European Journal of Control 73 (2023) 100880

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Learning-based state estimation and control using MHE and MPC schemes with imperfect models

Hossein Nejatbakhsh Esfahani*, Arash Bahari Kordabad, Wenqi Cai, Sebastien Gros

Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE INFO

Article history:
Received 24 August 2022
Revised 4 June 2023
Accepted 18 June 2023
Available online 21 June 2023

Recommended by Prof. T Parisini

Keywords:
Reinforcement learning
Moving horizon estimation
Model predictive control
Imperfect models

ABSTRACT

This paper presents a reinforcement learning-based observer/controller using Moving Horizon Estimation (MHE) and Model Predictive Control (MPC) schemes where the models used in the MHE-MPC cannot accurately capture the dynamics of the real system. We first show how an MHE cost modification can improve the performance of the MHE scheme such that a true state estimation is delivered even if the underlying MHE model is imperfect. A compatible Deterministic Policy Gradient (DPG) algorithm is then proposed to directly tune the parameters of both the estimator (MHE) and controller (MPC) in order to achieve the best closed-loop performance based on inaccurate MHE-MPC models. To demonstrate the effectiveness of the proposed learning-based estimator-controller, three numerical examples are illustrated.

1. Introduction

In the context of model-based control approaches, Model Predictive Control (MPC) is a well-known control scheme, which uses a dynamic model to predict the future behavior of the real system over a finite time horizon. At each time instant, MPC calculates the input and corresponding state sequence minimizing a given cost function while satisfying constraints over a given prediction horizon [26]. In many real applications, a state estimator (observer) is needed to provide an estimation of the current system states to the MPC scheme. In this paper, we adopt a Moving Horizon Estimation (MHE) scheme as a state observer, which is a simple choice in combination with an MPC scheme. MHE is an optimization-based state observer that works on a horizon window covering a limited history of past measurements [21].

Accurate models of dynamical systems are often difficult to obtain due to uncertainties and unknown dynamics. It is also worth noting that even if an accurate model is available, it may be in general too complex to be used in MHE and MPC schemes. However, if the model is imperfect, the inaccuracies can significantly degrade the performance of the MHE-MPC scheme. To cope with this problem, data-driven methods can be used in order to either improve the MPC and MHE models [3,16,18,31] or modify the MHE/MPC cost functions [10,11].

The data-driven MPC/MHE schemes mentioned above often incorporate Machine Learning (ML)-based techniques such as Reinforcement Learning (RL) and Gaussian Process (GP). RL is a powerful ML method for Markov Decision Processes (MDPs), which seeks to improve the closed-loop performance of the control policy deployed on the MDPs as observations are collected [28]. Most RL methods use a Deep Neural Network (DNN) to approximate either the optimal policy underlying the MDP directly or the action-value function from which the optimal policy can be indirectly extracted [2].

The idea of using an MPC scheme as a value function/policy approximator in the RL context was proposed in [11,35]. Specifically, the motivation was to replace the DNN-based approximators with the MPC schemes such that some challenging issues in the context of RL including stability guarantee and safety were addressed. In an MPC-based RL, it was established that an MPC scheme can generate jointly the optimal (action-) value function and optimal policy underlying an MDP even if the MPC model does not capture the real system dynamics accurately. As a data-driven MPC, the MPC-based RL framework has shown promising results for different applications [6–9,17]. Inspired by the researches mentioned above in the context of MPC-based RL, in the present paper, we will use an MHE-MPC scheme as a policy approximator for a deterministic policy gradient algorithm.

In some real-world control applications, the measurements available from the real system at a given time instant do not constitute a Markov state. In the context of RL, these systems are then formulated as Partially Observable MDPs (POMDPs) [15,36].

* Corresponding author.
E-mail address: hossein.n.esfahani@ntnu.no (H. Nejatbakhsh Esfahani).

https://doi.org/10.1016/j.ejcon.2023.100880

To tackle a POMDP, one solution is to formulate a belief MDP where the information about the current state is described as a probability distribution over the state space a.k.a belief state. Hence, POMDPs can be regarded as traditional MDPs using the concept of belief states as complete observable states [32].

Most previous works in the context of POMDPs rely on training a Neural Network (NN) or a Recurrent Neural Network (RNN) to summarise past observations and learn a policy based on DNN-based approximators [14,19,33]. An NN-based framework (posterior distributions over states) was proposed in [12,29] in order to estimate a belief state based on historical information. These NN-based algorithms are formulated as completely model-free approaches. Most recently, as a combined model-based/data-driven technique for dealing with POMDPs, a Q-learning method based on MHE-MPC with inaccurate models was developed in [10]. In this research, the authors proposed to integrate MHE and MPC to treat the hidden Markovian state evolution. More specifically, a structured solution by using MHE as a model-based approach was proposed to build a state from the measurement history.

In this paper, we seek to improve the performance of MHE-MPC as a combined observer/controller based on an inaccurate model. Assuming the real system is fully observable and the MHE model has a correct state structure, we show that both the arrival cost and the stage cost of the MHE scheme can be modified such that a perfect state estimation is delivered even if the underlying model is imperfect. However, the proposed method can arguably perform well on an incomplete model structure (partially observable), which is demonstrated by a numerical example. To tackle the performance degradation of the MHE scheme due to the use of an imperfect model, we propose to modify the MHE cost function rather than adapting the MHE model. An NN-based approximator is proposed to deliver the modified MHE cost. To achieve the best closed-loop performance even if the underlying MHE-MPC model is imperfect, we then propose to jointly tune the MHE-MPC parameters using a compatible Deterministic Policy Gradient (DPG) reinforcement learning algorithm.

The paper is structured as follows. The central theorem upon the cost modification of the MHE scheme using an imperfect model is detailed in Section 2. Section 3 describes a tractable approach for the MHE cost modification. Section 4 is dedicated to the parameterization method upon the MHE cost and the MPC scheme in order to formulate an adjustable and learning-based MHE-MPC scheme. To achieve the best closed-loop performance for an MHE-MPC scheme, a policy gradient-based RL algorithm is detailed in Section 5 to adjust the parameterized MHE cost function and learn a policy captured from a parameterized MHE-MPC scheme. Section 6 provides three numerical examples: 1) a linear system with model mismatch 2) a POMDP test case in which a smart building is described as an imperfect dynamical model and its climate is controlled by the proposed approach, and finally 3) a Continuous Stirred Tank Reactor (CSTR) as a nonlinear system is investigated.

Notation. a is a scalar while $\mathbf{a}$ is a vector. For n vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ we define $\text{col}(\mathbf{x}_1, \dots, \mathbf{x}_n) := [\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top]^\top$. $\mathbb{R}$ is the set of real numbers and $\mathbb{I}$ is the set of integers.

2. Modified MHE with imperfect model

In this section, we first consider an ideal stochastic MHE, which is formulated as a Full Information Estimation (FIE) problem. The FIE problems are fundamentally formulated based on an optimization problem in which the entire history of the measurements is used at each time instant [25]. We then formulate an MHE scheme using an imperfect model, and show that the stage cost function can be modified so that the MHE delivers the same estimation as an ideal MHE. At the end of this section, as opposed to the FIE version of the MHE, we will formulate a finite version of the modified MHE scheme in order to make it computationally tractable.

2.1. Stochastic MHE scheme

To formulate an ideal stochastic MHE scheme, we consider discrete dynamical systems evolving on a continuous state space over $\mathbb{R}^n$, with stochastic states $\mathbf{s}_k \in \mathcal{S} \subseteq \mathbb{R}^n$, where k denotes the time index. Let ϱ_k be a probability measure associated with the stochastic states as follows:

$$\mathbf{s}_k \sim \varrho_k(.) \tag{1}$$

We will consider a measure space for $\mathbf{s}_k$, which is equipped with the Lebesgue measure as a reference measure, and the set of Lebesgue-measurable sets as σ-algebra. Let us define stochastic dynamics as a conditional probability density as follows:

$$\zeta[\mathbf{s}_{k+1} \mid \mathbf{s}_k, \mathbf{a}_k] \tag{2}$$

where $\mathbf{s}_k, \mathbf{a}_k \in \mathcal{A} \subseteq \mathbb{R}^m$ and $\mathbf{s}_{k+1}$ are the current state-input pair and subsequent state, respectively, and $\mathcal{A}$ is the set of inputs available for the system.

