2019 27th Mediterranean Conference on Control and Automation (MED)
July 1–4, 2019 · Akko · Israel

Parameterized value iteration for output reference model tracking of a high order nonlinear aerodynamic system*

Timotei Lala and Mircea-Bogdan Radac, *Member, IEEE*

***Abstract*— Linearly and nonlinearly parameterized approximated value iteration (VI) approaches used for output reference model (ORM) tracking control are proposed herein. The ORM problem is of significant interest in practice since, by selecting a linear ORM, the closed-loop control system is indirectly feedback linearized and value iteration (VI) offers the means to achieve this feedback linearization in a model-free manner. We show that a linearly parameterized VI such as the one used for linear systems is still effective for a nonlinear complex process and on similar performance level with that of a neural-network (NN)-based implementation that is more complex and takes significantly more time to learn. While the nonlinearly parameterized NN-based VI proves to be generally more robust to parameters selection, to dataset size and to exploration strategies. The case study is aimed at ORM tracking of a nonlinear two inputs-two outputs aerodynamic process as a representative high dimensional system. Convergence analysis accounting for approximation errors in the VI is also proposed.**

I. Introduction

The output reference model (ORM) tracking problem is of significant interest in practice, especially for nonlinear systems control, since by selection of a linear ORM, feedback linearization is enforced on the controlled process. Then, the closed-loop control system can act linearly in a wide range. Subsequently, linearized control systems are then subjected to higher level learning schemes such as the Iterative Learning Control ones, with practical implications such as primitive-based learning [1].

P01

Suitable ORM selection is not straightforward. It has to be matched with the process bandwidth and with several process nonlinearities such as, e.g., input and output saturations. Additionally, dead-time and non-minimum-phase (NMP) characters of the process cannot be compensated for and must be reflected in the ORM. Apart from this information that can be measured or inferred from working experience with the process, avoiding knowledge of the process' state transition function (process dynamics) – the most time consuming to identify and the most uncertain part of the process – in designing high performance control is very attractive in practice.

Reinforcement Learning (RL) has developed both from the artificial intelligence (AI), and from classical control theory [2]–[5], where it is better known as Approximate (Adaptive, Neuro) Dynamic Programming (ADP). Certain ADP variants can be used to ensure ORM tracking control without knowing the state-space dynamics of the controlled process, which is of high importance into the practice of model-free and data-driven control schemes that are able to compensate for poor modeling and uncertainty in the process. Thus, model-free ADP only uses data collected from the process called state transitions. While plenty of mature ADP schemes already exist in the literature, tuning such schemes requires significant experience. Although successful stories on RL and ADP applied to large state-action spaces are reported mainly with AI [6], in control theory, most approaches use low-order processes as representative case studies and mainly in linear quadratic regulator (LQR)-like settings. While the reference input tracking control problem has been tackled before for linear time-invariant (LTI) processes, known as Linear Quadratic Tracking (LQT) [7], [8], model-free ORM tracking for nonlinear processes was rarely addressed [9], [10].

The iterative model-free approximate Value Iteration (IMF-AVI) proposed in this work belongs to the family of batch-fitted Q-learning schemes [11] also known to the ADP community as action-dependent heuristic dynamic programming (ADHDP), popular and representative ADP approaches owing to their simplicity and model-free character. These schemes have been implemented in many variants: online vs. offline, adaptive or batch, for discrete/continuous states and actions, with/without function approximators, such as Neural Networks (NNs).

Suitable exploration that covers well the state-action space is not trivially ensured but it is critical to ADP control. Randomly generated control input signals will almost surely fail to guide the exploration in the entire state-action space, at least not in a reasonable amount of time. Then, a priori designed feedback controllers can be used under a variable reference input serving to guide the exploration [9]. However, such input-output (IO) or input-state feedback controllers were traditionally not to be designed without using a process model, until the advent of data-driven model-free controller design techniques that have appeared from the field of control theory: Virtual Reference Feedback Tuning (VRFT) [12], Iterative Feedback Tuning [13], data-driven Iterative Learning Control [1], [14], Model Free (Adaptive) Control [15], [16].

The case study deals with the challenging ORM tracking control for a nonlinear real-world two-inputs two-outputs aerodynamic process (TITOAP) having six natural states that are extended with four additional ones according to the proposed theory. The process uses aerodynamic thrust to create vertical (pitch) and horizontal (azimuth) motion. It is shown that IMF-AVI can be used to attain ORM tracking of first order lag type, despite the high order of the

* T. Lala and M.-B. Radac are with the Politehnica University of Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, 300223 Timisoara, Romania (phone: +40 256403240, fax: +40 256403214; e-mail: timotei.lala@student.upt.ro, mircea.radac@upt.ro.

multivariable process, and despite the pitch motion being naturally oscillatory and the azimuth motion practically behaving close to an integrator. The state transitions dataset is collected under the guidance of an input-output (IO) feedback controller designed using model-free VRFT. To the best of authors' knowledge, the ORM tracking context with linear parameterizations was not studied before for high-order nonlinear real-world processes. Moreover, theoretical analysis shows convergence of the IMF-AVI while accounting for approximation errors.

Section II formulates the ORM tracking control problem, while Section III solves it using an IMF-AVI approach. Section IV validates the approach on the TITOAP.

II. Model Reference Control for Unknown Nonlinear Processes

P02

A. The Process

A discrete-time nonlinear unknown open-loop stable minimum-phase (MP) state-space deterministic strictly causal process is defined as

$$P:\begin{cases}\mathbf{x}_{k+1}=\mathbf{f}(\mathbf{x}_k,\mathbf{u}_k),\\ \mathbf{y}_k=\mathbf{g}(\mathbf{x}_k),\end{cases} \tag{1}$$

where k indexes the discrete time, $\mathbf{x}_k=[x_{k,1}\dots x_{k,n}]^T\in\Omega_X\subset\Re^n$ is the n-dimensional state vector, $\mathbf{u}_k=[u_{k,1},\dots,u_{k,m_u}]^T\in\Omega_U\subset\Re^{m_u}$ is the control input signal, $\mathbf{y}_k=[y_{k,1},\dots,y_{k,p}]^T\in\Omega_Y\subset\Re^p$ is the measurable controlled output, $\mathbf{f}:\Omega_X\times\Omega_U\to\Omega_X$ is an *unknown* nonlinear system function continuously differentiable within its domain, $\mathbf{g}:\Omega_X\to\Omega_Y$ is an *unknown* nonlinear continuously differentiable output function. Initial conditions are not accounted for at this point. Let known Ω_U,Ω_Y and unknown Ω_X domains be compact convex. Equation (1) is a general un-restrictive form for most controlled processes. Two widely used data-driven assumptions are:

A1: (1) is fully state controllable with measurable states.
A2: (1) is input-to-state stable on known domain $\Omega_U\times\Omega_X$.

