Asociaţia Grupul pentru Reformă şi Alternativă Universitară (GRAUR)
Cluj-Napoca
Indexul Operelor Plagiate în România
www.plagiate.ro

Decizie de indexare a faptei de plagiat la poziţia
00428 / 01.10.2019
şi pentru admitere la publicare în volum tipărit

care se bazează pe:

A. Nota de constatare şi confirmare a indiciilor de plagiat prin fişa suspiciunii inclusă în decizie.

Fişa suspiciunii de plagiat / Sheet of plagiarism's suspicion		
Opera suspicionată (OS)		Opera autentică (OA)
Suspicious work		Authentic work
OS	LALA, Timotei and RADAC, Mircea-Bogdan. Parameterized value iteration for output reference model tracking of a high order nonlinear aerodynamic system. Proceedings of 27th Mediterranean Conference on Control and Automation (MED19), Akko, Israel, July 1-4, 2019. pp. 43-49, DOI: 10.1109/MED.2019.8798580. Presented: 1 - 4 July 2019, Published: 15 August 2019.	
OA	RADAC, Mircea-Bogdan and LALA, Timotei, Learning Output Reference Model Tracking for Higher-Order Nonlinear Systems with Unknown Dynamics. *Algorithms.* 2019, 12, 121, DOI: 10.3390/a12060121, Received: 1 May 2019, Accepted: 9 June 2019, Published: 12 June 2019.	
Incidenţa minimă a suspiciunii / Minimum incidence of suspicion		
P01:	p.43:10s – p.43:27d	p.01:09 – p.02:28
P02:	p.44:16s – p.44:34s	p.03:17 – p.03:30
P03:	p.44:45s – p.44:18d	p.04:04 – p.04:18
P04:	p.44:19d – p.44:28d	p.04:33 – p.04:39
P06:	p.45:15s – p.46:02d	p.05:25 – p.07:08
P11:	p.46:31s – p.46:00s	p.09:11 – p.09:18
P12:	p.46:01d – p.46:00d	p.09:21 – p.10:10
P13:	p.47;01s – p.47:09s	p.10:22 – p.10.26
P15:	p.47;31s – p.47:11d	p.13:13 – p.14:03
P16:	p.47:19d – p.47:39d	p.14:17 – p.15:11
P17:	p.48:08s – p.48:23s	p.15:14 – p.16:05
P18:	p.48;25s – p.48:08d	p.16:11 – p.17:04
P19:	p.48: Fig.1	p.16:Figure 6
P20:	p.48:09d – p.48:32d	p.17:05 – p.17:21
P21:	p.49:04s – p.49:10s	p.21:04 – p.21:10
Fişa întocmită pentru includerea suspiciunii în Indexul Operelor Plagiate în România de la Sheet drawn up for including the suspicion in the Index of Plagiarized Works in Romania at www.plagiate.ro		

Notă: Prin „p.72:00" se înţelege paragraful care se termină la finele pag.72. Notaţia „p.00:00" semnifică până la ultima pagină a capitolului curent, în întregime de la punctul iniţial al preluării.

Note: By „p.72:00" one understands the text ending with the end of the page 72. By „p.00:00" one understands the taking over from the initial point till the last page of the current chapter, entirely.

B. Fişa de argumentare a calificării de plagiat alăturată, fişă care la rândul său este parte a deciziei.

Echipa Indexului Operelor Plagiate în România

Asociaţia Grupul pentru Reformă şi Alternativă Universitară (GRAUR)
Cluj-Napoca
Indexul Operelor Plagiate în România
www.plagiate.ro

Fişa de argumentare a calificării

Nr. crt.	Descrierea situaţiei care este încadrată drept plagiat	Se confirmă
1.	Preluarea identică a unor pasaje (piese de creaţie de tip text) dintr-o operă autentică publicată, fără precizarea întinderii şi menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	✓
2.	Preluarea a unor pasaje (piese de creaţie de tip text) dintr-o operă autentică publicată, care sunt rezumate ale unor opere anterioare operei autentice, fără precizarea întinderii şi menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
3.	Preluarea identică a unor figuri (piese de creaţie de tip grafic) dintr-o operă autentică publicată, fără menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	✓
4.	Preluarea identică a unor tabele (piese de creaţie de tip structură de informaţie) dintr-o operă autentică publicată, fără menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
5.	Republicarea unei opere anterioare publicate, prin includerea unui nou autor sau de noi autori fără contribuţie explicită în lista de autori	
6.	Republicarea unei opere anterioare publicate, prin excluderea unui autor sau a unor autori din lista iniţială de autori.	
7.	Preluarea identică de pasaje (piese de creaţie) dintr-o operă autentică publicată, fără precizarea întinderii şi menţionarea provenienţei, fără nici o intervenţie personală care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	✓
8.	Preluarea identică de figuri sau reprezentări grafice (piese de creaţie de tip grafic) dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
9.	Preluarea identică de tabele (piese de creaţie de tip structură de informaţie) dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
10.	Preluarea identică a unor fragmente de demonstraţie sau de deducere a unor relaţii matematice care nu se justifică în regăsirea unei relaţii matematice finale necesare aplicării efective dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
11.	Preluarea identică a textului (piese de creaţie de tip text) unei lucrări publicate anterior sau simultan, cu acelaşi titlu sau cu titlu similar, de un acelaşi autor / un acelaşi grup de autori în publicaţii sau edituri diferite.	
12.	Preluarea identică de pasaje (piese de creaţie de tip text) ale unui cuvânt înainte sau ale unei prefeţe care se referă la două opere, diferite, publicate în două momente diferite de timp.	

Alte argumente particulare: a) Preluările de poze nu indică sursa, locul unde se află, autorul real sau posibil.

Notă:

a) Prin „provenienţă" se înţelege informaţia din care se pot identifica cel puţin numele autorului / autorilor, titlul operei, anul apariţiei.

b) Plagiatul este definit prin textul legii[1].

> *„...plagiatul – expunerea într-o operă scrisă sau o comunicare orală, inclusiv în format electronic, a unor texte, idei, demonstraţii, date, ipoteze, teorii, rezultate ori metode ştiinţifice extrase din opere scrise, inclusiv în format electronic, ale altor autori, fără a menţiona acest lucru şi fără a face trimitere la operele originale...".*

Tehnic, plagiatul are la bază conceptul de **piesă de creaţie** care[2]:

> *„...este un element de comunicare prezentat în formă scrisă, ca text, imagine sau combinat, care posedă un subiect, o organizare sau o construcţie logică şi de argumentare care presupune nişte premise, un raţionament şi o concluzie. Piesa de creaţie presupune în mod necesar o formă de exprimare specifică unei persoane. Piesa de creaţie se poate asocia cu întreaga operă autentică sau cu o parte a acesteia..."*

cu care se poate face identificarea operei plagiate sau suspicionate de plagiat[3]:

> *„...O operă de creaţie se găseşte în poziţia de operă plagiată sau operă suspicionată de plagiat în raport cu o altă operă considerată autentică dacă:*
>
> *i) Cele două opere tratează acelaşi subiect sau subiecte înrudite.*
> *ii) Opera autentică a fost făcută publică anterior operei suspicionate.*
> *iii) Cele două opere conţin piese de creaţie identificabile comune care posedă, fiecare în parte, un subiect şi o formă de prezentare bine definită.*
> *iv) Pentru piesele de creaţie comune, adică prezente în opera autentică şi în opera suspicionată, nu există o menţionare explicită a provenienţei. Menţionarea provenienţei se face printr-o citare care permite identificarea piesei de creaţie preluate din opera autentică.*
> *v) Simpla menţionare a titlului unei opere autentice într-un capitol de bibliografie sau similar acestuia fără delimitarea întinderii preluării nu este de natură să evite punerea în discuţie a suspiciunii de plagiat.*
> *vi) Piesele de creaţie preluate din opera autentică se utilizează la construcţii realizate prin juxtapunere fără ca acestea să fie tratate de autorul operei suspicionate prin poziţia sa explicită.*
> *vii) În opera suspicionată se identifică un fir sau mai multe fire logice de argumentare şi tratare care leagă aceleaşi premise cu aceleaşi concluzii ca în opera autentică..."*

---

[1] Legea nr. 206/2004 privind buna conduită în cercetarea ştiinţifică, dezvoltarea tehnologică şi inovare, publicată în Monitorul Oficial al României, Partea I, nr. 505 din 4 iunie 2004

[2] ISOC, D. *Ghid de acţiune împotriva plagiatului: bună-conduită, prevenire, combatere*. Cluj-Napoca: Ecou Transilvan, 2012.