Let us define a transition operator $\mathcal{T}_{\mathbf{a}_k} : \mathcal{M} \times \mathcal{A} \to \mathcal{M}$ as the map from a probability measure ϱ_k to its successor ϱ_{k+1} under input $\mathbf{a}_k$, and $\mathcal{M}$ is the set of probability measures over $\mathcal{S}$ such that the sequence of probability measures $\varrho_k \in \mathcal{M}, k = 0, \cdots, \infty$. We then define the Law of Total Probability (LTP) with stochastic dynamics (2) as follows:

$$\varrho_{k+1}(.) = \mathcal{T}_{\mathbf{a}_k}\varrho_k(.) = \int_{\mathcal{S}} \zeta[. \mid \mathbf{s}_k, \mathbf{a}_k]\varrho_k(\mathrm{d}\mathbf{s}_k) \tag{3}$$

Let us label $\mathbb{E}_{\mathbf{s}_k \sim \varrho_k}[.]$ the expected value operator with respect to probability measure $\varrho_k \in \mathcal{M}$. To formulate a stochastic MHE scheme a.k.a Full Information Estimation (FIE), its cost function can be derived using a functional stage cost where this functional is either an expectation or the Maximum A Posterior (MAP) [23]. In the present paper, we use an expectation to formulate a stochastic MHE under some conditions detailed in the remainder of the paper. We then define a value functional associated with the stochastic MHE scheme as follows:

$$V[\varrho_k, \mathbf{o}_k] := \sum_{i=-\infty}^{k} \gamma^{k-i} \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}\Big[L(\mathbf{s}_i, \mathbf{a}_{i-1}, \mathbf{y}_i)\Big], \tag{4}$$

where $\gamma \in (0, 1]$ is a discount factor, $\mathbf{y}_i \in \mathcal{Y} \subseteq \mathcal{S}$, $\mathbf{o}_k = \text{col}\{\mathbf{a}_{\dots,k-1}, \mathbf{y}_{\dots,k}\} \in \mathcal{O}$ is the available history of measurements up to time k, $L : \mathcal{S} \times \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$ is a fitting function. It is worth noting that the discounting above ensures the existence of the estimation problem on an infinite horizon for an MDP. However, the basic Theorem on the cost modification structure detailed in the remainder of this section also holds for the undiscounted setting, e.g., $\gamma = 1$. We assume that the forward transition operator $\mathcal{T}_{\mathbf{a}_k}$ has a backward transition operator $\mathcal{T}^{-1}_{\mathbf{a}_{i-1}}$ such that $\varrho_{i-1} = \mathcal{T}^{-1}_{\mathbf{a}_{i-1}}\varrho_i, \forall i \in \mathbb{I}_{\leq k}$. Note that we use the backward transition operator since an MHE scheme at the current time k is formulated based on past information. Then the aim of the stochastic MHE scheme is to find the best probability measure $\rho^\star$ as a function of $\mathbf{o}_k$ that minimizes $V[\varrho_k, \mathbf{o}_k]$. More specifically:

$$\rho_k^\star(\mathbf{o}_k) \in \arg\min_{\varrho_k} V[\varrho_k, \mathbf{o}_k] \tag{5}$$

However, we only have access to an imperfect model of (2) (typically deterministic). To cope with this issue, in the remainder of this section, we first develop the central theorem on the modification of the stochastic MHE schemes with imperfect models. We then propose a more practical formulation of the modified stochastic MHE in which a deterministic state estimation can be delivered.

2.2. Modification of the MHE cost function

The main contribution of this paper is described by the next theorem, where an MHE scheme equipped with a modified stage cost function is proposed to tackle the performance degradation due to an imperfect MHE model. It will be shown that one can, under some assumptions, find a modified MHE cost such that a probability measure equal to (5) is delivered even if the underlying model is inaccurate. In this paper, we define a cost functional $\Phi : \mathcal{M} \times \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$ such that this cost functional linearly depends on the stage cost function as follows:

$$\Phi[\varrho_i, \mathbf{a}_{i-1}, \mathbf{y}_i] = \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[L(\mathbf{s}_i, \mathbf{a}_{i-1}, \mathbf{y}_i)] \tag{6}$$

Note that the above equality is valid under some conditions detailed in the next section. Then, the value functional (4) can be rewritten as follows:

$$V[\varrho_k, \mathbf{o}_k] = \sum_{i=-\infty}^{k} \gamma^{k-i} \Phi[\varrho_i, \mathbf{a}_{i-1}, \mathbf{y}_i], \tag{7}$$

Let $\zeta_b\big[\mathbf{s}_{k-1} \,|\, \mathbf{s}_k, \mathbf{a}_{k-1}\big]$ and $\hat{\zeta}_b\big[\hat{\mathbf{s}}_{k-1} \,|\, \hat{\mathbf{s}}_k, \mathbf{a}_{k-1}\big]$ be a backward model of (2) and an imperfect model of ζ_b, respectively. We then label $\hat{\mathcal{T}}^{-1}_{\mathbf{a}_{i-1}}$ the corresponding imperfect backward transition operator. Now we propose to modify the MHE cost Φ detailed in the next theorem in order to cope with the performance degradation of an MHE scheme where the MHE model is imperfect. Hence, the corresponding value functional for a modified stochastic MHE scheme is formulated as follows:

$$\hat{V}[\hat{\varrho}_k, \mathbf{o}_k] := \sum_{i=-\infty}^{k} \gamma^{k-i} \hat{\Phi}[\hat{\varrho}_i, \mathbf{o}_i] \tag{8}$$

where $\hat{\varrho}_{i-1} = \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{i-1}} \hat{\varrho}_i, \ \forall i \in \mathbb{I}_{\leq k}$ and $\hat{\Phi} : \mathcal{M} \times \mathcal{O} \to \mathbb{R}$ is an MHE cost functional based on the measurement history for the model. Note that the arguments of the stage cost Φ in (7) include $\mathbf{a}_{i-1}, \mathbf{y}_i$ while they are shown as a measurement history $\mathbf{o}_i$ for the arguments of the modified stage cost $\hat{\Phi}$ in (8). More precisely, the modified stage cost will be a function of the measurement history at each time step i, which is detailed in the proof of the next theorem.

Analogous to the previous, we define the best probability measure resulting from the imperfect model, as follows:

$$\hat{\rho}^\star_k(\mathbf{o}_k) \in \arg\min_{\hat{\varrho}_k} \hat{V}[\hat{\varrho}_k, \mathbf{o}_k] \tag{9}$$

We then aim to propose $\hat{\Phi}$ such that $\hat{\rho}^\star_k(\mathbf{o}_k) = \rho^\star_k(\mathbf{o}_k)$. In the following, we make mild assumptions on the boundedness of the discounted value function.

Assumption 1. There exists a non-empty set of probability measures $\mathcal{M}_0 \subseteq \mathcal{M}$, including $\hat{\varrho}^\star_k$, such that for all $\hat{\varrho}_k \in \mathcal{M}_0$ and for all $\gamma \in (0, 1]$ it holds that

$$\big|\gamma^N V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}]\big| < \infty, \quad \forall N \in \mathbb{I}_{\geq 0} \tag{10}$$

where $\hat{\varrho}_{k-N} = \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{k-N}} \dots \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{k-1}} \hat{\varrho}_k$.

Assumption 2. For a discount factor $\gamma \in (0, 1]$ and $\hat{\varrho}_{k-N} \in \mathcal{M}_0, \forall N \in \mathbb{I}_{\geq 0}$:

$$\lim_{N \to \infty} \gamma^N V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] = 0 \tag{11}$$

Theorem 1. *Under Assumptions 1, 2, there exists a modified stage cost functional $\hat{\Phi} : \mathcal{M} \times \mathcal{O} \to \mathbb{R}$ such that the following equalities hold for all $\hat{\varrho}_k \in \mathcal{M}_0$ and all $\mathbf{o}_k \in \mathcal{O}$:*

$$\hat{V}[\hat{\varrho}_k, \mathbf{o}_k] = V[\hat{\varrho}_k, \mathbf{o}_k], \qquad \hat{\varrho}^\star_k(\mathbf{o}_k) = \varrho^\star_k(\mathbf{o}_k) \tag{12}$$

Proof. Let us define the modified stage cost functional $\hat{\Phi}$ as follows:

$$\hat{\Phi}[\hat{\varrho}_i, \mathbf{o}_i] = V[\hat{\varrho}_i, \mathbf{o}_i] - \gamma V\big[\hat{\mathcal{T}}^{-1}_{\mathbf{a}_{i-1}} \hat{\varrho}_i, \mathbf{o}_{i-1}\big] \tag{13}$$