A1 and *A2* are common to data-driven control, not verifiable with unknown model (1), but derivable from literature and from working experience with the process. If above information is not deducible, the user can try process control under the safety operating conditions managed by the supervisory control. Input to state stability (*A2*) is mandatory if open-loop input-state samples are collected to be used for learning state feedback control. *A2* can be omitted if a stabilizing state-fedback controller exists and it is used just for input-state data collection.

P03

B. ORM tracking problem formulation

Let the discrete-time known open-loop stable minimum-phase (MP) state-space deterministic strictly causal ORM be

$$ORM:\begin{cases}\mathbf{x}^m_{k+1}=\mathbf{f}^m(\mathbf{x}^m_k,\mathbf{r}_k),\\ \mathbf{y}^m_k=\mathbf{g}^m(\mathbf{x}^m_k),\end{cases} \tag{2}$$

where $\mathbf{x}^m_k=[x^m_{k,1},\dots,x^m_{k,n_m}]^T\in\Omega_{X_m}\subset\Re^{n_m}$ is the state vector of the ORM, $\mathbf{r}_k=[r_{k,1},\dots,r_{k,p}]^T\in\Omega_{R_m}\subset\Re^p$ is the reference input signal, $\mathbf{y}^m_k=[y^m_{k,1},\dots,y^m_{k,p}]^T\in\Omega_{Y_m}\subset\Re^p$ is the ORM's output, $\mathbf{f}^m:\Omega_{X_m}\times\Omega_{R_m}\to\Omega_{X_m}$, $\mathbf{g}^m:\Omega_{X_m}\to\Omega_{Y_m}$ are *known* nonlinear mappings. Initial conditions are zero unless stated otherwise. Note that $\mathbf{r}_k,\mathbf{y}_k,\mathbf{y}^m_k$ have size p for square feedback CSs. If the ORM (2) is LTI, it is always possible to express the ORM as an IO LTI transfer matrix $\mathbf{y}^m_k=\mathbf{M}(z)\mathbf{r}_k$, where $\mathbf{M}(z)$ is commonly an asymptotically stable unit gain rational transfer matrix and $\mathbf{r}_k$ is the reference input that drives both the feedback CS and the ORM. We introduce an extended process comprising of the process (1) coupled with the ORM (2). For this, the reference input $\mathbf{r}_k$ is treated as a set of measurable exogenous signals (possibly seen as disturbance) that evolve as $\mathbf{r}_{k+1}=\mathbf{h}^m(\mathbf{r}_k)$, with known nonlinear $\mathbf{h}^m:R^m\to R^m$. $\mathbf{h}^m(.)$ is as a generative model for the reference input.

P04

Consider next that the extended state-space model that consists of (1), (2) and the state-space generative model of the reference input signal is, in the most general form:

$$\mathbf{x}^E_{k+1}=\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{x}^m_{k+1}\\ \mathbf{r}_{k+1}\end{bmatrix}=\begin{bmatrix}\mathbf{f}(\mathbf{x}_k,\mathbf{u}_k)\\ \mathbf{f}^m(\mathbf{x}^m_k,\mathbf{r}_k)\\ \mathbf{h}^m(\mathbf{r}_k)\end{bmatrix}=\mathbf{E}(\mathbf{x}^E_k,\mathbf{u}_k),\mathbf{x}^E_k\in\Omega_{X^E}, \tag{3}$$

where $\mathbf{x}^E_k$ is called the extended state vector. Note that the extended state-space fulfils the Markov property. The ORM tracking problem is defined in an optimal control framework. Thus, the infinite horizon cost function (c.f.) to be minimized starting with $\mathbf{x}_0$ is [4]

$$\$_{MR}(\mathbf{x}^E_0,\boldsymbol{\theta})=\sum_{k=0}^{\infty}\gamma^k\left\|\mathbf{y}^m_k(\mathbf{x}^E_k)-\mathbf{y}_k(\mathbf{x}^E_k,\mathbf{u}_k,\boldsymbol{\theta})\right\|_2^2=\sum_{k=0}^{\infty}\gamma^k\left\|\boldsymbol{\varepsilon}_k(\mathbf{x}^E_k,\mathbf{u}_k,\boldsymbol{\theta})\right\|_2^2. \tag{4}$$

In (4), the discount $0<\gamma\le1$ sets the controller's horizon, $\gamma<1$ is usually used to guarantee learning convergence to optimal control. $\|\mathbf{x}\|_2=\sqrt{\mathbf{x}^T\mathbf{x}}$ is the Euclidean norm of the column vector $\mathbf{x}$, $\mathcal{U}_{MR}(\mathbf{x}^E_k,\mathbf{u}_k)=\left\|\mathbf{y}^m_k(\mathbf{x}^E_k)-\mathbf{y}_k(\mathbf{x}^E_k,\mathbf{u}_k)\right\|_2^2>0$ is the stage cost where measurable $\mathbf{y}_{k+1}$ depends via unknown $\mathbf{g}()$ on $\mathbf{x}_k,\mathbf{u}_k$ ((1) is strictly causal) and $\mathcal{U}_{MR}$ penalizes the deviation of $\mathbf{y}_{k+1}$ from the ORM's output $\mathbf{y}^m_{k+1}$. $\boldsymbol{\theta}\in\Re^{n_\theta}$ parameterizes a nonlinear state-feedback admissible controller [4] defined as $\mathbf{u}_k\overset{def}{=}C(\mathbf{x}^E_k,\boldsymbol{\theta})$, which used in (3) makes all CS's trajectories depend on $\boldsymbol{\theta}$. Any stabilizing controller sequence (or controller) rendering a finite c.f. is called *admissible*. A finite $\$_{MR}$ holds if $\boldsymbol{\varepsilon}_k$ is a square-summable sequence, ensured by an asymptotically stabilizing controller if $\gamma=1$ or by a stabilizing controller if $\gamma<1$.

