, S. Kovács, S. Preitl, and E. M. Petriu, *Topics in Intelligent Engineering and Informatics*, Springer-Verlag, 2012, Vol. 1, pp. 83–94.
doi: 10.1007/978-3-642-28305-5_7
- [37] Vaščák, J., “Adaptation of Fuzzy Cognitive Maps by Migration Algorithms,” *Kybernetes*, Vol. 41, No. 3–4, 2012, pp. 429–443.
doi: 10.1108/03684921211229505
- [38] Guzmán, L. G., Gómez, A. S., Ardila, C. J., and Jabba, D., “Adaptation of the GRASP Algorithm to Solve a Multiobjective Problem Using the Pareto Concept,” *International Journal of Artificial Intelligence*, Vol. 11, No. A13, 2013, pp. 222–236.
- [39] Abbeel, P., Quigley, M., and Ng, A. Y., “Using Inaccurate Models in Reinforcement Learning,” *Proceedings of 23rd International Conference on Machine Learning (ICML 2006)*, New York, NY, USA, 2006, Vol. 3, pp. 1–8.

- [40] Lupashin, S., Schöllig, A., Sherback, M., and D'Andrea, R., "A simple learning strategy for high-speed quadcopter multi-flips," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '10)*, Anchorage, AK, USA, 2010, pp. 1642–1648.
doi: 10.1109/ROBOT.2010.5509452
- [41] 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 Engineering Practice*, Vol. 17, No. 2, 2009, pp. 311–321.
doi: 10.1016/j.conengprac.2008.08.001
- [42] 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," *Proceedings of the 38th Annual Conference of IEEE Industrial Electronics Society (IECON 2012)*, Montreal, QC, Canada, 2012, pp. 2500–2505.
doi: 10.1109/IECON.2012.6388856
- [43] Precup, R.-E., Tomescu, M. L., and Preitl, S., "Lorenz System Stabilization Using Fuzzy Controllers," *International Journal of Computers, Communications & Control*, Vol. II, No. 3, 2007, pp. 279–287.
- [44] Nguyen, N. T., and Krishnakumar, K., "Hybrid Intelligent Flight Control with Adaptive Learning Parameter Estimation," *Journal of Aerospace Computing, Information, and Communication*, Vol. 6, No. 3, 2009, pp. 171–186.
doi: 10.2514/1.35929
- [45] Blažič, S., Matko, D., and Škrjanc, I., "Adaptive Law with a New Leakage Term," *IET Control Theory & Applications*, Vol. 4, No. 9, 2010, pp. 1533–1542.
doi: 10.1049/iet-cta.2009.0349
- [46] Alfaro, V.M., and Vilanova, R., "Performance/Robustness Trade-off Design Framework for 2DoF PI Controllers," *Studies in Informatics and Control*, Vol. 21, No. 1, 2012, pp. 75–83.
- [47] Lee, D. J., and Bang, H., "Autonomous Autorotation of an Unmanned Helicopter Using a Reinforcement Learning Algorithm," *Journal of Aerospace Information Systems*, Vol. 10, No. 2, 2013, pp. 98–104.
doi: 10.2514/1.48253
- [48] Triharminto, H. H., Adji, T. B., and Setiawan, N. A., "3D Dynamic UAV Path Planning for Interception of Moving Target in Cluttered Environment," *International Journal of Artificial Intelligence*, Vol. 10, No. S13, 2013, pp. 154–163.
- [49] 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 Engineering Practice*, Vol. 11, No. 9, 2003, pp. 1079–1086.
doi: 10.1016/S0967-0661(02)00231-9
- [50] Precup, R.-E., Doboli, S., and Preitl, S., "Stability Analysis and Development of a Class of Fuzzy Control Systems," *Engineering Applications of Artificial Intelligence*, Vol. 13, No. 3, 2000, pp. 237–247.

doi: 10.1016/S0952-1976(00)00002-6

- [51] Precup, R.-E., and Preitl, S., “Stability and Sensitivity Analysis of Fuzzy Control Systems. Mechatronics applications”, *Acta Polytechnica Hungariva*, Vol. 3, No. 1, 2006, pp. 61–76.
- [52] Precup, R.-E., and Preitl, S., “PI and PID Controllers Tuning for Integral-Type Servo Systems to Ensure Robust Stability and Controller Robustness”, *Electrical Engineering*, Vol. 88, No. 2, 2006, pp. 149–156.
doi: 10.1007/s00202-004-0269-86
- [53] Ure, N. K., Geramifard, A., Chowdhary, G., and How, J. P., “Adaptive Planning for Markov Decision Processes with Uncertain Transition Models via Incremental Feature Dependency Discovery,” *Machine Learning and Knowledge Discovery in Databases*, edited by P. A. Flach, T. De Bie, and N. Cristianini, Lecture Notes in Computer Science, Springer-Verlag, 2012, Vol. 7524, pp. 99–115.
doi: 10.1007/978-3-642-33486-3_7
- [54] De Bruyne, F., and Carrette, P., “Synthetic Generation of the Gradient for an Iterative Controller Optimization Method,” *Proceedings of 4th European Control Conference (ECC’97)*, Brussels, Belgium, 1997, paper THA-F2, pp. 1–6.
- [55] Akerblad, M., Hansson, A., and Wahlberg, B., “Automatic Tuning for Classical Step-Response Specifications Using Iterative Feedback Tuning,” *Proceedings of 39th IEEE Conference on Decision and Control*, Sydney, NSW, Australia, 2000, Vol. 4, pp. 3347–3348.
doi: 10.1109/CDC.2000.912218
- [56] Park, U. S., Yamada, Y., and Nakabo, Y., “Force Control with Safety Constraints via Iterative Feedback Tuning,” *Proceedings of IEEE International Conference on Robotics and Automation (ICRA ’09)*, Kobe, Japan, 2009, pp. 3670–3675.
doi: 10.1109/ROBOT.2009.5152444
- [57] Kim, B. S., and Calise, A. J., “Nonlinear Flight Control Using Neural Networks,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26–33.
doi: 10.2514/2.4029
- [58] Owens, D. H., Freeman, C. T., and Thanh, V. D., “Norm-Optimal Iterative Learning Control With Intermediate Point Weighting: Theory, Algorithms, and Experimental Evaluation,” *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 3, 2013, pp. 999–1007.
doi: 10.1109/TCST.2012.2196281
- [59] Precup, R.-E., David, R.-C., Petriu, E. M., Preitl, S., and Radac, M.-B., “Fuzzy Control Systems with Reduced Parametric Sensitivity Based on Simulated Annealing,” *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 8, 2012, pp. 3049–3061.
doi: 10.1109/TIE.2011.2130493

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

doi: 10.1080/00207170701611123

[61] Two Rotor Aerodynamical System, User’s Manual. Krakow, Poland: Inteco Ltd., 2007.

[62] Hamamoto, K., Fukuda, T., and Sugie, T., “Iterative Feedback Tuning of Controllers for a Two-Mass Spring System with Friction,” *Proceedings of 39th IEEE Conference on Decision and Control*, Sydney, NSW, Australia, 2000, Vol. 3, pp. 2438–2443.

doi: 10.1109/CDC.2000.914167