By substituting the modified stage cost (13) in (8), the value functional then becomes a telescoping sum as follows:

$$\begin{aligned}
\hat{V}[\hat{\varrho}_k, \mathbf{o}_k] &= \sum_{i=-\infty}^{k} \gamma^{k-i} \hat{\Phi}[\hat{\varrho}_i, \mathbf{o}_i] \\
&= \sum_{i=-\infty}^{k} \gamma^{k-i} \big(V[\hat{\varrho}_i, \mathbf{o}_i] - \gamma V[\hat{\varrho}_{i-1}, \mathbf{o}_{i-1}]\big) \\
&= V[\hat{\varrho}_k, \mathbf{o}_k] - \gamma V[\hat{\varrho}_{k-1}, \mathbf{o}_{k-1}] \\
&\quad + \gamma V[\hat{\varrho}_{k-1}, \mathbf{o}_{k-1}] - \gamma^2 V[\hat{\varrho}_{k-2}, \mathbf{o}_{k-2}] \\
&\quad + \gamma^2 V[\hat{\varrho}_{k-2}, \mathbf{o}_{k-2}] - \dots - \lim_{N \to \infty} \gamma^N V[\hat{\varrho}_{-N}, \mathbf{o}_{-N}] \\
&= V[\hat{\varrho}_k, \mathbf{o}_k] - \lim_{N \to \infty} \gamma^N V[\hat{\varrho}_{-N}, \mathbf{o}_{-N}]
\end{aligned} \tag{14}$$

for all $\hat{\varrho}_k \in \mathcal{M}_0$. Note that under Assumption 1 all terms in (14) are bounded and the following equality holds:

$$\hat{V}[\hat{\varrho}_k, \mathbf{o}_k] = V[\hat{\varrho}_k, \mathbf{o}_k] \tag{15}$$

and under Assumption 2,

$$\arg\min_{\hat{\varrho}_k} \hat{V}[\hat{\varrho}_k, \mathbf{o}_k] = \arg\min_{\hat{\varrho}_k} V[\hat{\varrho}_k, \mathbf{o}_k] \tag{16}$$

implies $\hat{\varrho}^\star_k(\mathbf{o}_k) = \varrho^\star_k(\mathbf{o}_k)$ since $\hat{\varrho}^\star_k \in \mathcal{M}_0$. □

It is worth noting that the modified stage cost function (13) proposed as a cost modification is constructed based on a full history of the measurements. Hence, this fundamental observation can impact on the practical implementation of the modified cost. More specifically, the central Theorem 1 aims to show that there exists such a modification and to understand its structure. However, the proposed modification structure above is not tractable in terms of implementation since it is too complex to compute the modified stage cost (13) and apply it to the modified MHE scheme directly. To tackle this problem, we will provide a finite H-step structure of the modified stage cost in the next section. Finally, we will propose to construct an approximate structure of the modified stage cost using a Neural Network (NN) and adopt a reinforcement learning algorithm to learn the parameters of the NN in practice, which is detailed in the Section 4.

Although Theorem 1 shows that the modified stochastic MHE scheme with the corresponding value functional (8) can deliver a correct estimation of the probability measure using an imperfect model, this infinite-horizon model-based fitting problem requires an infinite amount of data, which makes this full information observer unsuitable in practice. To cope with this problem, we propose a more practical formulation detailed by the next theorem, which provides a finite-horizon stochastic MHE problem so that it delivers the same optimal density and value functional as (8).

It is worth mentioning that the proposed modified state cost (13) has been constructed based on the value functionals, and then Assumption 1 ensures that all intermediate terms (value functionals) appeared in the telescoping sum (14) cancel out. However, there are infinitely many intermediate terms in the telescoping sum (14) that must be bounded while Assumption 1 may not be satisfied for a situation with an arbitrarily large N, e.g., let us consider the case $\gamma = 1$, which then imposes the condition $\lim_{N \to \infty} V\big[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}\big] = 0$. Hence, the additional Assumption 2 is needed to establish the Theorem 1. To address this issue, one can consider a milder assumption with a specific horizon window N to be used in a finite-horizon MHE problem. We then provide the following assumption and develop the corresponding theorem.

Assumption 3. There exists a non-empty set of probability measures $\mathcal{M}_1 \subseteq \mathcal{M}$, including $\hat{\varrho}^\star_k$, such that for all $\hat{\varrho}_k \in \mathcal{M}_1$ and for all $\gamma \in (0, 1]$ it holds that

$$\big|\gamma^{N_0} V\big[\hat{\varrho}_{k-N_0}, \mathbf{o}_{k-N_0}\big]\big| < \infty, \quad 0 \leq N_0 \leq N \tag{17}$$

where $\hat{\varrho}_{k-N_0} = \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{k-N_0}} \dots \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{k-1}} \hat{\varrho}_k$ and $N < \infty$ is labeled the horizon window.

Note that this assumption is weaker than Assumption 1, indeed we have $\mathcal{M}_0 \subseteq \mathcal{M}_1$.

Theorem 2. *Consider the MHE scheme with a horizon window of N steps at the current time k :*

$$\hat{V}^N[\hat{\varrho}_k, \mathbf{o}_k] := \gamma^N \ell[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] + \sum_{i=k-N+1}^{k} \gamma^{k-i} \hat{\Phi}[\hat{\varrho}_i, \mathbf{o}_i], \tag{18a}$$

$$\hat{\rho}^{\star,N}_k(\mathbf{o}_k) \in \arg\min_{\hat{\varrho}_k} \hat{V}^N[\hat{\varrho}_k, \mathbf{o}_k] \tag{18b}$$

where $\ell : \mathcal{M} \times \mathcal{O} \to \mathbb{R}$ reads as an arrival cost functional. Then, under Assumption 3, the following equalities hold for all $\hat{\varrho}_k \in \mathcal{M}_1$ and all $\mathbf{o}_k \in \mathcal{O}$:

$$\hat{V}^N[\hat{\varrho}_k, \mathbf{o}_k] = V[\hat{\varrho}_k, \mathbf{o}_k], \quad \hat{\varrho}^{\star,N}_k(\mathbf{o}_k) = \varrho^\star_k(\mathbf{o}_k) \tag{19}$$

Proof. Let us define the modified stage cost $\hat{\Phi}$ as (13) and arrival cost ℓ as follows:

$$\ell[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] = V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] \tag{20}$$

By substituting the modified stage cost (13) and the modified arrival cost (20) in (18a), the value functional then becomes a telescoping sum as follows:

$$\begin{aligned}
\hat{V}^N[\hat{\varrho}_k, \mathbf{o}_k] &= \gamma^N V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] \\
&\quad + \sum_{i=k-N+1}^{k} \gamma^{k-i} \big(V[\hat{\varrho}_i, \mathbf{o}_i] - \gamma V[\hat{\varrho}_{i-1}, \mathbf{o}_{i-1}] \big) \\
&= \gamma^N V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] + V[\hat{\varrho}_k, \mathbf{o}_k] - \gamma V[\hat{\varrho}_{k-1}, \mathbf{o}_{k-1}] \\
&\quad + \gamma V[\hat{\varrho}_{k-1}, \mathbf{o}_{k-1}] - \gamma^2 V[\hat{\varrho}_{k-2}, \mathbf{o}_{k-2}] + \dots \\
&\quad + \gamma^{N-1} V[\hat{\varrho}_{k-N+1}, \mathbf{o}_{k-N+1}] - \gamma^N V[\hat{\varrho}_{k-N}, \mathbf{o}_{k-N}] \\
&= V[\hat{\varrho}_k, \mathbf{o}_k]
\end{aligned} \tag{21}$$

for all $\hat{\varrho}_k \in \mathcal{M}_1$, and

$$\arg\min_{\hat{\varrho}_k} \hat{V}^N[\hat{\varrho}_k, \mathbf{o}_k] = \arg\min_{\hat{\varrho}_k} V[\hat{\varrho}_k, \mathbf{o}_k] \tag{22}$$

delivers $\hat{\varrho}^{\star,N}_k(\mathbf{o}_k) = \varrho^\star_k(\mathbf{o}_k)$, since $\hat{\varrho}^\star_k \in \mathcal{M}_1$. Then it delivers (19). □

As an observation in the proposed finite-horizon MHE scheme (18a), the modified stage cost still depends on the complete measurement history despite using an arrival cost. Therefore, a practical modification of the stage cost will be detailed in the next section.

3. Tractable method for the MHE cost modification

Although Theorem 2 proposes the modified finite-horizon stochastic MHE as a more practical scheme than an infinite problem, there are still two implementation issues to address: (1) implementing a stage cost functional (13) is not tractable in practice because it is constructed based on time-varying value functionals in which the current distribution function $\hat{\varrho}_k$ as given probability measure at the current time k is difficult to model and calculate exactly. Then, it is reasonable to consider a function version of the cost functional in the modified MHE scheme. (2) implementing a modified stage cost based on the full measurement history is not tractable. In the rest of this section, we discuss the solutions to tackle these problems.

3.1. Modified stage cost function

To construct a practical cost modification based on the above results, one can consider a deterministic state estimation at the physical time k such that the modified cost is then constructed based on a value function instead of a value functional. Although this choice makes the implementation more practical, the estimation of a single state rather than a probability measure will sacrifice the MHE capability in order to explicitly describe the state estimation uncertainty. More specifically, we replace a belief state with a unique state such that the MHE solution cannot incorporate any information upon the uncertainty level of the current state.