P05

$\$_{MR}(\boldsymbol{\theta})$ in (4) is the value function of using controller $C(\boldsymbol{\theta})$. The optimal controller $\mathbf{u}_k^* = C^*(\mathbf{x}_k^E, \boldsymbol{\theta}^*)$ minimizing (4) is

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \$_{MR}(\mathbf{x}_0^E, \boldsymbol{\theta}). \quad (5)$$

Nonlinear ORM tracking can be attempted, however, an LTI ORM forces a very desirable indirect feedback CS linearization, where the LTI CS's behavior is well extrapolated in a wide range [1]. Therefore, the ORM tracking problem's purpose herein, is to ensure $\mathcal{U}_{MR} \approx 0$ when $\mathbf{r}_k$ drives both the CS and the ORM.

As classical control guidelines, the process time delay and non-minimum-phase (NMP) character should also be contained in $\mathbf{M}(z)$. Still, $\mathbf{M}(z)$'s NMP zeroes render it non-invertible and complicates the subsequent VRFT IO control design [17], motivating the MP assumption on the process.

P06

Depending on the learning scenario, the user may select a piece-wise constant generative model for the reference input signal such as $\mathbf{r}_{k+1} = \mathbf{r}_k$, or a ramp-like model, a sine-like model, etc. In all cases, the states of the generative model are known, measurable and need to be introduced in the extended state vector, to fulfil the Markov property of the extended state-space model. For ORM tracking practical applications, the CS's outputs are required to track the ORM's outputs when both the ORM and the CS are driven by the piece-wise constant reference input signal captured by the generative model $\mathbf{r}_{k+1} = \mathbf{r}_k$. This model will be used herein for learning ORM tracking controllers.

III. Solving the ORM Tracking Problem

For unknown extended process dynamics (3), minimization of (4) will be attempted by an iterative model-free approximate Value Iteration (IMF-AVI). A c.f. that extends $\$_{MR}(\mathbf{x}_k^E)$ called the Q-function (or action-value function) is first defined for each state-action pair. Let the Q-function of acting as $\mathbf{u}_k$ in state $\mathbf{x}_k^E$ and then following the control (policy) $\mathbf{u}_k = C(\mathbf{x}_k^E)$ be

$$\Theta^C(\mathbf{x}_k^E, \mathbf{u}_k) = \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \Theta^C(\mathbf{x}_{k+1}^E, C(\mathbf{x}_{k+1}^E)). \quad (6)$$

The optimal Q-function $\Theta^*(\mathbf{x}_k^E, \mathbf{u}_k)$ corresponding to the optimal controller obeys Bellman's optimality equation

$$\Theta^*(\mathbf{x}_k^E, \mathbf{u}_k) = \min_{C(.)} \left\{ \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \Theta^*(\mathbf{x}_{k+1}^E, C(\mathbf{x}_{k+1}^E)) \right\}, \quad (7)$$

where the optimal controller and optimal Q-functions are

$$\mathbf{u}_k^* = C^*(\mathbf{x}_k^E) = \arg\min_C \Theta^C(\mathbf{x}_k^E, \mathbf{u}_k), \Theta^*(\mathbf{x}_k^E, \mathbf{u}_k) = \min_{C(.)} \Theta^C(\mathbf{x}_k^E, \mathbf{u}_k). \quad (8)$$

Then, for $\$_{MR}^*(\mathbf{x}_k^E) = \min_{\mathbf{u}} \$_{MR}(\mathbf{x}_k^E, \mathbf{u})$ it follows that $\$_{MR}^*(\mathbf{x}_k^E) = \Theta^*(\mathbf{x}_k^E, \mathbf{u}_k^* = C^*(\mathbf{x}_k^E))$. Implying that finding Θ^* is equivalent to determining the optimal c.f. $\$_{MR}^*$.

The optimal Q-function and optimal controller can be found using either Policy Iteration (PoIt) or Value Iteration (VI) strategies. For continuous state-action spaces, IMF-AVI is one possible solution, using different linear and/or nonlinear parameterizations for the Q-function and/or for the controller. NNs are most widely used as nonlinearly parameterized function approximators. As it is well-known, VI alternates two steps: the Q-function estimate update step and the controller improvement step. For example, linear parameterizations of the Q-function allow analytic calculation of the improved controller as in

$$\tilde{C}(\mathbf{x}_k^E, \boldsymbol{\pi}) = \arg\min_C \Theta^C(\mathbf{x}_k^E, \mathbf{u}_k, \boldsymbol{\pi}), \quad (9)$$

by directly minimizing $\Theta^C(\mathbf{x}_k^E, \mathbf{u}_k, \boldsymbol{\pi})$ w.r.t. $\mathbf{u}_k$, where the parameterization $\boldsymbol{\pi}$ is moved from the controller into the Q-function. In these special case, it is possible to eliminate the controller approximator and use only one for the Q-function Θ. Then, given a dataset D of transition samples, $D = \{(\mathbf{x}_k^E, \mathbf{u}_k, \mathbf{x}_{k+1}^E)\}, k = \overline{1,N}$ the IMF-AVI amounts to solving the following optimization problem (OP) at each iteration

$$\boldsymbol{\pi}_{j+1} = \arg\min_{\boldsymbol{\pi}} \sum_{k=1}^{N} \left(\Theta(\mathbf{x}_k^E, \mathbf{u}_k, \boldsymbol{\pi}) - \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) - \gamma \Theta(\mathbf{x}_{k+1}^E, \tilde{C}(\mathbf{x}_{k+1}^E, \boldsymbol{\pi}_j), \boldsymbol{\pi}_j) \right)^2 \quad (10)$$

which is a Bellman residual minimization problem where the (usually separate) controller improvement step is now embedded inside the OP (10).