[3] ISOC, D. *Preventor de plagiat*. Cluj-Napoca: Ecou Transilvan, 2014.

algorithms

MDPI

Article

Learning Output Reference Model Tracking for Higher-Order Nonlinear Systems with Unknown Dynamics

Mircea-Bogdan Radac * and Timotei Lala

Department of Automation and Applied Informatics, Politehnica University of Timisoara, 2 Bd. V. Parvan, 300223 Timisoara, Romania; timotei.lala@student.upt.ro

* Correspondence: bogdan.ttl@gmail.com; Tel.: +40-256-403240; Fax: +40-256-403214

Received: 1 May 2019; Accepted: 9 June 2019; Published: 12 June 2019

Abstract: Linearly and nonlinearly parameterized approximate dynamic programming approaches used for output reference model (ORM) tracking control are proposed. The ORM tracking problem is of significant interest in practice since, with a linear ORM, the closed-loop control system is indirectly feedback linearized and value iteration (VI) offers the means to achieve ORM tracking without using process dynamics. Ranging from linear to nonlinear parameterizations, a successful approximate VI implementation for continuous state-action spaces depends on several key parameters such as: problem dimension, exploration of the state-action space, the state-transitions dataset size, and suitable selection of the function approximators. We show that using the same transitions dataset and under a general linear parameterization of the Q-function, high performance ORM tracking can be achieved with an approximate VI scheme, on the same performance level as that of a neural-network (NN)-based implementation that is more complex and takes significantly more time to learn. However, the latter proves to be more robust to hyperparameters selection, dataset size, and to exploration strategies, recommending it as the *de facto* practical implementation. The case study is aimed at ORM tracking of a real-world nonlinear two inputs–two outputs aerodynamic process with ten internal states, as a representative high order system.

Keywords: approximate dynamic programming; reinforcement learning; data-driven control; model-free control; reference trajectory tracking; output reference model; multivariable control; aerodynamic rotor system; neural networks; learning systems

1. 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 and not only in the vicinity of an operating point. 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] that can extrapolate optimal behavior to previously unseen tracking scenarios.

P01 On another side, selection of a suitable ORM is not straightforward because of several reasons. The ORM has to be matched with the process bandwidth and with several process nonlinearities such as, e.g., input and output saturations. From classical control theory, dead-time and non-minimum-phase 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 [2], and from classical control [3–7], where it is better known as Adaptive (Approximate, Neuro) Dynamic Programming (ADP). Certain ADP variants can ensure ORM tracking control without knowing the state-space (transition function) dynamics of the controlled process, which is of high importance in the practice of model-free (herein accepted as unknown dynamics) and data-driven control schemes that are able to compensate for poor modeling and process model uncertainty. Thus, ADP relies only on data collected from the process called state transitions. While plenty of mature ADP schemes already exist in the literature, tuning such schemes for a particular problem requires significant experience. Firstly, it must be specified whether ADP deals with continuous (infinite) or discrete (finite) state-action spaces. Then, the intended implementation will decide upon online/offline and/or adaptive/batch processing, the suitable selection of the approximator used for the extended cost function (called the Q-function) and/or for the controller. Afterwards, linear or nonlinear parameterizations are sought. Exploration of the state-action spaces is critical, as well as the hyperparameters of the overall learning scheme such as the number of transition samples, trading off exploration with exploitation, etc. Although successful stories on RL and ADP applied to large state-action spaces are reported mainly with artificial intelligence [8], in control theory, most approaches use low-order processes as representative case studies and mainly in linear quadratic regulator (LQR)-like settings (regulating states to zero). While, in an ADP, the reference input tracking control problem has been tackled before for linear time-invariant (LTI) processes by the name of Linear Quadratic Tracking (LQT) [9,10], the ORM tracking for nonlinear processes was rarely addressed [11].

The iterative model-free approximate Value Iteration (IMF-AVI) proposed in this work belongs to the family of batch-fitted Q-learning schemes [12,13] known as action-dependent heuristic dynamic programming (ADHDP) that are 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) [14–22].

Concerning the exploration issue in ADP for control, a suitable exploration that covers as well as possible the state-action space is not trivially ensured. 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 [23]. The existence of an initial feedback stabilizing controller, not necessarily of a high performance one, can accelerate the transition samples dataset collection under exploration. This allows for offline IMF-AVI based on large datasets, leading to improved convergence speed for high-dimensional processes. 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) [24], Iterative Feedback Tuning [25], Model Free Iterative Learning Control [26–28], Model Free (Adaptive) Control [29,30], with representative applications [31–33]. This work shows a successful example of a model-free output feedback controller used to collect input-to-state transition samples from the process for learning state-feedback ADP-based ORM tracking control. Therefore it fits with the recent data-driven control [34–42] and reinforcement learning [43,44] applications.

The case study deals with the challenging ORM tracking control for a nonlinear real-world two-inputs–two-outputs aerodynamic system (TITOAS) 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 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.

As a main contribution, the paper is focused on a detailed comparison of the advantages and disadvantages of using linear and nonlinear parameterizations for the IMF-AVI scheme, while covering complete implementation details. To the best of authors' knowledge, the ORM tracking context with linear parameterizations was not studied before for high-order real-world processes. Moreover, theoretical analysis shows convergence of the IMF-AVI while accounting for approximation errors and explains for the robust learning convergence of the NN-based IMF-AVI. The results indicate that the nonlinearly parameterized NN-based IMF-AVI implementation should be *de facto* in practice since, although more time-consuming, it automatically manages the basis function selection, it is more robust to dataset size and exploration settings, and generally more well-suited for nonlinear processes with unknown dynamics.

Section 2 is dedicated to the formalization of the ORM tracking control problem, while Section 3 proposes a solution to this problem using an IMF-AVI approach. Section 4 validates the proposed approach on the TITOAS system, with concluding remarks presented in Section 5.

2. Output Model Reference Control for Unknown Dynamics Nonlinear Processes

P02

2.1. The Process

A discrete-time nonlinear unknown open-loop stable state-space deterministic strictly causal process is defined as

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

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

Assumption 1 (A1). *(1) is fully state controllable with measurable states.*

Assumption 2 (A2). *(1) is input-to-state stable on known domain $\Omega_U \times \Omega_X$.*

Assumption 3 (A3). *(1) is minimum-phase (MP).*

A1 and A2 are widely used in data-driven control, cannot be checked analytically for the unknown model (1) but can be inferred from historical and working knowledge with the process. Should such information not be available, the user can attempt process control under restraining safety operating conditions, that are usually dealt with at supervisory level control. Input to state stability (A2) is necessary if open-loop input-state samples collection is intended to be used for state space control design. Assumption A2 can be relaxed if a stabilizing state-space controller is already available and used just for the purpose of input-state data collection. A3 is the least restrictive assumption and it is used in the context of the VRFT design of a feedback controller based on input–output process data. Although solutions exist to deal with nonminimum-phase systems processes, the MP assumption simplifies the VRFT design and the output reference model selection (to be introduced in the following section).

Comment 1. Model (1) accounts for a wide range of processes including fixed time-delay ones. For positive integer nonzero delay d on the control input $\mathbf{u}_{k-d}$, additional states can extend the

initial process model (1) as $\mathbf{x}^u_{k,1} = \mathbf{u}_{k-1}, \mathbf{x}^u_{k,2} = \mathbf{u}_{k-2}, ..., \mathbf{x}^u_{k,d} = \mathbf{u}_{k-d}$ and arrive at a state-space model without delays, in which the additional states are measurable as past input samples. A delay in the original states in (1), i.e., $\mathbf{x}_{k-d}$, are similarly treated.

P03

2.2. Output Reference Model Control Problem Definition

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

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

where $\mathbf{x}^m_k = [x^m_{k,1}, ..., x^m_{k,n}]^\top \in \Omega_{X_m} \subset \mathbb{R}^n_m$ is the ORM state, $\mathbf{r}_k = [r_{k,1}, ..., r_{k,p}]^\top \in \Omega_{R_m} \subset \mathbb{R}^p$ is the reference input signal, $\mathbf{y}^m_k = [y^m_{k,1}, ..., y^m_{k,p}]^\top \in \Omega_{Y_m} \subset \mathbb{R}^p$ is the ORM output, $\mathbf{f}^m : \Omega_{X_m} \times \Omega_{R_m} \mapsto \Omega_{X_m}$, $\mathbf{g}^m : \Omega_{X_m} \mapsto \Omega_{Y_m}$ are *known* nonlinear mappings. Initial conditions are zero unless otherwise stated. Notice that $\mathbf{r}_m, \mathbf{y}_k, \mathbf{y}^m_k$ are size p for square feedback control systems (CSs). If the ORM (2) is LTI, it is always possible to express the ORM as an IO LTI transfer function (t.f.) $\mathbf{M}(z)$ ensuring $\mathbf{y}^m_k = \mathbf{M}(z)\mathbf{r}_k$, where $\mathbf{M}(z)$ is commonly an asymptotically stable unit-gain rational t.f. 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, we consider the reference input $\mathbf{r}_k$ as a set of measurable exogenous signals (possibly interpreted as a disturbance) that evolve according to $\mathbf{r}_{k+1} = \mathbf{h}^m(\mathbf{r}_k)$, with known nonlinear $\mathbf{h}^m : \mathbb{R}^m \mapsto \mathbb{R}^m$, where $\mathbf{r}_k$ is measurable. Herein, $\mathbf{h}^m(.)$ is a generative model for the reference input.