In order to form an MHE scheme with a deterministic estimation of the state at time k, the proposed structure entails significant characteristics established by the next propositions. In the next Propositions 1,2, we first show that the backward transition operator $\mathcal{T}^{-1}$ is a linear transformation and the value functional $V[\varrho_i, \mathbf{o}_i]$ is linear in the probability measure.

Proposition 1. *The inverse of a linear operator $\mathcal{T}$ is a linear backward transition operator $\mathcal{T}^{-1}$ such that:*

$$\mathcal{T}^{-1}(\varrho + \bar{\varrho}) = \mathcal{T}^{-1}\varrho + \mathcal{T}^{-1}\bar{\varrho} \tag{23a}$$

$$\mathcal{T}^{-1}(\alpha\varrho) = \alpha\mathcal{T}^{-1}\varrho \tag{23b}$$

where the probability measures $\varrho, \bar{\varrho} \in \mathcal{M}$ and $\alpha \in \mathbb{C}$.

Proof.

$$\begin{aligned}
&\mathcal{T}^{-1}(\varrho + \bar{\varrho}) = \mathcal{T}^{-1}\big(\mathcal{T}\big(\mathcal{T}^{-1}\varrho\big) + \mathcal{T}\big(\mathcal{T}^{-1}\bar{\varrho}\big)\big) \\
&\mathcal{T}^{-1}\big(\mathcal{T}\big(\mathcal{T}^{-1}\varrho + \mathcal{T}^{-1}\bar{\varrho}\big)\big) = \mathcal{T}^{-1}\varrho + \mathcal{T}^{-1}\bar{\varrho}
\end{aligned} \tag{24}$$

and

$$\begin{aligned}
&\mathcal{T}^{-1}(\alpha\varrho) = \mathcal{T}^{-1}\big(\alpha\mathcal{T}\big(\mathcal{T}^{-1}\varrho\big)\big) \\
&\mathcal{T}^{-1}\big(\mathcal{T}\big(\alpha\mathcal{T}^{-1}\varrho\big)\big) = \alpha\mathcal{T}^{-1}\varrho
\end{aligned} \tag{25}$$

Then, the backward operator $\mathcal{T}^{-1}$ fulfills the requirements of a linear transformation. □

Proposition 2. *The value functional $V[\varrho_i, \mathbf{o}_i]$ is linear in the probability measure.*

Proof. According to (4), the value functional $V[\varrho_i, \mathbf{o}_i]$ at time step i is defined as follows:

$$\begin{aligned}
V[\varrho_i, \mathbf{o}_i] &= \sum_{j=-\infty}^{i} \gamma^{i-j} \mathbb{E}_{\mathbf{s}_j \sim \varrho_j} \Big[L\big(\mathbf{s}_j, \mathbf{a}_{j-1}, \mathbf{y}_j\big) \Big] \\
&= \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[L(\mathbf{s}_i, \cdot, \cdot)] + \gamma \mathbb{E}_{\mathbf{s}_{i-1} \sim \varrho_{i-1}}[L(\mathbf{s}_{i-1}, \cdot, \cdot)] + \dots
\end{aligned} \tag{26}$$

First a backward transition $\mathcal{T}^{-1}$ is a linear transformation, as established by Proposition 1. We then conclude that each ϱ_j is linear in ϱ_i. Hence, each expected stage cost term, appearing on the right-hand side of (26) is linear in ϱ_i. Then, the summation of discounted expectations on the right-hand side of the above equation will also be linear in the probability measure, which proves the proposition. □

Now, in the next proposition, we will show a relation between the value functional and the value function such that the following assumption must be satisfied:

Assumption 4. Let us assume that the expected value function v is bounded for all $\varrho_i \in \mathcal{M}$ and $\mathbf{s}_i \in \mathcal{S}$:

$$\mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[|v(\mathbf{s}_i, \mathbf{o}_i)|] < \infty \tag{27}$$

Note that the assumption above ensures that the expected value of the value function $v(\mathbf{s}_i, \mathbf{o}_i)$ will remain finite for all $\varrho_i \in \mathcal{M}$ and $\mathbf{s}_i \in \mathcal{S}$, a harmless restriction in practice.

Proposition 3. *Let the value function $v(\mathbf{s}_i, \mathbf{o}_i)$ be a Lebesgue measurable function (a.k.a Borel measurable) on the σ-algebra of Borel sets, and the probability measure ϱ_i be a compactly supported continuous function. A value functional then can be represented as an expected value function as follows:*

$$V[\varrho_i, \mathbf{o}_i] = \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[v(\mathbf{s}_i, \mathbf{o}_i)] \tag{28}$$

Proof. Under Assumption 4 and the linearity of $V[\varrho_i, \mathbf{o}_i]$ in ϱ_i, see Proposition 2, the proof follows the Riesz-Markov theorem, see [4], chapter 9, page 105, such that:

$$V[\varrho_i, \mathbf{o}_i] = \int_{\mathcal{S}} v(\mathbf{s}_i, \mathbf{o}_i)\varrho_i(d\mathbf{s}_i) = \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[v(\mathbf{s}_i, \mathbf{o}_i)]$$

and a value function $v(\mathbf{s}_i, \mathbf{o}_i)$ can be found so that the equality above holds. □

Now we choose the probability measure at time k as a Dirac measure centered at the current state such that $\varrho_k = \delta_{\mathbf{s}_k}(.)$. According to Proposition 3, the value functional then becomes a value function:

$$V[\varrho_k, \mathbf{o}_k] = \mathbb{E}_{\mathbf{s}_k \sim \delta_{\mathbf{s}_k}(.)}[v(\mathbf{s}_k, \mathbf{o}_k)] = v(\mathbf{s}_k, \mathbf{o}_k) \tag{29}$$

Then, the ideal stochastic MHE scheme with the value functional (4) can be rewritten as follows:

$$v(\mathbf{s}_k, \mathbf{o}_k) := \sum_{i=-\infty}^{k} \gamma^{k-i} \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}\Big[L(\mathbf{s}_i, \mathbf{a}_{i-1}, \mathbf{y}_i)\Big], \tag{30a}$$

$$\mathbf{s}_k^\star(\mathbf{o}_k) \in \arg\min_{\mathbf{s}_k} v(\mathbf{s}_k, \mathbf{o}_k) \tag{30b}$$

where $\varrho_{i-1} = \mathcal{T}_{\mathbf{a}_{i-1}}^{-1}\varrho_i$ and $\varrho_k = \delta_{\mathbf{s}_k}(\cdot)$. We next show that the modified stage cost functional (13) can be rewritten as stage cost function at each time step i, which is constructed based on the value functions defined as (30a).

We first remind that the linear relation (6) between the stage cost functional and the stage cost function is valid based on the same synthesis as Proposition 3 using two underlying conditions: 1) expected stage cost function is bounded $\mathbb{E}_{\mathbf{s}_i \sim \varrho_i}[|L(\mathbf{s}_i, \mathbf{a}_{i-1}, \mathbf{y}_i)|] < \infty$ 2) stage cost functional Φ is linear in ϱ_i. Hence, this relation also holds for $\hat{\Phi}$ in (13) considering the next remark.

Remark 1. The modified stage cost functional (13) is also linear in the probability measures since it is defined based on a linear equation of the value functionals, which are linear in the probability measures, see Proposition 2.

Now the modified stage cost functional can be described as:

$$\hat{\Phi}[\hat{\varrho}_i, \mathbf{o}_i] = \mathbb{E}_{\hat{\mathbf{s}}_i \sim \hat{\varrho}_i}\big[\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i)\big] \tag{31}$$

where $\hat{L} : \mathcal{S} \times \mathcal{O} \to \mathbb{R}$ reads the modified stage cost function.

By considering Proposition 3, equality (31) and adopting $\hat{\varrho}_i = \delta_{\hat{\mathbf{s}}_i}(\cdot)$, one can obtain a practical cost modification of (13) at each time step i as follows:

$$\begin{aligned}\mathbb{E}_{\hat{\mathbf{s}}_i \sim \delta_{\hat{\mathbf{s}}_i}(\cdot)}\big[\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i)\big] &= \mathbb{E}_{\hat{\mathbf{s}}_i \sim \delta_{\hat{\mathbf{s}}_i}(\cdot)}\big[v(\hat{\mathbf{s}}_i, \mathbf{o}_i)\big] - \gamma \mathbb{E}_{\hat{\mathbf{s}}_{i-1} \sim \hat{\varrho}_{i-1}}\big[v(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})\big] \\ &= v(\hat{\mathbf{s}}_i, \mathbf{o}_i) - \gamma \mathbb{E}_{\hat{\mathbf{s}}_{i-1} \sim \hat{\varrho}_{i-1}}\big[v(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})\big]\end{aligned} \tag{32}$$

where $\hat{\varrho}_{i-1} = \hat{\zeta}_b[\,\cdot \mid \hat{\mathbf{s}}_i, \mathbf{a}_{i-1}]$.