For a linear parameterization $\Theta(\mathbf{x}_k^E, \mathbf{u}_k, \boldsymbol{\pi}) = \boldsymbol{\Phi}^T(\mathbf{x}_k^E, \mathbf{u}_k)\boldsymbol{\pi}$ using a set of n_Φ basis functions of the form $\boldsymbol{\Phi}^T(\mathbf{x}_k^E, \mathbf{u}_k) = [\Phi_1(\mathbf{x}_k^E, \mathbf{u}_k), ..., \Phi_{n_\Phi}(\mathbf{x}_k^E, \mathbf{u}_k)]$, the least squares solution to (10) is equivalent to solving the following overdetermined linear system of equations w.r.t. $\boldsymbol{\pi}_{j+1}$:

$$\begin{bmatrix} \boldsymbol{\Phi}^T(\mathbf{x}_1^E, \mathbf{u}_1) \\ \dots \\ \boldsymbol{\Phi}^T(\mathbf{x}_N^E, \mathbf{u}_N) \end{bmatrix} \boldsymbol{\pi}_{j+1} = \begin{bmatrix} \mathcal{U}(\mathbf{x}_1^E, \mathbf{u}_1) + \gamma \boldsymbol{\Phi}^T\left(\mathbf{x}_2^E, \tilde{C}(\mathbf{x}_2^E, \boldsymbol{\pi}_j)\right)\boldsymbol{\pi}_j \\ \dots \\ \mathcal{U}(\mathbf{x}_N^E, \mathbf{u}_N) + \gamma \boldsymbol{\Phi}^T\left(\mathbf{x}_{N+1}^E, \tilde{C}(\mathbf{x}_{N+1}^E, \boldsymbol{\pi}_j)\right)\boldsymbol{\pi}_j \end{bmatrix}. \quad (11)$$

Starting with an initial parameter $\boldsymbol{\pi}_0$ of the Q-function, the IMF-AVI that allows explicit controller improvement calculation as in (9), embeds both VI steps into solving (11). Linearly parameterized IMF-AVI (LP-IMF-AVI) are validated in the case study and compared to nonlinearly parameterized IMF-AVI (NP-IMF-AVI). IMF-AVI convergence is next analyzed under approximation errors.

A. IMF-AVI convergence with approximation errors

The proposed iterative model-free VI-based Q-learning *Algorithm 1* consists of the next steps:

S1. Select an initial (not necessarily admissible) controller C_0 and an initialization value $\Theta_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$ of the Q-function. Initialize iteration index $j = 1$.

S2. Use the one step back-up equation for the Q-function

$$\begin{aligned} \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) &= \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \Theta_{j-1}(\mathbf{x}_{k+1}^E, C_{j-1}(\mathbf{x}_{k+1}^E)) \\ &= \min_{\mathbf{u}} \{ \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \Theta_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u}) \}. \end{aligned} \quad (12)$$

S3. Improve the controller using the equation

P12

$$C_j(\mathbf{x}_k^E) = \arg\min_{\mathbf{u}} \Theta_j(\mathbf{x}_k^E, \mathbf{u}) . \quad (13)$$

S4. Set $j = j+1$ and repeat steps S2, S3, until convergence.

P07

Lemma 1. For an arbitrary sequence of controllers $\{\kappa_j\}$, define the VI-like update for extended c.f. ξ_j as [18]

$$\xi_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k) = \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \xi_j(\mathbf{x}_{k+1}^E, \kappa_j(\mathbf{x}_{k+1}^E)) . \quad (14)$$

If $\Theta_0(\mathbf{x}_k^E, \mathbf{u}_k) = \xi_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0$, then $\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) \le \xi_j(\mathbf{x}_k^E, \mathbf{u}_k)$.

Proof. For limited space, see [21].

P08

Lemma 2. For the sequence $\{\Theta_j\}$ from (12), under controllability assumption *A1*, it is valid that:

1) $0 \le \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) \le B(\mathbf{x}_k^E, \mathbf{u}_k)$ with $B(\mathbf{x}_k^E, \mathbf{u}_k)$ an upper bound.

2) If there exists a solution $\Theta^*(\mathbf{x}_k^E, \mathbf{u}_k)$ to (8), then $0 \le \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) \le \Theta^*(\mathbf{x}_k^E, \mathbf{u}_k) \le B(\mathbf{x}_k^E, \mathbf{u}_k)$.

Proof. For limited space, see [21].

P09

Theorem 1. For the extended process (3) with c.f. (4), under *A1*, *A2*, with the sequences $\{C_j\}$ and $\{\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ generated by the Q-learning *Algorithm 1*, it is true that:

1) $\{\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ is a non-decreasing sequence for which $\Theta_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k) \ge \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)$ holds, $\forall j, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$ and

2) $\lim_{j\to\infty} C_j = C^*$ and $\lim_{j\to\infty} \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) = \Theta^*(\mathbf{x}_k^E, \mathbf{u}_k)$.

Proof. For limited space, see [21].

P10

Comment 2. (12) is practically solved in the sense of the OP (10) (either as a linear or nonlinear regression) using a batch (dataset) of transition samples collected from the process using any controller, i.e. in "off-policy" mode. While the step (13) can be solved either as a regression or explicitly analytically when the expression of $\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)$ allows it. Moreover, (12) and (13) can be solved batch-wise either online or offline. When the batch of transition samples is updated each sample time, the VI-scheme becomes adaptive.

P11

Comment 3. *Theorem 1* proves the VI-based learning convergence of the sequence of Q-functions $\lim_{j\to\infty} \Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) = \Theta^*(\mathbf{x}_k^E, \mathbf{u}_k)$ assuming that the true Q-function parameterization is used. In practice, this is rarely possible, such as, e.g. in the case of LTI systems. For general nonlinear processes of type (1), different function approximators are employed for the Q-function, most commonly using NNs. Then the convergence of the VI Q-learning scheme is to a suboptimal controller and to a suboptimal Q-function, owing to the approximation errors. A convergence proof of the learning scheme under approximation errors is next shown and accounts for generic parameterizations of the Q-function [19].

Let the IMF-AVI *Algorithm 2* consists of the steps:

S1. Select an initial (not necessarily admissible) controller $\widetilde{C}_0$ and an initialization value $\widetilde{\Theta}_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$ of the Q-function. Initialize iteration $j = 1$.

S2. Use the update equation for the approximate Q-function

$$\widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k) = \mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \widetilde{\Theta}_{j-1}(\mathbf{x}_{k+1}^E, \widetilde{C}_{j-1}(\mathbf{x}_{k+1}^E)) + \delta_j(\mathbf{x}_k^E, \mathbf{u}_k) \quad (15)$$
$$= \min_{\mathbf{u}} \{\mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \widetilde{\Theta}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} + \delta_j .$$

S3. Improve the approximate controller using

$$\widetilde{C}_j(\mathbf{x}_k^E) = \arg\min_{\mathbf{u}} \widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}) . \quad (16)$$

S4. Set $j = j+1$ and repeat steps S2, S3, until convergence.