The class of LTI generative models $\mathbf{h}^m(.)$ has been studied before in [9] but it is a rather restrictive one. For example, reference inputs signals modeled as a sequence of steps of constant amplitude cannot be modeled by LTI generative models. A step reference input signal with constant amplitude over time can be modeled as $\mathbf{r}_{k+1} = \mathbf{r}_k$ with some initial condition $\mathbf{r}_0$. On the other hand, a sinusoidal scalar reference input signal r_k can be modeled only through a second order state-space model. To see this, let the Laplace transform of $cos(\omega t)\sigma(t)$ ($\sigma(t)$ is the unit step function) be $H(s) = \ell\{cos(\omega t)\sigma(t)\}$ with the complex Laplace variable s. If $sH(s)$ is considered a t.f. driven by the unit step function with Laplace transform $\ell\{\sigma(t)\} = 1/s$, then the LTI discrete-time state-space associated with $sH(s)$ acting as a generative model for r_k is of the form

$$\begin{aligned} \mathbf{o}_{k+1} &= \mathbf{A}\mathbf{o}_k + \mathbf{B}\sigma_k, \\ r_k &= \mathbf{C}\mathbf{o}_k + \mathbf{D}\sigma_k, \end{aligned} \tag{3}$$

with known $\mathbf{A} \in \mathbb{R}^{2\times2}, \mathbf{B} \in \mathbb{R}^{2\times1}, \mathbf{C} \in \mathbb{R}^{1\times2}, D \in \mathbb{R}, \mathbf{o}_0 = [0,0]^\top$, and $\sigma_k = \{1,1,1,...\}$ is the discrete-time unit step function. The combination of $H(s)$ driven by the Dirac impulse with $\ell\{\delta(t)\}$ could also have been considered as a generative model. Based on the state-space model above, modeling p sinusoidal reference inputs $\mathbf{r}_k \in \Omega_{R_m} \subset \mathbb{R}^p$ requires $2p$ states. Generally speaking, the generative model of the reference input must obey the Markov property.

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:

$$\mathbf{x}^E_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}^m_{k+1} \\ \mathbf{r}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{f}^m(\mathbf{x}^m_k, \mathbf{r}_k) \\ \mathbf{h}^m(\mathbf{r}_k) \end{bmatrix} = \mathrm{E}(\mathbf{x}^E_k, \mathbf{u}_k), \mathbf{x}^E_k \in \Omega_{X^E}, \tag{4}$$

where $\mathbf{x}^E_k$ is called the extended state vector. Note that the extended state-space fulfils the Markov property. The ORM tracking control problem is formulated in an optimal control framework. Let the infinite horizon cost function (c.f.) to be minimized starting with $\mathbf{x}_0$ be [6]

$$J^\infty_{MR}(\mathbf{x}^E_0, \theta) = \sum_{k=0}^{\infty} \gamma^k \|\mathbf{y}^m_k(\mathbf{x}^E_k) - \mathbf{y}_k(\mathbf{x}^E_k, \theta)\|^2_2 = \sum_{k=0}^{\infty} \gamma^k \|e_k(\mathbf{x}^E_k, \theta)\|^2_2. \tag{5}$$

In (5), the discount factor $0 < \gamma \leq 1$ sets the controller's horizon, $\gamma < 1$ is usually used in practice to guarantee learning convergence to optimal control. $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$ is the Euclidean norm of the column vector $\mathbf{x}$. $v_{MR} = \|\mathbf{y}_k^m(\mathbf{x}_k^E) - \mathbf{y}_k(\mathbf{x}_k^E)\|_2^2$ is the stage cost where measurable $\mathbf{y}_k$ depends via unknown $\mathbf{g}$ in (1) on $\mathbf{x}_k$ and v_{MR} penalizes the deviation of $\mathbf{y}_k$ from the ORM's output $\mathbf{y}_k^m$. In ORM tracking, the stage cost does not penalize the control effort with some positive definite function $W(\mathbf{u}_k) > 0$ since the ORM tracking instills an inertia on the CS that indirectly acts as a regularizer on the control effort. Secondly, if the reference inputs $\mathbf{r}_k$ do not set to zero, the ORM's outputs also do not. For most processes, the corresponding constant steady-state control will be non-zero, hence making $J_{MR}^{\infty}(\boldsymbol{\theta})$ infinite when $\gamma = 1$.

Herein, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ parameterizes a nonlinear state-feedback admissible controller [6] defined as $\mathbf{u}_k \triangleq \mathbf{C}(\mathbf{x}_k^E, \boldsymbol{\theta})$, which used in (4) shows that all CS's trajectories depend on $\boldsymbol{\theta}$. Any stabilizing controller sequence (or controller) rendering a finite c.f. is called admissible. A finite J_{MR}^{∞} holds if $\boldsymbol{\epsilon}_k$ is a square-summable sequence, ensured by an asymptotically stabilizing controller if $\gamma = 1$ or by a stabilizing controller if $\gamma < 1$. $J_{MR}^{\infty}(\boldsymbol{\theta})$ in (5) is the value function of using the controller $\mathbf{C}(\boldsymbol{\theta})$. Let the optimal controller $\mathbf{u}_k^* = \mathbf{C}(\mathbf{x}_k^E, \boldsymbol{\theta}^*)$ that minimizes (5) be

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} J_{MR}^{\infty}(\mathbf{x}_0^E, \boldsymbol{\theta}). \tag{6}$$

Tracking a nonlinear ORM can also be used, however, tracking a linear one renders highly desirable indirect feedback linearization of the CS, where a linear CS's behavior generalizes well in wide operating ranges [1]. Then the ORM tracking control problem of this work should make $v_{MR} \approx 0$ when $\mathbf{r}_k$ drives both the CS and the ORM.

Under classical control rules, following *Comment 1*, the process time delay and non-minimum-phase (NMP) character should be accounted for in $\mathbf{M}(z)$. However, the NMP zeroes make $\mathbf{M}(z)$ non-invertible in addition to requiring their identification, thus placing a burden on the subsequent VRFT IO control design [45]. This motivates the MP assumption on the process.

Depending on the learning context, 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 fulfill the Markov property of the extended state-space model. In many practical applications, for the ORM tracking problem, 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 expressed by a generative model of the form $\mathbf{r}_{k+1} = \mathbf{r}_k$. This generative model will be used subsequently in this paper for learning ORM tracking controllers. Obviously, the learnt solution will depend on the proposed reference input generative model, while changing this model requires re-learning.

3. Solution to the ORM Tracking Problem

For unknown extended process dynamics (4), minimization of (5) can be tackled using an iterative model-free approximate Value Iteration (IMF-AVI). A c.f. that extends $J_{MR}^{\infty}(\mathbf{x}_k^E)$ 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) $\mathbf{u}_k = \mathbf{C}(\mathbf{x}_k^E)$ be

$$Q^{\mathbf{C}}(\mathbf{x}_k^E, \mathbf{u}_k) = v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q^{\mathbf{C}}(\mathbf{x}_{k+1}^E, \mathbf{C}(\mathbf{x}_{k+1}^E)). \tag{7}$$

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

$$Q^*(\mathbf{x}_k^E, \mathbf{u}_k) = \min_{\mathbf{C}(.)} \{v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q^{\mathbf{C}}(\mathbf{x}_{k+1}^E, \mathbf{C}(\mathbf{x}_{k+1}^E))\}, \tag{8}$$

where the optimal controller and Q-functions are

$$\mathbf{u}_k^* = \mathbf{C}^*(\mathbf{x}_k^E) = arg\min_{\mathbf{C}} Q^{\mathbf{C}}(\mathbf{x}_k^E, \mathbf{u}_k), Q^*(\mathbf{x}_k^E, \mathbf{u}_k) = \min_{\mathbf{C}(.)} Q^C(\mathbf{x}_k^E, \mathbf{u}_k). \tag{9}$$