Hence, the modified cost functional $\hat{\Phi}$ can then be replaced by the stage cost function $\hat{L}$ at time step i in practice:

$$\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i) = v(\hat{\mathbf{s}}_i, \mathbf{o}_i) - \gamma \mathbb{E}_{\hat{\mathbf{s}}_{i-1} \sim \hat{\varrho}_{i-1}}\big[v(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})\big] \tag{33}$$

We then adopt the above-modified stage cost function to formulate a modified MHE scheme using an imperfect model in practice. Hence, the modified stochastic MHE scheme based on the value (stage cost) function instead of the value (stage cost) functional (8) is formulated as follows:

$$\hat{v}(\hat{\mathbf{s}}_k, \mathbf{o}_k) := \sum_{i=-\infty}^{k} \gamma^{k-i} \mathbb{E}_{\hat{\mathbf{s}}_i \sim \hat{\varrho}_i}\Big[\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i)\Big] \tag{34a}$$

$$\hat{\mathbf{s}}_k^\star(\mathbf{o}_k) \in \arg\min_{\hat{\mathbf{s}}_k} \hat{v}(\hat{\mathbf{s}}_k, \mathbf{o}_k) \tag{34b}$$

where $\hat{\varrho}_{i-1} = \hat{\mathcal{T}}_{\mathbf{a}_{i-1}}^{-1}\hat{\varrho}_i$ and $\hat{\varrho}_k = \delta_{\hat{\mathbf{s}}_k}(\cdot)$.

Now, according to the developments above, we have shown that the modified MHE scheme with a stage cost functional $\hat{\Phi}$ can be formulated as a tractable MHE (34) in which the modified cost function $\hat{L}$ (33) is practically constructed based on the value functions instead of the value functionals. The following corollary shows that the structure (34) can still preserve the property established by Theorem 1.

Corollary 1. *By adopting the same approach as was detailed to prove Theorem 1 and under the assumption*

$$\gamma^N \mathbb{E}_{\hat{\mathbf{s}}_{k-N} \sim \hat{\varrho}_{k-N}}\Big[\big|v(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N})\big|\Big] < \infty, \qquad \forall N \in \mathbb{I}_{\geq 0} \tag{35}$$

one can show that the following equalities hold:

$$\hat{v}(\cdot) = v(\cdot), \qquad \hat{\mathbf{s}}_k^\star = \mathbf{s}_k^\star \tag{36}$$

Proof. By substituting the modified stage cost function (33) in the value function associated to the problem (34) and using a telescoping sum argument, one can observe that:

$$\begin{aligned}\hat{v}(\hat{\mathbf{s}}_k, \mathbf{o}_k) = \sum_{i=-\infty}^{k} \gamma^{k-i} \mathbb{E}_{\hat{\mathbf{s}}_i \sim \hat{\varrho}_i}\Big[&v(\hat{\mathbf{s}}_i, \mathbf{o}_i) \\ &- \gamma \mathbb{E}_{\hat{\mathbf{s}}_{i-1} \sim \hat{\varrho}_{i-1}}\big[v(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})\big]\Big] = v(\hat{\mathbf{s}}_k, \mathbf{o}_k)\end{aligned} \tag{37}$$

and

$$\arg\min_{\hat{\mathbf{s}}_k} \hat{v}(\hat{\mathbf{s}}_k, \mathbf{o}_k) = \arg\min_{\hat{\mathbf{s}}_k} v(\hat{\mathbf{s}}_k, \mathbf{o}_k) \tag{38}$$

results in $\hat{\mathbf{s}}_k^\star(\mathbf{o}_k) = \mathbf{s}_k^\star(\mathbf{o}_k)$. □

3.2. Tractable modified stage cost

In the proposed modified stage cost function (33), the value functions captured from the MHE scheme (30) are based on the complete measurement history, and the amount of historical data is growing at each time instant. Hence, constructing the corresponding modified stage cost is intractable in practice. We then propose to formulate a finite version (H-step) of the optimization problem (30) so that the corresponding value function reads as:

$$\begin{aligned}v^H(\mathbf{s}_k, \mathbf{o}_k) := \ &\gamma^H \mathbb{E}_{\mathbf{s}_{k-H} \sim \varrho_{k-H}}[Z_{k-H}(\mathbf{s}_{k-H}, \mathbf{o}_{k-H})] \\ &+ \sum_{i=k-H+1}^{k} \gamma^{k-i} \mathbb{E}_{\mathbf{s}_i \sim \varrho_i}\Big[L(\mathbf{s}_i, \mathbf{a}_{i-1}, \mathbf{y}_i)\Big]\end{aligned} \tag{39}$$

where $\varrho_{i-1} = \mathcal{T}_{\mathbf{a}_{i-1}}^{-1}\varrho_i$ and $\varrho_k = \delta_{\mathbf{s}_k}(\cdot)$. Notice that the cost term $Z_{k-H}(\mathbf{s}_{k-H}, \mathbf{o}_{k-H})$ is labeled the exact arrival cost function, which summarizes the effects of past information before time $k-H$. Then, under an exact arrival cost, the stochastic MHE scheme based on the value function (39) can be regarded as an ideal MHE scheme, i.e.,

$$v^H(\mathbf{s}_k, \mathbf{o}_k) = v(\mathbf{s}_k, \mathbf{o}_k), \tag{40}$$

Now the modified stage cost (33) can be rewritten based on the value function (39) as follows:

$$\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i) = v^H(\hat{\mathbf{s}}_i, \mathbf{o}_i) - \gamma \mathbb{E}_{\hat{\mathbf{s}}_{i-1} \sim \hat{\varrho}_{i-1}}\big[v^H(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})\big] \tag{41}$$

Note that the expectation above is taken over the imperfect model whereas the expected values appeared in the definition of the value functions $v^H(\cdot)$ in (39) are on the real system. Although the value function (39) is based on the full measurement history, the implementation of the modified stage cost (41) will be finally tractable for a finite MHE scheme with a horizon N proposed in the next theorem. More specifically, the full history of the measurements due to the arrival cost term of v^H is transferred to the arrival cost Z_{k-N}, which can be approximated in practice. A practical implementation based on the mentioned argument above will be discussed in detail in Section 4. Now we develop the next theorem for a modified MHE scheme based on the above-modified stage cost function. To this end, let us consider the following assumption:

Assumption 5. There exists a non-empty set $\mathcal{S}_0 \subseteq \mathcal{S}$ such that for all $\hat{\mathbf{s}} \in \mathcal{S}_0$ and for all $\gamma \in (0, 1]$ it holds that

$$\left|\gamma^{N_0}\mathbb{E}_{\hat{\mathbf{s}}_{k-N_0}\sim\hat{\varrho}_{k-N_0}}\left[v^H\left(\hat{\mathbf{s}}_{k-N_0}, \mathbf{o}_{k-N_0}\right)\right]\right| < \infty, \qquad 0 \le N_0 \le N \tag{42}$$

where N is labeled a specific horizon window. Note that the expectation in the above inequality is taken over the imperfect model density $\hat{\varrho}_{k-N-1} = \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{k-N-1}}\hat{\varrho}_{k-N}$.

Then, the following theorem is defined under the above-mentioned assumption:

Theorem 3. *There exists an exact arrival cost function including some prior information as available observation $Z_{k-N} : \mathcal{S} \times \mathcal{O} \to \mathbb{R}$ and a modified stage cost function $\hat{L} : \mathcal{S} \times \mathcal{O} \to \mathbb{R}$. We then formulate the following finite stochastic MHE scheme at the current time k:*

$$\begin{aligned}\hat{v}^N\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) := {} & \gamma^N\mathbb{E}_{\hat{\mathbf{s}}_{k-N}\sim\hat{\varrho}_{k-N}}\left[Z_{k-N}\left(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}\right)\right] \\ & + \sum_{i=k-N+1}^{k}\gamma^{k-i}\mathbb{E}_{\hat{\mathbf{s}}_i\sim\hat{\varrho}_i}\left[\hat{L}\left(\hat{\mathbf{s}}_i, \mathbf{o}_i\right)\right]\end{aligned} \tag{43a}$$

$$\hat{\mathbf{s}}_k^{\star,N} \in \arg\min_{\hat{\mathbf{s}}_k}\hat{v}^N\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) \tag{43b}$$

where $\hat{\varrho}_{i-1} = \hat{\mathcal{T}}^{-1}_{\mathbf{a}_{i-1}}\hat{\varrho}_i$, $\hat{\varrho}_k = \delta_{\hat{\mathbf{s}}_k}(\cdot)$.