Comment 4. In *Algorithm 2*, the sequences $\{\widetilde{C}_j(\mathbf{x}_k^E)\}$ and $\{\widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ are approximations of the true sequences $\{C_j(\mathbf{x}_k^E)\}$ and $\{\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$. Since the true Q-function and controller parameterizations are not known, (15) must be solved in the sense of the OP (10) with respect to the unknown $\widetilde{\Theta}_j$, in order to minimize the residuals δ_j at each iteration. If the true parameterizations of the Q-function and of the controller were known, then $\delta_j = 0$ and the IMF-AVI updates (15), (16) coincide with (12), (13), respectively. Next, let the following assumption hold.

A3. There exist two positive scalar constants $\underline{\psi}, \overline{\psi}$ such that $0 < \underline{\psi} \le 1 \le \overline{\psi} < \infty$, ensuring

$$\min_{\mathbf{u}} \{\underline{\psi}\mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \widetilde{\Theta}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} \le \widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k) \le \quad (17)$$
$$\le \min_{\mathbf{u}} \{\overline{\psi}\mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \widetilde{\Theta}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} .$$

Comment 5. Inequalities from (17) account for nonzero positive or negative residuals δ_j, i.e. for the approximation errors in the Q-function, since $\widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k)$ can over- or under-estimate $\min_{\mathbf{u}} \{\mathcal{U}(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \widetilde{\Theta}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\}$ in (15). $\underline{\psi}, \overline{\psi}$ can span large intervals ($\underline{\psi}$ close to 0 and $\overline{\psi}$ very large). Hope is that, if $\underline{\psi}, \overline{\psi}$ are close to 1 -meaning low approximation errors-, then the entire IMF-AVI process preserves $\delta_j \approx 0$. In practice, this amounts to using high performance approximators. For example, with NNs, adding more layers and more neurons, enhances the approximation capability and theoretically reduces the residuals in (15).

Theorem 2. Let the sequences $\{\widetilde{C}_j(\mathbf{x}_k^E)\}$ and $\{\widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ evolve as in (15), (16), the sequences $\{C_j(\mathbf{x}_k^E)\}$ and $\{\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ evolve as in (12), (13). Initialize $\widetilde{\Theta}_0(\mathbf{x}_k^E, \mathbf{u}_k) = \Theta_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$ and let *A3* hold. Then

$$\underline{\psi}\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) \le \widetilde{\Theta}_j(\mathbf{x}_k^E, \mathbf{u}_k) \le \overline{\psi}\Theta_j(\mathbf{x}_k^E, \mathbf{u}_k) \quad (18)$$

Proof. For limited space, see [21].

P13

Comment 6. Theorem 2 shows that the trajectory of $\{\widetilde{\Theta}_j(\mathbf{x}_k^E,\mathbf{u}_k)\}$ closely follows that of $\{\Theta_j(\mathbf{x}_k^E,\mathbf{u}_k)\}$ in a bandwidth set by $\underline{\psi},\overline{\psi}$. It does not ensure that $\{\widetilde{\Theta}_j(\mathbf{x}_k^E,\mathbf{u}_k)\}$ converges to a steady-state value, but in the worst case, it oscillates around $\Theta^*(\mathbf{x}_k^E,\mathbf{u}_k)=\lim_{j\to\infty}\Theta_j(\mathbf{x}_k^E,\mathbf{u}_k)$ in a band that can be made arbitrarily small by using powerful approximators. By minimizing over $\mathbf{u}_k$ both sides of (17), similar conclusions result for the controller sequence $\{\widetilde{C}_j(\mathbf{x}_k^E)\}$ that closely follows $\{C_j(\mathbf{x}_k^E)\}$.

P14

IV. Validation Case Study on The TITOAP

The ORM tracking problem on the more challenging TITOAP position control by Inteco [20] is solved. The azimuth motion acts as an integrator while the pitch positioning is affected differently by the gravity for the up/down motions. Coupling between the channels is present. The process model is [20]:

$$\begin{cases}\dot{\omega}_h=\left(sat(U_h)-M_h(\omega_h)\right)/2.7\cdot10^{-5},\\ \dot{K}_h=\left(0.216F_h(\omega_h)\cos\alpha_v-0.058\Omega_h+0.0178sat(U_v)\cos\alpha_v\right),\\ \Omega_h=K_h/\left(0.0238\cdot\cos^2\alpha_v+3\cdot10^{-3}\right),\\ \dot{\alpha}_h=\Omega_h,\end{cases}\quad(19)$$

$$\begin{cases}\dot{\omega}_v=\left(sat(U_v)-M_v(\omega_v)\right)/1.63\cdot10^{-4},\\ \dot{\Omega}_v=\dfrac{1}{0.03}\begin{pmatrix}0.2F_v(\omega_v)-0.0127\Omega_v-0.0935\sin\alpha_v-\\ 9.28\cdot10^{-6}\Omega_v|\omega_v|+4.17\cdot10^{-3}sat(U_h)-0.05\cos\alpha_v\\ -0.0211\Omega_h^2\sin\alpha_v\cos\alpha_v-0.093\sin\alpha_v+0.05\end{pmatrix},\\ \dot{\alpha}_v=\Omega_v,\end{cases}$$

where $sat(\,)$ is the saturation function on $[-1,1]$, $U_h=u_1$ is the azimuth motion control input, $U_v=u_2$ is the vertical motion control input, $\alpha_h\,(rad)=y_1\in[-\pi,\pi]$ is the azimuth angle, $\alpha_v\,(rad)=y_2\in[-\pi/2,\pi/2]$ is the pitch angle, other states being described in [20], [22]. Nonlinear maps $M_v(\omega_v),F_v(\omega_v),M_h(\omega_h),F_h(\omega_h)$ were polynomially fitted from experimental data for $\omega_v,\omega_h\in(-4000;4000)$ [20].

An equivalent MP discrete-time model of relative degree one at sampling time $T_s=0.1\,\mathrm{s}$ obtained from (1) is suitable for input-state data collection where $\mathbf{x}_k=[\omega_{h,k},\Omega_{h,k},\alpha_{h,k},\omega_{v,k},\Omega_{v,k},\alpha_{v,k}]^T\in\Re^6$, $\mathbf{u}_k=[u_{k,1},u_{k,2}]^T\in\Re^2$. The process's dynamics will *not be used* for learning ORM tracking in the following.