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

The optimal Q-function and optimal controller can be found using either Policy Iteration (PolI) 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. Several Q-function parameterizations allow for explicit analytic calculation of the improved controller as the following optimization problem

$$\bar{\mathbf{C}}(\mathbf{x}_k^E, \pi) = arg\min_{\mathbf{C}} Q^{\mathbf{C}}(\mathbf{x}_k^E, \mathbf{u}_k, \pi), \tag{10}$$

by directly minimizing $Q^{\mathbf{C}}(\mathbf{x}_k^E, \mathbf{u}_k, \pi)$ w.r.t. $\mathbf{u}_k$, where the parameterization π has been moved from the controller into the Q-function. (10) is the controller improvement step specific to both the PolI and VI algorithms. In these special cases, it is possible to eliminate the controller approximator and use only one for the Q-function Q. 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 every iteration j

$$\pi_{j+1} = arg\min_{\pi} \sum_{k=1}^{N} (Q(\mathbf{x}_k^E, \mathbf{u}_k, \pi) - v(\mathbf{x}_k^E, \mathbf{u}_k) - \gamma Q(\mathbf{x}_{k+1}^E, \bar{\mathbf{C}}(\mathbf{x}_{k+1}^E, \pi_j), \pi_j))^2, \tag{11}$$

which is a Bellman residual minimization problem where the (usually separate) controller improvement step is now embedded inside the OP (11). More explicitly, for a linear parameterization $Q(\mathbf{x}_k^E, \mathbf{u}_k, \pi) = \mathbf{\Phi}^\top(\mathbf{x}_k^E, \mathbf{u}_k)\pi$ using a set of n_Φ basis functions of the form $\mathbf{\Phi}^\top(\mathbf{x}_k^E, \mathbf{u}_k) = [\Phi_1(\mathbf{x}_k^E, \mathbf{u}_k), ..., \Phi_{n_\Phi}(\mathbf{x}_k^E, \mathbf{u}_k)]$, the least squares solution to (11) is equivalent to solving the following over-determined linear system of equations w.r.t. π_{j+1} in the least-squares sense:

$$\begin{bmatrix} \mathbf{\Phi}^\top(\mathbf{x}_1^E, \mathbf{u}_1) \\ ... \\ \mathbf{\Phi}^\top(\mathbf{x}_N^E, \mathbf{u}_N) \end{bmatrix} \pi_{j+1} = \begin{bmatrix} v(\mathbf{x}_1^E, \mathbf{u}_1) + \gamma \mathbf{\Phi}^\top(\mathbf{x}_2^E, \bar{\mathbf{C}}(\mathbf{x}_2^E, \pi_j))\pi_j \\ ... \\ v(\mathbf{x}_N^E, \mathbf{u}_N) + \gamma \mathbf{\Phi}^\top(\mathbf{x}_{N+1}^E, \bar{\mathbf{C}}(\mathbf{x}_{N+1}^E, \pi_j))\pi_j \end{bmatrix}. \tag{12}$$

Concluding, starting with an initial parameterization π_0 , the IMF-AVI approach with linearly parameterized Q-function that allows explicit controller improvement calculation as in (10), embeds both VI steps into solving (12). Linearly parameterized IMF-AVI (LP-IMF-AVI) will be validated in the case study and compared to nonlinearly parameterized IMF-AVI (NP-IMF-AVI). Convergence of the generally formulated IMF-AVI is next analysed under approximation errors.

IMF-AVI Convergence Analysis with Approximation Errors for ORM Tracking

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

Algorithm 1 VI-based Q-learning.

S1: Initialize controller $\mathbf{C}_0$ and the Q-function value to $Q_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0$, initialize iteration index $j = 1$
S2: Use one step backup equation for the Q-function as in (13)
S3: Improve the controller using the Equation (14)
S4: Set $j = j + 1$ and repeat steps S2, S3, until convergence

To be detailed as follows:

S1. Select an initial (not necessarily admissible) controller $\mathbf{C}_0$ and an initialization value $Q_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0$ of the Q-function. Initialize iteration $j = 1$.

S2. Use one step backup equation for the Q-function

$$\begin{aligned} Q_j(\mathbf{x}_k^E, \mathbf{u}_k) &= v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{C}_{j-1}(\mathbf{x}_{k+1}^E)) \\ &= \min_{\mathbf{u}} \{v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} \end{aligned} \tag{13}$$

S3. Improve the controller using the equation

$$\mathbf{C}_j(\mathbf{x}_k^E) = arg\min_{\mathbf{u}} Q_j(\mathbf{x}_k^E, \mathbf{u}). \tag{14}$$

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-update for extended c.f. ξ_j as [46]*

$$\xi_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k) = v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \xi_j(\mathbf{x}_{k+1}^E, \kappa_j(\mathbf{x}_{k+1}^E)). \tag{15}$$

If $Q_0(\mathbf{x}_k^E, \mathbf{u}_k) = \xi_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0$, then $Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq \xi_j(\mathbf{x}_k^E, \mathbf{u}_k)$.

Proof. It is valid that

$$\begin{aligned} Q_1(\mathbf{x}_k^E, \mathbf{u}_k) &= v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \overbrace{Q_0(\mathbf{x}_{k+1}^E, \mathbf{C}_0(\mathbf{x}_{k+1}^E))}^{0} = \\ &= v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \overbrace{\xi_0(\mathbf{x}_{k+1}^E, \kappa_0(\mathbf{x}_{k+1}^E))}^{0} = \xi_1(\mathbf{x}_k^E, \mathbf{u}_k). \end{aligned} \tag{16}$$

Meaning that $Q_1(\mathbf{x}_k^E, \mathbf{u}_k) \leq \xi_1(\mathbf{x}_k^E, \mathbf{u}_k)$. Assume by induction that $Q_{j-1}(\mathbf{x}_k^E, \mathbf{u}_k) \leq \xi_{j-1}(\mathbf{x}_k^E, \mathbf{u}_k)$. Then

$$\begin{aligned} Q_j(\mathbf{x}_k^E, \mathbf{u}_k) &= v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{C}_{j-1}(\mathbf{x}_{k+1}^E)) \leq \\ &\leq v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q_{j-1}(\mathbf{x}_{k+1}^E, \kappa_{j-1}(\mathbf{x}_{k+1}^E)) \leq \\ &\leq v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \xi_{j-1}(\mathbf{x}_{k+1}^E, \kappa_{j-1}(\mathbf{x}_{k+1}^E)) = \xi_j(\mathbf{x}_k^E, \mathbf{u}_k), \end{aligned} \tag{17}$$

which completes the proof. Here, it was used that $\mathbf{C}_{j-1}(\mathbf{x}_k^E)$ is the optimal controller for $Q_{j-1}(\mathbf{x}_k^E, \mathbf{u}_k)$ per (14), then, for any other controller $\mathbf{C}(\mathbf{x}_k^E)$ (in particular it can also be $\kappa_{j-1}(\mathbf{x}_k^E)$) it follows that

$$Q_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{C}_{j-1}(\mathbf{x}_{k+1}^E)) \leq Q_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{C}(\mathbf{x}_{k+1}^E)). \tag{18}$$

□

P08

Lemma 2. *For the sequence $\{Q_j\}$ from (13), under controllability assumption A1, it is valid that:*
(1) $0 \leq Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq B(\mathbf{x}_k^E, \mathbf{u}_k)$ with $B(\mathbf{x}_k^E, \mathbf{u}_k)$ an upper bound.
(2) If there exists a solution $Q^*(\mathbf{x}_k^E, \mathbf{u}_k)$ to (8), then $0 \leq Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq Q^*(\mathbf{x}_k^E, \mathbf{u}_k) \leq B(\mathbf{x}_k^E, \mathbf{u}_k)$.

Proof. For any fixed admissible controller $\eta(\mathbf{x}_k^E)$, $Q^\eta(\mathbf{x}_k^E, \mathbf{u}_k) = v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q^\eta(\mathbf{x}_{k+1}^E, \eta(\mathbf{x}_{k+1}^E))$ is the Bellman equation. Update (13) renders

2 of Lemma 2 states that $Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq Q^*(\mathbf{x}_k^E, \mathbf{u}_k)$ implying $Q_\infty(\mathbf{x}_k^E, \mathbf{u}_k) \leq Q^*(\mathbf{x}_k^E, \mathbf{u}_k)$. Then from $Q_\infty(\mathbf{x}_k^E, \mathbf{u}_k) \leq Q^*(\mathbf{x}_k^E, \mathbf{u}_k) \leq Q_\infty(\mathbf{x}_k^E, \mathbf{u}_k)$ it must hold true that $Q_\infty(\mathbf{x}_k^E, \mathbf{u}_k) = Q^*(\mathbf{x}_k^E, \mathbf{u}_k)$ and $\mathbf{C}_\infty(\mathbf{x}_k^E) = \mathbf{C}^*(\mathbf{x}_k^E)$ which proves the second part of Theorem 1. □

P10

Comment 2. (13) is practically solved in the sense of the OP (11) (either as a linear or nonlinear regression) using a batch (dataset) of transition samples collected from the process using any controller, that is in *off-policy* mode. While the controller improvement step (14) can be solved either as a regression or explicitly analytically when the expression of $Q_j(\mathbf{x}_k^E, \mathbf{u}_k)$ allows it. Moreover, (13) and (14) can be solved batch-wise in either online or offline mode. When the batch of transition samples is updated with one sample at a time, the VI-scheme becomes adaptive.