Then under Assumption 5, the MHE scheme above will deliver the following equalities for all $\hat{\mathbf{s}}_k \in \mathcal{S}_0$:

$$\hat{v}^N\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) = v^H\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right), \quad \hat{\mathbf{s}}_k^{\star,N}(\mathbf{o}_k) = \mathbf{s}_k^{\star}(\mathbf{o}_k) \tag{44}$$

Proof. Let us select the modified stage cost (41) and define the arrival cost Z_{k-N} as follows:

$$Z_{k-N} = v^H\left(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}\right) \tag{45}$$

By substituting the modified stage cost function (41) and the arrival cost function (45) in the value function (43a), it then becomes a telescoping sum as follows:

$$\begin{aligned}\hat{v}^N\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) = {} & \gamma^N\mathbb{E}_{\hat{\mathbf{s}}_{k-N}\sim\hat{\varrho}_{k-N}}\left[v^H\left(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}\right)\right] \\ & + \sum_{i=k-N+1}^{k}\gamma^{k-i}\mathbb{E}_{\hat{\mathbf{s}}_i\sim\hat{\varrho}_i}\left[v^H\left(\hat{\mathbf{s}}_i, \mathbf{o}_i\right) - \gamma\mathbb{E}_{\hat{\mathbf{s}}_{i-1}\sim\hat{\varrho}_{i-1}}\left[v^H\left(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1}\right)\right]\right] \\ = {} & \gamma^N\mathbb{E}_{\hat{\mathbf{s}}_{k-N}\sim\hat{\varrho}_{k-N}}\left[v^H\left(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}\right)\right] + v^H\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) \\ & - \gamma\mathbb{E}_{\hat{\mathbf{s}}_{k-1}\sim\hat{\varrho}_{k-1}}\left[v^H\left(\hat{\mathbf{s}}_{k-1}, \mathbf{o}_{k-1}\right)\right] \\ & + \gamma\mathbb{E}_{\hat{\mathbf{s}}_{k-1}\sim\hat{\varrho}_{k-1}}\left[v^H\left(\hat{\mathbf{s}}_{k-1}, \mathbf{o}_{k-1}\right)\right] + \dots \\ & - \gamma^N\mathbb{E}_{\hat{\mathbf{s}}_{k-N}\sim\hat{\varrho}_{k-N}}\left[v^H\left(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}\right)\right] \\ = {} & v^H\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right)\end{aligned} \tag{46}$$

for all $\hat{\mathbf{s}}_k \in \mathcal{S}_0$, and

$$\arg\min_{\hat{\mathbf{s}}_k}\hat{v}^N\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) = \arg\min_{\hat{\mathbf{s}}_k}v^H\left(\hat{\mathbf{s}}_k, \mathbf{o}_k\right) \tag{47}$$

delivers $\hat{\mathbf{s}}_k^{\star,N}(\mathbf{o}_k) = \mathbf{s}_k^{\star}(\mathbf{o}_k)$. □

Note that the horizon H is the length of the measurement history used in the modified stage cost in the MHE scheme (43) with a horizon of length N. In the next section, we will describe how this measurement history of length H can be used in the proposed convex neural network to modify the MHE stage cost in practice. It is worth noting that the horizon H may be selected larger than N to capture the modified stage cost accurately. However, one can choose a small length of H in order to provide an acceptable trade-off between the computational effort and the approximate value captured from the neural network.

4. Proposed learning-based MHE-MPC scheme

4.1. Practical implementation

In what follows, we provide a practical version of the modified MHE scheme (43). We then discuss the approaches in order to approximate both the arrival cost Z_{k-N} and the modified stage cost $\hat{L}\left(\hat{\mathbf{s}}_i, \mathbf{o}_i\right)$.

4.1.1. Learning-based arrival cost

To approximate the arrival cost, we adopt a common approach so that the arrival cost takes the form of a quadratic function as follows:

$$\hat{Z}_{k-N} = \left\|\hat{s}_{k-N} - \tilde{\mathbf{s}}_{k-N}\right\|^2_{\Pi^{-1}_{k-N}} \tag{48}$$

where $\tilde{\mathbf{s}}_{k-N}$ is obtained as:

$$\tilde{\mathbf{s}}_{k-N} = \mathbf{s}^{\star}_{k-N|k-1} \tag{49}$$

Note that $\mathbf{s}^{\star}_{k-N|k-1}$ is the first element of the horizon window at the previous physical time $k-1$. The prior weighting Π_{k-N} is obtained from the Kalman filter covariance update rule [30]:

$$\Pi_{k+1} = A_k\Pi_kA_k^\top - A_k\Pi_kC_k^\top\left(C_k\Pi_kC_k^\top + R\right)^{-1}C_k\Pi_kA_k^\top \tag{50}$$

initialized with the covariance matrix of the initial state Π_0. Let $f\left(\hat{\mathbf{s}}, \mathbf{a}\right)$ be a nonlinear model as a deterministic approximation of (2). The matrices A_k and C_k are then obtained by linearization as follows:

$$A_k = \frac{\partial f}{\partial\hat{\mathbf{s}}}|_{\hat{\mathbf{s}}_{k|k-1}}, \qquad C_k = \frac{\partial h}{\partial\hat{\mathbf{s}}}|_{\hat{\mathbf{s}}_{k|k-1}} \tag{51}$$

and R is the covariance of the output noise ν_k where the measurements are delivered as $\mathbf{y}_k = h(\hat{\mathbf{s}}_k) + \nu_k$. However, the approach detailed above is based on classic Kalman filtering that may not be the best choice from a parameterization standpoint where the model is imperfect. More specifically, the update rule (50) cannot deliver a perfect approximation of Π since the matrices A, C captured, respectively, from the dynamical model f and the measurement model h are imperfect. To tackle this problem, we propose to adopt reinforcement learning in order to adjust the entries of the matrices A_k^θ, C_k^θ and the covariance matrix R_θ used in (50), where θ will be parameters that can be adjusted via RL. Then, the parameterized covariance update rule reads as:

$$\begin{aligned}\Pi_{k+1} = {} & A_k^\theta\Pi_k\left(A_k^\theta\right)^\top \\ & - A_k^\theta\Pi_k\left(C_k^\theta\right)^\top\left(C_k^\theta\Pi_k\left(C_k^\theta\right)^\top + R_\theta\right)^{-1}C_k^\theta\Pi_k\left(A_k^\theta\right)^\top\end{aligned} \tag{52}$$

It is worth noting that the policy π captured from the MHE-MPC scheme will have an extra state Π_k, which is obtained from the above dynamics. More specifically, Π_k has its own dynamics in the MHE scheme such that the state estimation and the policy delivered, respectively, from the MHE and MHE-MPC will depend on Π_k. We then consider the effect of Π_k on the policy gradient in an MHE/MPC-based reinforcement learning detailed in the next section.

4.1.2. Learning-based MHE stage cost

According to Theorem 3, a finite stochastic MHE scheme can deliver a true state estimation using an imperfect model of the real system by adopting a cost modification.

Remark 2. In this paper, we denote the true state estimation (perfect estimation) by the estimation captured from the FIE problems with the correct model in (30) and (39).

In this theorem, we have proposed to construct a modified stage cost (41) based on the H-step value function (39), and we practically propose to approximate this modified stage cost. To this end, let us consider the MHE scheme (43) and the value function (39) where $\hat{\varrho}_i = \delta_{\hat{\mathbf{s}}_i}(\cdot)$. We then observe that all expected arrival cost functions $Z_{i-H}, Z_{i-H-1}, i = k-N+1, \cdots, k$ including entire history can be transferred to the arrival cost Z_{k-N}. More precisely, the time step i used in the modified stage cost of the MHE scheme (43) is in the interval $i = k-N+1, \cdots, k$, and this stage cost defined in (41) is basically constructed based on the value functions $v^H(\hat{\mathbf{s}}_i, \mathbf{o}_i)$ and $v^H(\hat{\mathbf{s}}_{i-1}, \mathbf{o}_{i-1})$ captured from (39). The corresponding arrival costs then read as Z_{i-H} and Z_{i-H-1} for $i = k-N+1, \cdots, k$. Hence, the modified stage cost can practically be constructed based on a finite history as follows:

$$\hat{L}(\hat{\mathbf{s}}_i, \mathbf{o}_i^H) = L(\hat{\mathbf{s}}_i, \mathbf{a}_{i-1}, \mathbf{y}_i) + L_h(\hat{\mathbf{s}}_{i-H,\cdots,i-1}, \mathbf{o}_{i-1}^H) \tag{53}$$

where the cost term L_h is constructed based on a finite history and $\mathbf{o}_i^H = \text{col}\{\mathbf{y}_{i-H+1}, \ldots, \mathbf{y}_i, \mathbf{a}_{i-H}, \ldots, \mathbf{a}_{i-1}\}$. We then propose to somehow approximate this part of the modified stage cost $\hat{L}$ in practice.