P15

A. Initial linear MIMO controller with model-free VRFT

An initial model-free multivariable 2x2 IO controller is first designed using model-free VRFT, as previously described in [9]. This controller will be used afterwards for input-state transition samples collection. The ORM to be tracked is $\mathbf{M}(z)=\mathrm{diag}(M_1(z),M_2(z))$ where $M_1(z),M_2(z)$ are the discrete-time counterparts of $M_1(s)=M_2(s)=1/(3s+1)$ obtained for a sampling period of $T_s=0.1\,\mathrm{s}$. The VRFT prefilter is chosen as $\mathbf{L}(z)=\mathbf{M}(z)$. A pseudo-random binary signal of amplitude $[-0.1;0.1]$ is used on both inputs $u_{k,1},u_{k,2}$ to open-loop excite the pitch and azimuth dynamics. The IO data $\{\widetilde{\mathbf{u}}_k,\widetilde{\mathbf{y}}_k\}$ is collected with low-amplitude zero-mean inputs $u_{k,1},u_{k,2}$, to maintain the process linearity around the mechanical equilibrium, such that to fit the linear VRFT design framework. The linear VRFT output feedback error diagonal controller is

$$\mathbf{C}(z;\boldsymbol{\theta})=diag\left(P_{11}(z),P_{22}(z)\right)/(1-z^{-1})\quad(20)$$

$$P_{11}(z)=2.9341-5.8689z^{-1}+3.9303z^{-2}-0.9173z^{-3}-0.0777z^{-4},$$
$$P_{22}(z)=0.6228-1.1540\,z^{-1}+0.5467z^{-2},$$

where the parameter vector $\boldsymbol{\theta}$ groups all the coefficients of $P_{11}(z),P_{22}(z)$. The output feedback controller (20) processes the feedback control error $\mathbf{e}_k=\mathbf{r}_k-\mathbf{y}_k$ in closed loop. Nonlinear (in particular, linear) state-feedback controllers can also be found by VRFT as shown in [23] to serve as initializations for the IMF-AVI. Should this not be mandatory, IO feedback controllers should be first designed since they are very data-efficient.

P16

B. Collecting more input-state-output data

ORM tracking is next improved to make the closed loop CS better match the ORM $\mathbf{M}(z)$. With controller (20) used in closed-loop to stabilize the process, input-state-output data is collected for 7000 s. The reference inputs with amplitudes $r_{k,1}\in[-2;2],r_{k,2}\in[-1.4;1.1]$ model successive steps that switch their amplitudes uniformly random at 17 s and 25 s, respectively. On the outputs $u_{k,1},u_{k,2}$ of both controllers $C_{11}(z),C_{22}(z)$, an additive noise is added at every 2nd sample as an uniform random number in $[-1.6;1.6]$ for $C_{11}(z)$ and in $[-1.7;1.7]$ for $C_{22}(z)$. These additive disturbances provide an appropriate exploration, visiting many combinations of input-states-outputs. The computed controller outputs are saturated to $sat(u_{k,1}),sat(u_{k,2})\in[-1;1]$ after which they are sent to the process. The reference inputs $r_{k,1},r_{k,2}$ drive the ORM:

$$\begin{cases}x_{k+1,1}^m=0.9672x_{k,1}^m+0.03278r_{k,1},\\ x_{k+1,2}^m=0.9672x_{k,2}^m+0.03278r_{k,2}\\ \mathbf{y}_k^m=[y_{k,1}^m\;\;y_{k,2}^m]^T=[x_{k,1}^m\;\;x_{k,2}^m]^T.\end{cases}\quad(21)$$

The ORM's states (also ORM's outputs) are collected along with the process' states and controls, in order to build the extended process state (3). Let this extended state be:

$$\mathbf{x}_k^E=[\underbrace{x_{k,1}^m\;x_{k,2}^m}_{(\mathbf{x}_k^m)^T}\;\underbrace{r_{k,1}\;r_{k,2}}_{(\mathbf{r}_k)^T}(\mathbf{x}_k)^T]^T\in\Re^{10}\quad(22)$$

Essentially, the collected $\mathbf{x}_k^E$ and $\mathbf{u}_k$ builds the transitions dataset $D=\{(\mathbf{x}_1^E,\mathbf{u}_1,\mathbf{x}_2^E),\ldots,(\mathbf{x}_{70000}^E,\mathbf{u}_{70000},\mathbf{x}_{70001}^E)\}$ for $N=70000$, used for the IMF-AVI implementation. After collection, an important processing step is performed related

to data normalization. Some states of the process will be replaced by their scaled version. Thus, the transformed process state is $\tilde{\mathbf{x}}_k = [\tilde{\omega}_{h,k} = \omega_{h,k}/7200, \tilde{\Omega}_{h,k} = \Omega_{h,k}/25, \alpha_{h,k}, \tilde{\omega}_{v,k} = \omega_{v,k}/3500, \tilde{\Omega}_{v,k} = \Omega_{v,k}/40, \alpha_{v,k}]^T \in \Re^6$. The reference inputs, the ORM states and the saturated process inputs already have values around $[-1;1]$. The normalized states will finally serve for state feedback.

P17

Note that the reference input signals $r_{k,1}, r_{k,2}$ used as sequences of constant amplitude steps for ensuring good exploration do not have a generative model that obeys the Markov assumption. To avoid this problem, the piece-wise constant reference input generative model $\mathbf{r}_{k+1} = \mathbf{r}_k$ is employed by eliminating from the dataset D all the transition samples that correspond to switching reference inputs instants (i.e., when at least one of $r_{k,1}, r_{k,2}$ switches).