P11

Comment 3. Theorem 1 proves the VI-based learning convergence of the sequence of Q-functions $\lim_{j\to\infty} Q_j(\mathbf{x}_k^E, \mathbf{u}_k) = Q^*(\mathbf{x}_k^E, \mathbf{u}_k)$ 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 generic convergence proof of the learning scheme under approximation errors is next shown, accounting for general Q-function parameterizations [47].

Let the IMF-AVI Algorithm 2 consist of the steps.

Algorithm 2 IMF-AVI.

S1: Initialize controller $\bar{\mathbf{C}}_0$ and Q-function value $\bar{Q}_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$. Initialize iteration $j = 1$
S2: Update the approximate Q-function using Equation (24)
S3: Improve the approximate controller using Equation (25)
S4: Set $j = j + 1$ and repeat steps S2, S3, until convergence

P12

To be detailed as follows:

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

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

$$\begin{aligned}\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k) &= v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \bar{Q}_{j-1}(\mathbf{x}_{k+1}^E, \bar{\mathbf{C}}_{j-1}(\mathbf{x}_{k+1}^E)) + \delta_j(\mathbf{x}_k^E, \mathbf{u}_k) \\ &= \min_{\mathbf{u}}\{v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \bar{Q}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} + \delta_j\end{aligned} \tag{24}$$

S3. Improve the approximate controller using

$$\bar{\mathbf{C}}_j(\mathbf{x}_k^E) = arg\min_{\mathbf{u}} \bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}) \tag{25}$$

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

Comment 4. In Algorithm 2, the sequences $\{\bar{\mathbf{C}}_j(\mathbf{x}_k^E)\}$ and $\{\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u})\}$ are approximations of the true sequences $\{\mathbf{C}_j(\mathbf{x}_k^E)\}$ and $\{Q_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$. Since the true Q-function and controller parameterizations are not generally known, (24) must be solved in the sense of the OP (11) with respect to the unknown $\bar{Q}_j$, 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 (24), (25) coincide with (13), (14), respectively. Next, let the following assumption hold.

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

$$\begin{aligned}\min_{\mathbf{u}}\{\underline{\psi} v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \bar{Q}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\} \leq \bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq \\ \min_{\mathbf{u}}\{\overline{\psi} v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma \bar{Q}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\}.\end{aligned} \tag{26}$$

Comment 5. Inequalities from (26) account for nonzero positive or negative residuals δ_j , i.e., for the approximation errors in the Q-function, since $\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k)$ can over- or under-estimate $\min_{\mathbf{u}}\{v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma\bar{Q}_{j-1}(\mathbf{x}_{k+1}^E, \mathbf{u})\}$ in (24). $\underline{\psi}, \overline{\psi}$ can span large intervals ($\underline{\psi}$ close to 0 and $\overline{\psi}$ very large). The 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 (24).

Theorem 2. *Let the sequences $\{\bar{C}_j(\mathbf{x}_k^E)\}$ and $\{\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ evolve as in (24), (25), the sequences $\{C_j(\mathbf{x}_k^E)\}$ and $\{Q_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ evolve as in (13), (14). Initialize $\bar{Q}_0(\mathbf{x}_k^E, \mathbf{u}_k) = Q_0(\mathbf{x}_k^E, \mathbf{u}_k) = 0, \forall(\mathbf{x}_k^E, \mathbf{u}_k)$ and let A3 hold. Then*

$$\underline{\psi}Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq \bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq \overline{\psi}Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \tag{27}$$

Proof. First, the development proceeds by induction for the left inequality. For $j = 0$ it is clear that $\underline{\psi}Q_0(\mathbf{x}_k^E, \mathbf{u}_k) \leq \bar{Q}_0(\mathbf{x}_k^E, \mathbf{u}_k)$. For $j = 1$, (13) produces $Q_1(\mathbf{x}_k^E, \mathbf{u}_k) = v(\mathbf{x}_k^E, \mathbf{u}_k)$ and left-hand side of (26) reads $\min_{\mathbf{u}}\{\underline{\psi}v(\mathbf{x}_k^E, \mathbf{u}_k) + 0\} \leq \bar{Q}_1(\mathbf{x}_k^E, \mathbf{u}_k)$. Then $\underline{\psi}Q_1(\mathbf{x}_k^E, \mathbf{u}_k) \leq \bar{Q}_1(\mathbf{x}_k^E, \mathbf{u}_k)$. Next assume that

$$\underline{\psi}Q_j(\mathbf{x}_k^E, \mathbf{u}_k) \leq \bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k) \tag{28}$$

holds at iteration j. Based on (28) used in (26), it is valid that

$$\begin{aligned} &\min_{\mathbf{u}}\{\underline{\psi}v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma\underline{\psi}Q_j(\mathbf{x}_{k+1}^E, \mathbf{u})\} \leq \\ &\min_{\mathbf{u}}\{\underline{\psi}v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma\bar{Q}_j(\mathbf{x}_{k+1}^E, \mathbf{u})\} \leq \bar{Q}_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k). \end{aligned} \tag{29}$$

Notice from (29) that

$$\begin{aligned} \min_{u}\{\underline{\psi}v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma\underline{\psi}Q_j(\mathbf{x}_{k+1}^E, \mathbf{u})\} &= \underline{\psi}\min_{u}\{v(\mathbf{x}_k^E, \mathbf{u}_k) + \gamma Q_j(\mathbf{x}_{k+1}^E, \mathbf{u})\} \\ &\overset{(13)}{=} \underline{\psi}Q_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k) \end{aligned} \tag{30}$$

From (29), (30) it follows that $\underline{\psi}Q_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k) \leq \bar{Q}_{j+1}(\mathbf{x}_k^E, \mathbf{u}_k)$ proving the left side of (27) by induction. The right side of (27) is shown similarly, proving Theorem 2. □

P13

Comment 6. Theorem 2 shows that the trajectory of $\{\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ closely follows that of $\{Q_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ in a bandwidth set by $\underline{\psi}, \overline{\psi}$. It does not ensure that $\{\bar{Q}_j(\mathbf{x}_k^E, \mathbf{u}_k)\}$ converges to a steady-state value, but in the worst case, it oscillates around $Q^*(\mathbf{x}_k^E, \mathbf{u}_k) = lim_{j\to\infty}Q_j(\mathbf{x}_k^E, \mathbf{u}_k)$ in a band that can be made arbitrarily small by using powerful approximators. By minimizing over $\mathbf{u}_k$ both sides of (27), similar conclusions result for the controller sequence $\{\bar{C}_j(\mathbf{x}_k^E)\}$ that closely follows $\{C_j(\mathbf{x}_k^E)\}$.

In the following Section, the IMF-AVI is validated on two illustrative examples. The provided theoretical analysis supports and explains the robust learning performance of the nonlinearly parameterized IMF-AVI with respect to the linearly parameterized one.

4. Validation Case Studies

4.1. ORM Tracking for a Linear Process

A first introductory simple example of IMF-AVI for the ORM tracking of a first-order process motivates the more complex validation for the TITOAS process and offers insight into how the IMF-AVI solution scales up with the higher-order processes.

Let a scalar discrete-time process discretized at $T_s = 0.1$s be $x_{k+1} = 0.8187x_k + 0.1813u_k$. The continuous-time ORM $M(s) = 1/(s+1)$ ZOH discretized at the same T_s leads to the extended process equivalent to (4), (output equations also given):

with $\pi \in \mathbb{R}^{10}$. The controller improvement step equivalent to explicitly minimizing the Q-function w.r.t. the control input u_k is $u_k^* = C^*(\mathbf{x}_k^E, \pi_j) = -\frac{1}{2\pi_{4,j}}[\pi_{7,j}, \pi_{9,j}, \pi_{10,j}]\mathbf{x}_k^E$. This improved linear-in-the-state controller is embedded in the linear system of equations (12) that is solved for every iteration of IMF-AVI. Each iteration produces a new π_{j+1} that is tested on a test scenario where the uniformly random reference inputs have amplitude $r_k \in [-1; 1]$ and switch every 10 s. The ORM tracking performance is then measured by the Euclidean vector norm $\|y_k^m - y_k\|_2$ while $\|\pi_{j+1} - \pi_j\|_2$ serves as a stopping condition when it drops below a prescribed threshold. The practically observed convergence process is shown in Figure 2 over the first 400 iterations, with $\|\pi_{j+1} - \pi_j\|_2$ still decreasing after 1000 iterations. While $\|y_k^m - y_k\|_2$ is very small right from the first iterations, making the process output practically overlap with the ORM's output.