As one practical solution to approximate L_h, one can use a Neural Network (NN) as follows:

$$\hat{L}_\theta(\hat{\mathbf{s}}_i, \mathbf{o}_i^H) \approx L_{\theta_0}(\hat{\mathbf{s}}_i, \mathbf{a}_{i-1}, \mathbf{y}_i) + L_{\text{NN}}(\mathbf{Y}_i, \theta_{\text{NN}}) \tag{54}$$

where L_{θ_0} is a parameterized least-squares cost at the current time step i used for an *output noise MHE* scheme [21] and

$$\begin{aligned} \mathbf{Y}_i &= \text{col}\{\hat{\mathbf{s}}_{i-H,\cdots,i-1}, \mathbf{o}_{i-1}^H\} \\ \mathbf{o}_{i-1}^H &= \text{col}\{\mathbf{y}_{i-H}, \ldots, \mathbf{y}_{i-1}, \mathbf{a}_{i-H-1}, \ldots, \mathbf{a}_{i-2}\} \end{aligned} \tag{55}$$

Note that $\mathbf{o}_{i-1}^H \in \mathcal{O}^H \subset \mathcal{O}$ is regarded as a finite history of the measurements and $\mathbf{Y}_i \in \mathbb{R}^{n_y}$ is labeled the Neural Network (NN) input. To adjust the parameters θ_0, θ_{NN}, we will use a reinforcement learning algorithm based on the policy gradient method.

Neural networks are well-known universal function approximators so an NN including three layers is capable to approximate any continuous multivariate function down to prescribed accuracy, if there are no constraints on the number of neurons [34].

We then propose to use a convex class of NNs to approximate L_{NN} in the MHE stage cost (54). While enforcing convexity in the MHE stage cost does not imply that the overall MHE problem is convex, using a non-convex stage cost will often require significantly more caution in providing an initial guess for the NLP solver tackling the MHE scheme than if a convex stage cost is used. To preserve the convexity of the MHE stage cost function (54), we then propose to compute L_{NN} using an Input Convex Neural Network (ICNN). In this type of neural network, the partial weights meet certain constraints such that the output of the ICNN is a convex function of the input [1,5]. Compared to building conventional neural networks, ICNN structures must meet two additional requirements: (1) activation functions are convex and non-decreasing (2) the weights of NN are constrained to be non-negative. As a form of ICNN, we choose a Fully Input Convex NN (FICNN) architecture since the scalar output of the network is convex with respect to all inputs.

Let us consider a l-layer FICNN over $\mathbf{Y}_i$ in order to estimate $L_{\text{NN}}(\mathbf{Y}_i, \theta_{\text{NN}})$ as follows:

$$z_{j+1} = g_j(W_j^{(z)} z_j + W_j^{(y)} \mathbf{Y}_i + b_j), \tag{56a}$$

$$L_{\text{NN}}(\mathbf{Y}_i, \theta) = c \cdot z_l \tag{56b}$$

$$s.t. \quad W_{1:l-1}^{(z)} \geq 0, W_0^{(z)} \equiv 0, z_0 \equiv 0 \tag{56c}$$

where $j = 0, \ldots, l-1$ and $z_j \in \mathbb{R}^{n_y \times 1}$ denotes the middle layers (layer activations). The neural network weights are $W_j^{(z)} \in \mathbb{R}^{n_y \times n_y}, W_j^{(y)} \in \mathbb{R}^{n_y \times n_y}, b_j \in \mathbb{R}^{n_y \times 1}, c \in \mathbb{R}^{1 \times n_y}$. Note that c is considered as the connection weight between the output layer and the last middle layer. Then, $\theta_{NN} = \left\{W_{1:l-1}^{(z)}, W_{0:l-1}^{(y)}, b_{0:l-1}, c\right\}$ are the modifiable weights, and g_j are nonlinear activation functions (convex and non-decreasing, e.g., Rectified Linear Unit *ReLU*).

Then, the reformulated MHE scheme (43) reads as:

$$\begin{aligned} \hat{v}^N(\hat{\mathbf{s}}_k, \mathbf{o}_k) := \gamma^N \mathbb{E}_{\hat{\mathbf{s}}_{k-N} \sim \hat{\varrho}_{k-N}}\left[\hat{Z}_{k-N}(\hat{\mathbf{s}}_{k-N}, \mathbf{o}_{k-N}^H)\right] \\ + \sum_{i=k-N+1}^{k} \gamma^{k-i} \mathbb{E}_{\hat{\mathbf{s}}_i \sim \hat{\varrho}_i}\left[\hat{L}_\theta(\hat{\mathbf{s}}_i, \mathbf{o}_i^H)\right] \end{aligned} \tag{57a}$$

$$\hat{\mathbf{s}}_k^{\star,N} \in \arg\min_{\hat{\mathbf{s}}_k} \hat{v}^N(\hat{\mathbf{s}}_k, \mathbf{o}_k) \tag{57b}$$

where $\hat{\varrho}_{i-1} = \hat{\mathcal{T}}_{\mathbf{a}_{i-1}}^{-1} \hat{\varrho}_i, \quad \hat{\varrho}_k = \delta_{\hat{\mathbf{s}}_k}(\cdot)$.

In the remainder of this section, we practically formulate a deterministic version of the above MHE scheme (57) with a fully parameterized cost function including arrival and stage cost. We then propose a parameterization for the MPC scheme to deliver a policy approximation required in the context of policy gradient RL.

4.2. Deterministic MHE scheme with adjustable cost

As a result of Theorem 3, a finite optimization-based state estimation scheme with an imperfect model can deliver a true state estimation by modifying the stage and arrival costs. As a practical approach, we proposed to leverage NN in approximating the modified stage cost function (54). One can also choose a parameterization method on the arrival cost detailed in the previous section. Finally, a reinforcement learning algorithm is used to adjust all parameters.

Note that the theory proposed is very generic so that the acquired result holds e.g., for the states of a stochastic dynamical model being estimated by an MHE scheme based on an inaccurate deterministic model. Hence, the transition model $\hat{\zeta}[\hat{\mathbf{s}}_{i+1} \mid \hat{\mathbf{s}}_i, \mathbf{a}_i]$ trivially includes deterministic models as:

$$\hat{\zeta}[\hat{\mathbf{s}}_{i+1} \mid \hat{\mathbf{s}}_i, \mathbf{a}_i] = \delta\left(\hat{\mathbf{s}}_{i+1} - \hat{f}^{\text{MHE}}(\hat{\mathbf{s}}_i, \mathbf{a}_i)\right) \tag{58}$$

We then propose to formulate the following parameterized MHE scheme:

$$\begin{aligned} \mathbf{s}^\star_{k-N_{\text{MHE}},\ldots,k} = \arg\min_{\hat{\mathbf{s}}} \gamma^{N_{\text{MHE}}} \hat{Z}_\theta(\hat{\mathbf{s}}_{k-N_{\text{MHE}}}, \tilde{\mathbf{s}}) \\ + \sum_{i=k-N_{\text{MHE}}+1}^{k} \gamma^{k-i} \hat{L}_\theta(\hat{\mathbf{s}}_i, \mathbf{o}_i^H) \end{aligned} \tag{59a}$$

$$\text{s.t.} \qquad \hat{\mathbf{s}}_{i+1} = \hat{f}_\theta^{\text{MHE}}(\hat{\mathbf{s}}_i, \mathbf{a}_i) \tag{59b}$$

where $\hat{Z}_\theta = \left\|\hat{\mathbf{s}}_{k-N_{\text{MHE}}} - \tilde{\mathbf{s}}\right\|_{\Pi_k}^2$ and Π_k is obtained from the parameterized updating rule (52) and $\tilde{\mathbf{s}}$ is the available estimation $\mathbf{s}^\star_{k-N_{\text{MHE}}}$ at time $k-1$. Note that in the case of linear systems, another solution to adjust the arrival cost is to directly modify the arrival cost by adjusting the corresponding positive weight matrix Π_θ using e.g., Semi-Definite Programming (SDP).