C. Learning control with linearly parameterized IMF-AVI

Details of the linearly parameterized IMF-AVI (LP-IMF-AVI) applied to the ORM tracking control problem are next provided. The stage cost is defined $\mathcal{U}(\mathbf{x}_k^E) = (y_{k,1} - y_{k,1}^m)^2 + (y_{k,2} - y_{k,2}^m)^2$ and the discount factor in $\$_{MR}$ is $\gamma = 0.95$. The Q-function Θ is linearly parameterized using the basis functions

$$\begin{aligned}&\boldsymbol{\Phi}_c^T(\mathbf{x}_k^E, \mathbf{u}_k) = [x_{k,1}^{m\,2}, x_{k,2}^{m\,2}, r_{k,1}^{\;2}, ..., x_{k,6}^{\;2}, u_{k,1}^{\;2}, u_{k,2}^{\;2}, \\ &x_{k,1}^m \cdot x_{k,2}^m, x_{k,1}^m \cdot r_{k,1}, ..., x_{k,1}^m \cdot u_{k,2,}, x_{k,2}^m \cdot r_{k,1}, ..., u_{k,1} \cdot u_{k,2}] \in \Re^{78}.\end{aligned} \tag{23}$$

P18

The controller improvement step at each iteration of the LP-IMF-AVI explicitly minimizes the Q-function. Solving the linear system of equations resulting after setting the derivative of $\Theta(\mathbf{x}_k^E, \mathbf{u}_k)$ w.r.t. $\mathbf{u}_k$ equal to zero, it results

$$\begin{aligned}&\tilde{\mathbf{u}}_k^* = \begin{bmatrix} u_{k,1}^* \\ u_{k,2}^* \end{bmatrix} = \tilde{C}^*(\mathbf{x}_k^E, \boldsymbol{\pi}_j) = \begin{bmatrix} 2\pi_{j,11} & \pi_{j,78} \\ \pi_{j,78} & 2\pi_{j,12} \end{bmatrix}^{-1} \begin{bmatrix} F_1(\mathbf{x}_k^E) \\ F_2(\mathbf{x}_k^E) \end{bmatrix}, \\ &F_1(\mathbf{x}_k^E) = \pi_{j,22} r_{k,1} + \pi_{j,32} r_{k,2} + \pi_{j,41} r_{k,1} + \pi_{j,49} r_{k,2} + \pi_{j,56} x_{k,1} + \\ &\quad + \pi_{j,62} x_{k,2} + \pi_{j,67} x_{k,3} + \pi_{j,71} x_{k,4} + \pi_{j,74} x_{k,5} + \pi_{j,76} x_{k,6}, \\ &F_2(\mathbf{x}_k^E) = \pi_{j,23} r_{k,1} + \pi_{j,33} r_{k,2} + \pi_{j,42} r_{k,1} + \pi_{j,50} r_{k,2} + \pi_{j,57} x_{k,1} + \\ &\quad + \pi_{j,63} x_{k,2} + \pi_{j,68} x_{k,3} + \pi_{j,72} x_{k,4} + \pi_{j,75} x_{k,5} + \pi_{j,77} x_{k,6}.\end{aligned} \tag{24}$$

The improved controller is embedded in the system (11) of 70000 linear equations with 78 unknowns corresponding to the parameters of $\boldsymbol{\pi}_{j+1} \in \Re^{78}$. This linear system (11) is solved as a least squares regression with each of the 50 iterations of the LP-IMF-AVI. The practical convergence results are shown in Fig. 1 for $\|\boldsymbol{\pi}_j - \boldsymbol{\pi}_{j+1}\|_2$ and for the ORM tracking performance in terms of a normalized c.f. $J_{test} = 1/N \cdot (\| y_{k,1} - y_{k,1}^m \|_2 + \| y_{k,2} - y_{k,2}^m \|_2)$ measured for $N = 2000$ samples over 200 s in the test scenario displayed in Fig. 2. The test scenario has of a sequence of piece-wise constant reference inputs that switch at different moments of time for the azimuth and pitch ($y_{k,1}$ and $y_{k,2}$), to illustrate the coupling behavior between the two control channels and the extent by which the learned controller manages to achieve the decoupled behavior requested by the ORM.

P19

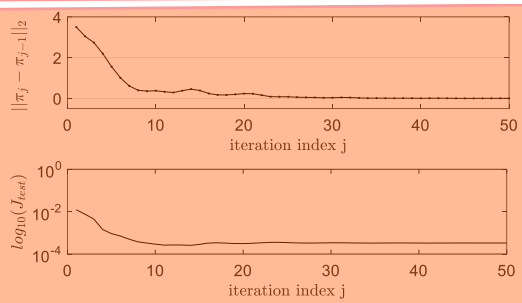

Fig. 1. The LP-IMF-AVI convergence on TITOAP.

The best LP-IMF-AVI controller found over the 50 iterations results in $J_{test} = 0.0017$ (tracking results in black lines in Fig. 2), which is more than 6 times smaller than the tracking performance recorded with the VRFT controller used for transition samples collection, for which $J_{test} = 0.0103$ (tracking results in green lines in Fig. 2).

D. Learning control with nonlinearly parameterized IMF-AVI using NNs

The previous LP-IMF-AVI for ORM tracking control learning scheme is challenged by a nonlinearly parameterized IMF-AVI (NP-IMF-AVI) implementation with NNs. In this case, two NNs are needed to approximate the Q-function and the controller. The procedure follows the NP-IMF-AVI implementation described in [9]; it uses Q-function estimate minimization by enumerating discrete actions [23], [24]. The trained NN controller still outputs continuous actions. The same dataset of transition samples is used as was previously used for the LP-IMF-AVI. The controller NN (C-NN) is a 10–3–2 (10 inputs because $\mathbf{x}_k^E \in \Re^{10}$, 3 neurons in the hidden layer and 2 outputs for $u_{k,1}, u_{k,2}$) with tanh activation function in the hidden layer and linear output activation. The Q-function NN (Q-NN) is 12–25–1 with the same parameters as C-NN. Initial weights of both NNs are uniform random numbers with zero-mean and variance 0.3. Both NNs are to be trained using scaled conjugate gradient for maximum 500 epochs. The available dataset is randomly divided into training (80%) and validation data (20%). Early stopping during training is enforced after 10 increases of the training c.f. mean sum of squared errors (MSSE)) evaluated on the validation data. The ORM tracking with the best NP-IMF-AVI controller producing the lowest $J_{test} = 0.0017$ is shown in Fig. 2.

P20

With the best ORM tracking not better than that with the LP-IMF-AVI controllers, extensive reruns of the NP-IMF-AVI convergent process under different dataset sizes, different exploration strategies and different Q-NN and C-NN architectures always produced converging learning process. The LP-IMF-AVI convergence is more sensitive to the mentioned aspects. The main reason appears to be the under-parameterization of the Q-function, hence the quadratic form may be too limited with more nonlinear processes. This explains for a violation of the low approximation error assumptions of *Theorem 2*. Both LP-IMF-AVI and NP-IMF-AVI well linearize the CS to ensure

ORM tracking [25], recommending further application of data-driven ILC [26] for primitive-based learning [27].