Comment 7. For LTI processes with an LQR-like c.f., an LTI ORM and an LTI generative reference input model, linear parameterizations of the extended Q-function of the form $Q(\mathbf{x}_k^E, \mathbf{u}_k) = \Phi^\top(\mathbf{x}_k^E, \mathbf{u}_k)\pi$ is the well-known [9] form $Q(\mathbf{x}_k^E, \mathbf{u}_k) = [(\mathbf{x}_k^E)^\top, (\mathbf{u}_k)^\top]\mathbf{P}[(\mathbf{x}_k^E)^\top, (\mathbf{u}_k)^\top]^\top$ of the quadratic Q-function, with parameter $\pi = vec(\mathbf{P})$ being the vectorized form of the symmetric positive-definite matrix $\mathbf{P}$ and the basis function vector $\Phi^\top(\mathbf{x}_k^E, \mathbf{u}_k)$ is obtained by the nonrepeatable terms of the Kronecker product of all the Q-function input arguments.

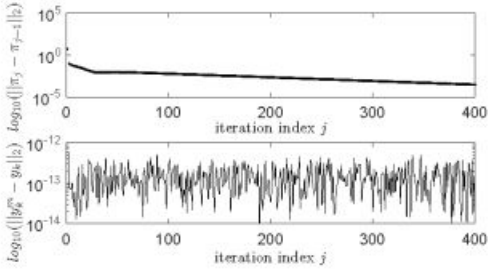

Figure 2. Convergence results of the linearly paramaterized iterative model-free approximate Value Iteration (LP-IMF-AVI) for the linear process example.

P14

4.2. IMF-AVI on the Nonlinear TITOAS Aerodynamic System

The ORM tracking problem on the more challenging TITOAS angular position control [48] (Figure 3) is aimed next. The azimuth (horizontal) motion behaves as an integrator while the pitch (vertical) positioning is affected differently by the gravity for the up and down motions. Coupling between the two channels is present. A simplified deterministic continuous-time state-space model of this process is given as two coupled state-space sub-systems:

$$\begin{cases} \dot{\omega}_h = (sat(U_h) - M_h(\omega_h))/2.5 \cdot 10^{-5}, \\ \dot{K}_h = 0.216F_h(\omega_h)cos(\alpha_v) - 0.058\Omega_h + 0.0178sat(U_v)cos(\alpha_v), \\ \Omega_h = K_k/(0.0238cos^2(\alpha_v) + 3 \cdot 10^{-3}), \\ \dot{\alpha}_h = \Omega_h, \end{cases}$$

$$\begin{cases} \dot{\omega}_v = (sat(U_v) - M_v(\omega_v))/1.63 \cdot 10^{-4}, \\ \dot{\Omega}_v = \frac{1}{0.03}\begin{pmatrix} 0.2F_v(\omega_v) - 0.0127\Omega_v - 0.0935sin\alpha_v + \\ -9.28 \cdot 10^{-6}\Omega_v|\omega_v| + 4.17 \cdot 10^{03}sat(U_h) - 0.05cos\alpha_v + \\ -0.021\Omega_h^2 sin\alpha_v cos\alpha_v - 0.093sin\alpha_v + 0.05 \end{pmatrix} \\ \dot{\alpha_v} = \Omega_v, \end{cases} \quad (33)$$

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 output, $\alpha_v(rad) = y_2 \in [-\pi/2, \pi/2]$ is the pitch angle output, other states being described in [11,48]. The nonlinear static characteristics obtained by polynomial fitting from experimental data are for $\omega_v, \omega_h \in (-4000; 4000)$:

$$\begin{array}{rcl} M_v(\omega_v) & = & 9.05 \times 10^{-12}\omega_v^3 + 2.76 \times 10^{-10}\omega_v^2 + 1.25 \times 10^{-4}\omega_v + 1.66 \times 10^{-4}, \\ F_v(\omega_v) & = & -1.8 \times 10^{-18}\omega_v^5 - 7.8 \times 10^{-16}\omega_v^4 + 4.1 \times 10^{-11}\omega_v^3 + 2.7 \times 10^{-8}\omega_v^2 \\ & & +3.5 \times 10^{-5}\omega_v - 0.014, \\ M_h(\omega_h) & = & 5.95 \times 10^{-13}\omega_h^3 - 5.05 \times 10^{-10}\omega_h^2 + 1.02 \times 10^{-4}\omega_h + 1.61 \times 10^{-3}, \\ F_h(\omega_h) & = & -2.56 \times 0^{-20}\omega_h^5 + 4.09 \times 10^{-17}\omega_h^4 + 3.16 \times 10^{-12}\omega_h^3 - 7.34 \times 10^{-9}\omega_h^2 \\ & & +2.12 \times 10^{-5}\omega_h + 9.13 \times 10^{-3}. \end{array} \quad (34)$$

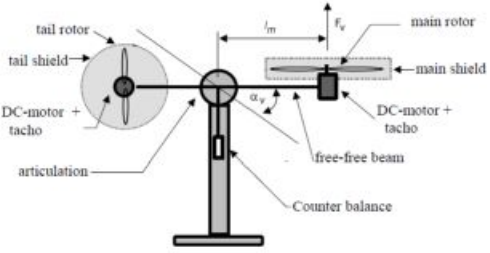

Figure 3. The two-inputs–two-outputs aerodynamic system (TITOAS) experimental setup.

A zero-order hold on the inputs and a sampler on the outputs of (33) lead to an equivalent MP discrete-time model of sampling time $T_s = 0.1s$ and of relative degree 1 (one), suitable for input-state data collection

$$p: \begin{cases} \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) = [\alpha_{h,k}, \alpha_{v,k}]^\top, \end{cases} \quad (35)$$

where $\mathbf{x}_k = [\omega_{h,k}, \Omega_{h,k}, \alpha_{h,k}, \omega_{v,k}, \Omega_{v,k}, \alpha_{v,k}]^\top \in \mathbb{R}^6$ and $\mathbf{u}_k = [u_{k,1}, u_{k,2}]^\top \in \mathbb{R}^2$. The process' dynamics will not be used for learning the control in the following.

P15

4.3. Initial Controller with Model-Free VRFT

An initial model-free multivariable IO controller is first found using model-free VRFT, as described in [11,23,31]. The ORM is $\mathbf{M}(z) = diag(M_1(z), M_2(z))$ 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 $T_s = 0.1$ s. The VRFT prefilter is chosen as $\mathbf{L}(z) = \mathbf{M}(z)$. A pseudo-random binary signal (PRBS) 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 simultaneously, as shown in Figure 4. The IO data $\{\bar{\mathbf{u}}_k, \mathbf{y}_k\}$ is collected with low-amplitude zero-mean inputs $u_{k,1}, u_{k,2}$, in order to maintain the process linearity around the mechanical equilibrium, such that to fit the linear VRFT design framework.

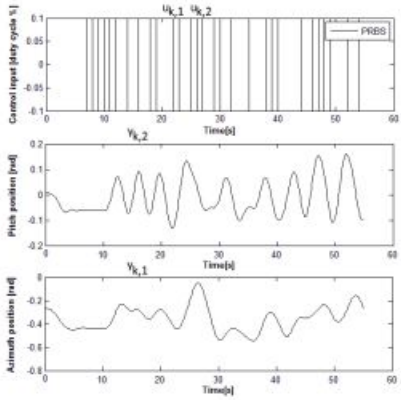

Figure 4. Open-loop input–output (IO) data from the two-inputs–two-outputs aerodynamic system (TITOAS) for Virtual Reference Feedback Tuning (VRFT) controller tuning.

An un-decoupling linear output feedback error diagonal controller with the parameters computed by the VRFT approach is

$$\begin{aligned} &\mathbf{C}(z,\theta) = \begin{bmatrix} P_{11}(z)/(1-z^{-1}) & 0 \\ 0 & P_{22}(z)/(1-z^{-1}) \end{bmatrix}, \\ &P_{11}(z) = 2.9341 - 5.8689z^{-1} + 3.9303z^{-2} - 0.9173z^{-3} - 0.0777z^{-4}, \\ &P_{22}(z) = 0.6228 - 1.1540z^{-1} + 0.5467z^{-2}, \end{aligned} \tag{36}$$

where the parameter vector θ groups all the coefficients of $P_{11}(z), P_{22}(z)$. Controller (36) is obtained for θ as the least squares minimizer of $J_{VR}(\theta) = \sum_{k=1}^{N} \|\bar{\mathbf{u}}_k^L - \mathbf{C}(z,\theta)\bar{\mathbf{e}}_k^L\|_2^2$ where $\bar{\mathbf{u}}_k^L = \mathrm{L}(z)\bar{\mathbf{u}}_k = \mathrm{L}(z)[\bar{u}_{k,1}, \bar{u}_{k,2}]^\top, \bar{\mathbf{e}}_k^L = \mathrm{L}(z)\bar{\mathbf{e}}_k = \mathrm{L}(z)[\bar{e}_{k,1}, \bar{e}_{k,2}]^\top, [\bar{e}_{k,1}, \bar{e}_{k,2}]^\top = (\mathbf{M}^{-1}(z) - \mathrm{I}_2)[\bar{y}_{k,1}, \bar{y}_{k,2}]^\top$. Here, $J_{VR}(\theta)$ is an approximation of the c.f. J_{MR}^{∞} from (5) obtained for $\gamma = 1$. The controller (36) will then close the feedback control loop as in $\mathbf{u}_k = \mathbf{C}(z,\theta)(\mathbf{r}_k - \mathbf{y}_k)$.

Notice that, by formulation, the VRFT controller tuning aims to minimize the undiscounted ($\gamma = 1$) J_{MR}^{∞} from (5), but via the output feedback controller (36) that processes the feedback control error $\mathbf{e}_k = \mathbf{r}_k - \mathbf{y}_k$. The same goal to minimize (5) is pursued by the subsequent IMF-AVI design of a state-feedback controller tuning for the extended process. Nonlinear (in particular, linear) state-feedback controllers can also be found by VRFT as shown in [23,31], to serve as initializations for the IMF-AVI, or possibly, even for PoIt-like algorithms. However, should this not be necessary, IO feedback controllers are much more data-efficient, requiring significantly less IO data to obtain stabilizing controllers.

P16

4.4. Input–State–Output Data Collection

ORM tracking is intended by making the closed loop CS match the same ORM $\mathbf{M}(z) = diag(M_1(z), M_2(z))$. With the linear controller (36) 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 $[-1;1]$, then sent to the process. The reference inputs $r_{k,1}, r_{k,2}$ drive the ORM:

$$\begin{cases} x^m_{k+1,1} = 0.9672x^m_{k,1} + 0.03278r_{k,1}, \\ x^m_{k+1,2} = 0.9672x^m_{k,2} + 0.03278r_{k,2}, \\ \mathbf{y}^m_k = [y^m_{k,1}, y^m_{k,2}]^\top = [x^m_{k,1}, x^m_{k,2}]^\top. \end{cases} \quad (37)$$

Then the states of the ORM (also outputs of the ORM) are also collected along with the states and control inputs of the process, to build the process extended state (4). Let the extended state be:

$$\mathbf{x}^E_k = [\underbrace{x^m_{k,1}, x^m_{k,2}}_{(\mathbf{x}^m_k)^\top}, \underbrace{r_{k,1}, r_{k,2}}_{\mathbf{r}^\top_k}, (\mathbf{x}_k)^\top]^\top. \quad (38)$$

Essentially, the collected $\mathbf{x}^E_k$ and $\mathbf{u}_k$ builds the transitions dataset $D = \{(\mathbf{x}^E_1, \mathbf{u}_1, \mathbf{x}^E_2), ..., (\mathbf{x}^E_{70000}, \mathbf{u}_{70000}, \mathbf{x}^E_{70001})\}$ for $N = 70,000$, used for the IMF-AVI implementation. After collection, an important processing step is the data normalization. Some process states are scaled in order to ensure that all states are inside $[-1;1]$. The scaled process state is $\tilde{\mathbf{x}}_k = [\omega_{h,k}/7200, 25 \cdot \Omega_{h,k}, \alpha_{h,k}, \omega_{v,k}/3500, 40 \cdot \Omega_{v,k}, \alpha_{v,k}]^\top \in \mathbb{R}^6$ and $\mathbf{u}_k = [u_{k,1}, u_{k,2}]^\top \in \mathbb{R}^2$. Other variables such as the reference inputs, the ORM states and the saturated process inputs already have values inside $[-1;1]$. The normalized state is eventually used for state feedback. Collected transition samples are shown in Figure 5 only for the process inputs and outputs, ORM's outputs and reference inputs, for the first 400 s (4000 samples) out of 7000 s.

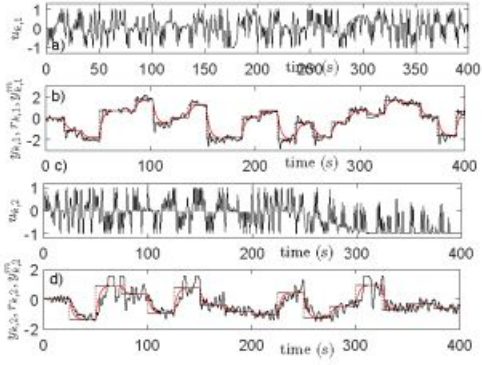

Figure 5. IO data collection with the linear controller [36]: (**a**) $u_{k,1}$; (**b**) $y_{k,1}$ (black), $y^m_{k,1}$ (red), $r_{k,1}$ (black dotted); (**c**) $u_{k,2}$; (**d**) $y_{k,2}$ (black), $y^m_{k,2}$ (red), $r_{k,1}$ (black dotted).

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).

4.5. Learning State-Feedback Controllers with Linearly Parameterized IMF-AVI

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

$$\begin{aligned}\boldsymbol{\Phi}^{\top}(\mathbf{x}_k^E, \mathbf{u}_k) =& [(x_{k,1}^m)^2, (x_{k,2}^m)^2, r_{k,1}^2, ..., x_{k,6}^2, u_{k,1}^2, u_{k,2}^2, x_{k,1}^m x_{k,2}^m, x_{k,1}^m r_{k,1}, ..., \\ & x_{k,1}^m u_{k,2}, x_{k,2}^m r_{k,1}, ..., u_{k,1} u_{k,2}] \in \mathbb{R}^{78}.\end{aligned} \tag{39}$$

This basis functions selection is inspired by the shape of the quadratic Q-function resulting from LTI processes with LQR-like penalties (see *Comment* 7). It is expected to be a sensible choice since the TITOAS process is a nonlinear one, therefore the quadratic Q-function may under-parameterize the true Q-function. Nevertheless, its computational advantage incentives the testing of such a solution.

P18

Notice that the controller improvement step at each iteration of the LP-IMF-AVI is based on explicit minimization of the Q-function. Solving the linear system of equations resulting after setting the derivative of $Q(\mathbf{x}_k^E, \mathbf{u}_k)$ w.r.t. $\mathbf{u}_k$ equal to zero, it is obtained that

$$\begin{aligned}\mathbf{u}_k^* &= \begin{bmatrix} u_{k,1}^* \\ u_{k,2}^* \end{bmatrix} = \bar{\mathbf{C}}^*(\mathbf{x}_k^E, \pi_j) = \begin{bmatrix} 2\pi_{j,11} & \pi_{j,78} \\ \pi_{j,78} & 2\pi_{j,12} \end{bmatrix}^{-1} \begin{bmatrix} F_1(\mathbf{x}_k^E) \\ F_2(\mathbf{x}_k^E) \end{bmatrix}, \\ F_1(\mathbf{x}_k^E) &= \pi_{j,22} x_{k,1}^m + \pi_{j,32} x_{k,2}^m + \pi_{j,41} r_{k,1} + \pi_{j,49} r_{k,2} + \pi_{j,56} x_{k,1} + \\ & \pi_{j,62} x_{k,2} + \pi_{j,67} x_{k,3} + \pi_{j,71} x_{k,4} + \pi_{j,74} x_{k,5} + \pi_{j,76} x_{k,6} = \boldsymbol{\pi}_{j,1}^{\top} \mathbf{x}_k^E, \\ F_2(\mathbf{x}_k^E) &= \pi_{j,23} x_{k,1}^m + \pi_{j,33} x_{k,2}^m + \pi_{j,42} r_{k,1} + \pi_{j,50} r_{k,2} + \pi_{j,57} x_{k,1} + \\ & \pi_{j,63} x_{k,2} + \pi_{j,68} x_{k,3} + \pi_{j,72} x_{k,4} + \pi_{j,75} x_{k,5} + \pi_{j,77} x_{k,6} = \boldsymbol{\pi}_{j,2}^{\top} \mathbf{x}_k^E. \end{aligned} \tag{40}$$

The improved controller is embedded in the system (12) of 70,000 linear equations with 78 unknowns corresponding to the parameters of $\boldsymbol{\pi}_{j+1} \in \mathbb{R}^{78}$. This linear system (12) is solved in least squares sense, with each of the 50 iterations of the LP-IMF-AVI. The practical convergence results are shown in Figure 6 for $\|\boldsymbol{\pi}_{j+1} - \boldsymbol{\pi}_j\|_2$ and for the ORM tracking performance in terms of a normalized c.f. $J_{test} = 1/N(\|y_{k,1} - y_{k,1}^m\|_2 + \|y_{k,2} - y_{k,2}^m\|_2)$ measured for samples over 200 s in the test scenario displayed in Figure 7. The test scenario consists 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 $y_{k,2}$, respectively), to illustrate the existing coupling behavior between the two control channels and the extent to which the learned controller manages to achieve the decoupled behavior requested but the diagonal ORM.