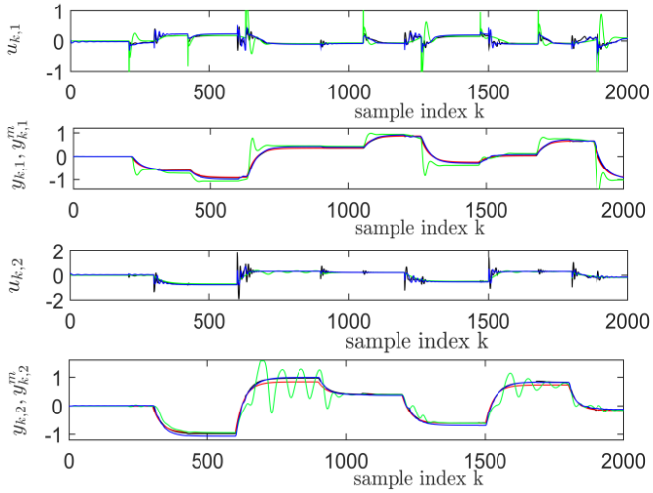

Fig. 2. The IMF-AVI convergence on TITOAP: $y_{k,1}^m$, $y_{k,2}^m$ (red); $u_{k,1}$, $u_{k,2}$, $y_{k,1}$, $y_{k,2}$ for LP-IMF-AVI (black), for NP-IMF-AVI with NNs (blue), for the initial VRFT controller used for transitions collection (green).

P21

V. Conclusion

This paper proves an IMF-AVI ADP scheme for the challenging ORM tracking of a high-order real-world complex nonlinear process with unknown. Learning high performance state-feedback control under the model-free mechanism offered by ADP builds upon the input-states-outputs transition samples collected with a model-free linear output feedback controller designed using VRFT.

References

[1] M.-B. Radac, R.-E. Precup, and E. M. Petriu, "Model-free primitive-based iterative learning control approach to trajectory tracking of MIMO systems with experimental validation," *IEEE Trans. Neural Netw. Learning Syst*., vol. 26, no. 11, pp. 2925–2938, Nov. 2015.

[2] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, 2009.

[3] F. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[4] F. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circ. Syst. Mag.*, vol. 9, no. 3, pp. 76–105, Aug. 2009.

[5] D. Wang, D. Liu, Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, Feb. 2012.

[6] V. Mnih, K. Kavukcouglu, D. Silver, A. A. Rusu, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb 2015.

[7] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani and A. Karimpour, "Optimal Tracking Control of Unknown Discrete-Time Linear Systems Using Input-Output Measured Data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, 2015.

[8] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.

[9] M.-B. Radac, R.-E. Precup and R.-C. Roman, "Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-learning," *ISA Trans*., vol. 73, pp. 227–238, Feb. 2018.

[10] Z. Wang, R. Lu, F. Gao, and D. Liu, "An indirect data-driven method for trajectory tracking control of a class of nonlinear discrete-time systems," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4121–4129, 2017.

[11] R. Hafner and M. Riedmiller, "Reinforcement learning in feedback control. Challenges and benchmarks from technical process control," *Mach. Learn.*, vol. 84, no. 1, pp. 137–169, July 2011.

[12] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, Aug. 2002.

[13] H. Hjalmarsson, "Iterative feedback tuning - an overview," *Int. J. Adapt. Control Signal Process.*, vol. 16, pp. 373–395, June 2002.

[14] R. Chi, Z.-S. Hou, S. Jin, and B. Huang, "An improved data-driven point-to-point ILC using additional on-line control inputs with experimental verification," *IEEE Trans. Syst., Man, Cybern.: Syst*., vol. 49, no. 4, pp. 687–696, 2019.

[15] H. Abouaïssa, M. Fliess, and C. Join, "On ramp metering: towards a better understanding of ALINEA via model-free control," *Int. J. Control*, vol. 90, no. 5, pp. 1018–1026, May 2017.

[16] Z.-S. Hou, S. Liu, and T. Tian, "Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1914–1928, Aug. 2017.

[17] L. Campestrini, D. Eckhard, M. Gevers, and A. Bazanella, "Virtual reference feedback tuning for non-minimum phase plants," *Automatica*, vol. 47, no. 8, pp. 1778–1784, Aug. 2011.

[18] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamics programming: convergence proof," *IEE Trans. Syst., Man, Cybern.: Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[19] A. Rantzer, "Relaxed dynamic programming in switching systems," *IEE Proc. – Control Theory & Appl.*, vol. 153, no. 5, pp. 567–574, 2006.

[20] http://ee.sharif.edu/~linearcontrol/Files/Lab/tras_um.pdf.

[21] https://drive.google.com/open?id=1pdEelk3i43WmWTboMZvtbRBR7mLobxkk .

[22] M.-B. Radac, R.-E. Precup and R.-C. Roman, "Model-free control performance improvement using virtual reference feedback tuning and reinforcement Q-learning," *Int. J. Syst. Sci*., vol. 48, no. 5, pp. 1071–1083, Apr. 2017.

[23] M.-B. Radac and R.-E. Precup, "Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning," *Neurocomput.*, vol. 275, pp. 317–329, Jan. 2018.

[24] M.-B. Radac and R.-E. Precup "Data-driven MIMO model-free reference tracking control with nonlinear state-feedback and fractional order controllers," *Appl. Soft Computing*., vol. 73, pp. 992–1003, Dec. 2018.

[25] M.-B. Radac and R.-E. Precup, "Data-Driven model-free tracking reinforcement learning control with VRFT-based adaptive actor-critic," *Appl. Sci.,* vol. 9, no. 9, 1807, 2019.

[26] M.-B. Radac and R.-E. Precup, "Model-free constrained data-driven iterative reference input tuning algorithm with experimental validation," *Int. J. Gen. Syst.,* vol. 45, no. 4, pp. 455–476, 2016.

[27] M.-B Radac and R.-E. Precup, "Three-level hierarchical model-free learning approach to trajectory tracking control," *Eng. Appl. Artif. Intell*., vol. 55, pp. 103–118, Oct. 2016.