P19

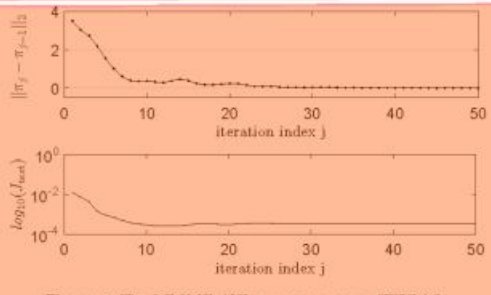

Figure 6. The LP-IMF-AVI convergence on TITOAS.

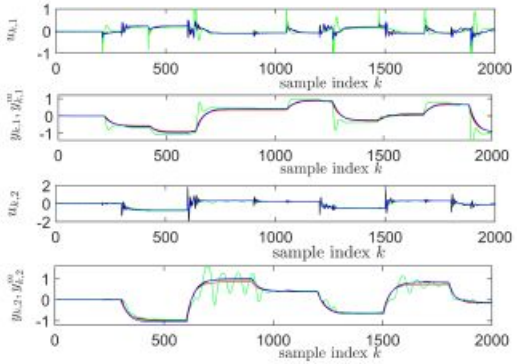

Figure 7. The IMF-AVI convergence on TITOAS: $y^m_{k,1}, y^m_{k,2}$, (red); $u_{k,1}, u_{k,2}, y_{k,1}, y_{k,2}$ for LP-IMF-AVI (black), for NP-IMF-AVI with NNs (blue), for the initial VRFT controller used for transitions collection (green).

The best LP-IMF-AVI controller found over the 50 iterations results in $J_{best} = 0.0017$ (tracking results in black lines in Figure 7), which is more than 6 times lower than the tracking performance of the VRFT controller used for transition samples collection, for which $J_{best} = 0.0103$ (tracking results in green lines in Figure 7). The convergence of the LP-IMF-AVI parameters is depicted in Figure 8.

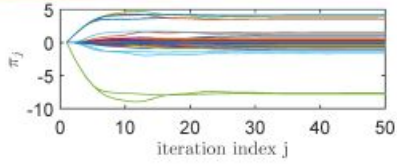

Figure 8. The LP-IMF-AVI parameters convergence.

P20

4.6. Learning State-Feedback Controllers with Nonlinearly Parameterized IMF-AVI Using NNs

The previous LP-IMF-AVI for ORM tracking control learning scheme is next challenged by a NP-IMF-AVI implemented with NNs. In this case, two NNs are used to approximate the Q-function and the controller (the latter is sometimes avoidable, see the comments later on in this sub-section). The procedure follows the NP-IMF-AVI implementation described in [23,49]. The same dataset of transition samples is used as was previously used for the LP-IMF-AVI. Notice that the NN-based implementation is widely used in the reinforcement learning-based approach of ADP and is generally more scalable to problems of high dimension.

The controller NN (C-NN) estimate is a 10–3–2 (10 inputs because $\mathbf{x}^E_k \in \mathbb{R}^{10}$, 3 neurons in the hidden layer, and 2 outputs corresponding to $u_{k,1}, u_{k,2}$) with *tanh* activation function in the hidden layer and linear output activation. The Q-function NN (Q-NN) estimate 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 a maximum of 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 (MSE) evaluated on the validation data. MSE is herein, for all networks, the default performance function used in training.

The NP-IMF-AVI proposed herein consists of two steps for each iteration j. The first one calculates the targets for the NN $Q(\mathbf{x}^E_k, \mathbf{u}_k, \pi_j)$ (having inputs $[(\mathbf{x}^E_k)^\top, (\mathbf{u}_k)^\top]^\top$ and current iteration weights π_j)

$\bar{\mathbf{x}}_k^E = \mathbf{S}\mathbf{x}_k^E$ resulting in the extended state-space model $\bar{\mathbf{x}}_{k+1}^E = \mathbf{S} \cdot \mathrm{E}(\mathbf{S}^{-1}\bar{\mathbf{x}}_k^E, \mathbf{u}_k)$ that still preserves the MDP property.

5. Conclusions

P21

This paper proves a functional design for an IMF-AVI ADP learning scheme dedicated to the challenging problem of ORM tracking control for a high-order real-world complex nonlinear process with unknown dynamics. The investigation revolves around a comparative analysis of a linear vs. a nonlinear parameterization of the IMF-AVI approach. Learning high performance state-feedback control under the model-free mechanism offered by IMF-AVI builds upon the input–states–outputs transition samples collection step that uses an initial exploratory linear output feedback controller that is also designed in a model-free setup using VRFT. From the practitioners' viewpoint, the NN-based implementation of IMF-AVI is more appealing since it easily scales up with problem dimension and automatically manages the basis functions selection for the function approximators.

Future work attempts to validate the proposed design approach to more complex high-order nonlinear processes of practical importance.

Author Contributions: conceptualization, M.-B.R.; methodology, M.-B.R.; software, T.L; validation, T.L.; formal analysis, M.-B.R.; investigation, T.L.; data curation, M.-B.R. and T.L.; writing–original draft preparation, M.-B.R. and T.L.; writing–review and editing, M.-B.R. and T.L.; supervision, M.-B.R.;

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Radac, M.B.; Precup, R.E.; Petriu, E.M. Model-free primitive-based iterative learning control approach to trajectory tracking of MIMO systems with experimental validation. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2925–2938. [CrossRef]
2. Sutton, R.S.; Barto, A.G. *In Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
3. Bertsekas, D.P.; Tsitsiklis, J.N. In *Neuro-Dynamic Programming*; Athena Scientific: Belmont, MA, USA, 1996.
4. Wang, F.Y.; Zhang, H.; Liu, D. Adaptive dynamic programming: an introduction. *IEEE Comput. Intell. Mag.* **2009**, *4*, 39–47. [CrossRef]
5. Lewis, F.; Vrabie, D.; Vamvoudakis, K.G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst. Mag.* **2012**, *32*, 76–105.
6. Lewis, F.; Vrabie, D.; Vamvoudakis, K.G. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circ. Syst. Mag.* **2009**, *9*, 76–105. [CrossRef]
7. Murray, J.; Cox, C.J.; Lendaris, G.G.; Saeks, R. Adaptive dynamic programming. *IEEE Trans. Syst. Man Cybern.* **2002**, *32*, 140–153. [CrossRef]
8. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
9. Kiumarsi, B.; Lewis, F.L.; Naghibi-Sistani, M.B.; Karimpour, A. Optimal tracking control of unknown discrete-time linear systems using input–output measured data. *IEEE Trans. Cybern.* **2015**, *45*, 2270–2779. [CrossRef]
10. Kiumarsi, B.; Lewis, F.L.; Modares, H.; Karimpour, A.; Naghibi-Sistani, M.B. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica* **2014**, *50*, 1167–1175. [CrossRef]
11. Radac, M.B.; Precup, R.E.; Roman, R.C. Model-free control performance improvement using virtual reference feedback tuning and reinforcement Q-learning. *Int. J. Syst. Sci.* **2017**, *48*, 1071–1083. [CrossRef]
12. Ernst, D.; Geurts, P.; Wehenkel, L. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.* **2005**, *6*, 2089–2099.
13. Hafner, R.; Riedmiller, M. Reinforcement learning in feedback control. Challenges and benchmarks from technical process control. *Mach. Learn.* **2011**, *84*, 137–169. [CrossRef]