4.3. Parameterized MPC scheme

Let us define the closed-loop performance of a parameterized policy π_θ delivered from an MHE-MPC scheme for a given stage cost $L(\mathbf{y}_k, \mathbf{a}_k)$ as the following total expected cost:

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{y}_k, \mathbf{a}_k) \middle| \mathbf{a}_k = \pi_\theta(\mathbf{s}_k^\star)\right] \tag{60}$$

where the expectation $\mathbb{E}_{\pi_\theta}$ is taken over the distribution of the Markov chain in closed-loop with policy π_θ. The cost $L(\mathbf{y}_k, \mathbf{a}_k)$ reads as a baseline cost (RL stage cost), which is a function of measurable states and actions at the current time k. It is worth noting that the initial conditions are defined by the environment (real MDP), e.g., in the simulation section, the MDPs for test cases 1 and 2 are defined with fixed initial conditions while the third test case has an MDP with random initial conditions. We then seek the optimal policy parameters as follows:

$$\theta_\star = \arg\min_\theta J(\pi_\theta) \tag{61}$$

As a learning-based control approach in this paper, we propose to use a parameterized MPC scheme as a policy approximation in order to deliver π_θ required in a policy gradient method. The MPC-based reinforcement learning then allows us to leverage the capability of MPC in handling the state-input constraints. Although a constraint violation may occur due to an imperfect MPC model, the constraints are finally satisfied by letting RL adjust the whole MPC scheme, e.g., the constraints can be adjusted in the case of constraint violation.

For a given estimated state $\mathbf{s}_k^\star$ obtained from the MHE scheme, the policy delivered by a parameterized MPC scheme is

$$\pi_\theta\left(\mathbf{s}_k^\star\right) = \mathbf{u}_0^\star\left(\mathbf{s}_k^\star, \theta\right) \tag{62}$$

where $\mathbf{u}_0^\star$ is the first element of the control input sequence $\mathbf{u}^\star$ delivered by the following parameterized MPC scheme:

$$\min_{\mathbf{x},\mathbf{u},\sigma} \quad \gamma^{k+N_{\text{MPC}}}\left(T_\theta(\mathbf{x}_{k+N_{\text{MPC}}}) + \mathbf{w}_f^\top \sigma_{k+N_{\text{MPC}}}\right) + \sum_{i=k}^{k+N_{\text{MPC}}-1} \gamma^i\left(l_\theta(\mathbf{x}_i, \mathbf{u}_i) + \mathbf{w}^\top \sigma_i\right) \tag{63a}$$

$$\text{s.t.} \quad \mathbf{x}_{i+1} = \hat{f}_\theta^{\text{MPC}}(\mathbf{x}_i, \mathbf{u}_i), \tag{63b}$$

$$\mathbf{x}_k = \mathbf{s}_k^\star, \tag{63c}$$

$$\mathbf{g}(\mathbf{u}_i) \le 0, \tag{63d}$$

$$\mathbf{h}_\theta(\mathbf{x}_i, \mathbf{u}_i) \le \sigma_i, \quad \mathbf{h}_\theta^f(\mathbf{x}_{k+N_{\text{MPC}}}) \le \sigma_{k+N_{\text{MPC}}} \tag{63e}$$

$$\sigma_{k,\dots,k+N_{\text{MPC}}} \ge 0 \tag{63f}$$

where l_θ and T_θ are the parameterized stage cost and terminal cost, respectively. Note that the imperfect MPC model $\hat{f}^{\text{MPC}}$ is possibly but not necessarily different from the MHE model. We label $\mathbf{h}_\theta$ the mixed constraints, $\mathbf{g}$ the pure input constraints, and $\mathbf{h}_\theta^f$ the terminal constraints. The MPC initial conditions in (63c) are delivered by the MHE scheme at the current time instant k. To relax the inequality constraints, an ℓ_1 relaxation of the mixed constraints (63e) is introduced. An exact penalty is then imposed on the corresponding slack variables σ_k with large enough weights $\mathbf{w}, \mathbf{w}_f$ such that the MPC scheme will not be infeasible under some constraint violations, which appears due to inaccurate MPC model, uncertainties and disturbances.

5. Policy gradient RL with MHE-MPC

In this section, we propose a new observer-based RL framework based on Deterministic Policy Gradient (DPG), MPC, and MHE to deal with the partially observable and imperfect dynamics.

5.1. Compatible deterministic actor-critic

In the context of DPG-based RL algorithms, the policy parameters θ can be directly optimized by the gradient descent step such that the best-expected closed-loop cost (a.k.a policy performance index J) can be captured by applying the policy π_θ. More specifically, the policy parameters θ can be updated as follows:

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\pi_\theta) \tag{64}$$

for some $\alpha > 0$ small enough as the step size. In the context of hybrid controller/observer scheme MHE-MPC, the input signal can be interpreted as a sequence of measurements $\bar{\mathbf{o}}_k = \text{col}\left\{\mathbf{a}_{k-N_{\text{MHE}},\dots,k-1}, \mathbf{y}_{k-N_{\text{MHE}},\dots,k}\right\} \in \mathcal{O}$ at the physical time k. Then, the intermediate variable $\mathbf{s}_k^\star$ is delivered by the MHE scheme based on the history of the measurements and fed to the MPC scheme to deliver the control policy. Let us assume that the measurement history $\bar{\mathbf{o}}_k$ of length $N_{\mathbf{o}}$ is sufficient to determine the statistics of the next output $\mathbf{y}_{k+1}$ such that it remains unaffected for any $\bar{N}_{\mathbf{o}} > N_{\mathbf{o}}$. It follows that $\bar{\mathbf{o}}_k$ is a Markov state. We then consider an input-output MDP based on $\bar{\mathbf{o}}_k$ rather than on the state of the real system, and consequently, this MDP can be described based on the state estimation $\mathbf{s}_k^\star$ as an implicit function of $\bar{\mathbf{o}}_k$. Therefore, the state estimation $\mathbf{s}_k^\star$ also reads as a Markov state, and one can use it in the state (-action) value functions. Let us define the policy performance index by the following expected value:

$$J(\pi_\theta) = \mathbb{E}_{\bar{\mathbf{o}}_k \sim p_k}\left[Q_{\pi_\theta}\left(\mathbf{s}_k^\star, \pi_\theta\left(\mathbf{s}_k^\star\right)\right)\right] = \mathbb{E}_{\bar{\mathbf{o}}_k \sim p_k}\left[V_{\pi_\theta}\left(\mathbf{s}_k^\star\right)\right] \tag{65}$$

where p_k is the measurement distribution at the current physical time k, e.g., a Gaussian distribution. Note that we remove $\bar{\mathbf{o}}_k$ from the arguments of the state (-action) value functions Q_{π_θ} and V_{π_θ} above as $\mathbf{s}_k^\star$ is implicitly constructed (delivered from the MHE scheme (59)) based on the history $\bar{\mathbf{o}}_k$. It also follows that the control policy $\pi_\theta\left(\mathbf{s}_k^\star\right) = \mathbf{u}_0^\star\left(\mathbf{s}_k^\star, \theta\right)$ captured from the MHE-MPC reads as an implicit function of the measurement history. The action-value function Q_{π_θ} is then defined as follows:

$$Q_{\pi_\theta}\left(\mathbf{s}_k^\star, \mathbf{a}_k\right) = L(\mathbf{y}_k, \mathbf{a}_k) + \gamma \mathbb{E}_\zeta\left[V_{\pi_\theta}\left(\mathbf{s}_{k+1}^\star\right) | \mathbf{s}_k^\star, \mathbf{a}_k\right] \tag{66}$$

where the expectation $\mathbb{E}_\zeta$ is taken over the distribution of the Markov chain (2). Based on the proposed DPG theorem by [27] and the fact that both the π_θ and Q_{π_θ} are functions of $\mathbf{s}_k^\star$, the policy gradient equation is described as follows:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \pi_\theta(\mathbf{s}_k^\star) \nabla_{\mathbf{a}_k} Q_{\pi_\theta}(\mathbf{s}_k^\star, \mathbf{a}_k)|_{\mathbf{a}_k = \pi_\theta}\right] \tag{67}$$

where the expectation $\mathbb{E}_{\pi_\theta}$ is taken over the distribution of the Markov chain resulting from the real system in closed-loop with π_θ. To represent the effect of the parameterized MHE upon the policy gradient, the sensitivity of the policy w.r.t θ can be updated such that the new policy gradient is described by the following expectation:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\Xi \nabla_{\mathbf{a}_k} Q_{\pi_\theta}(\mathbf{s}_k^\star, \mathbf{a}_k)|_{\mathbf{a}_k = \pi_\theta}\right] \tag{68}$$

where the Jacobian matrix Ξ is obtained by the following chain rule:

$$\Xi = \nabla_\theta \pi_\theta + \left(\nabla_\theta \mathbf{s}_k^\star + \nabla_\theta \Pi_k \nabla_{\Pi_k} \mathbf{s}_k^\star\right) \nabla_{\mathbf{s}_k^\star} \pi_\theta \tag{69}$$

Hence, the Jacobian matrix above is constructed based on both the MHE and MPC sensitivities where the optimal policy is delivered by a combined MHE-MPC scheme. In this paper, we adopt a *compatible deterministic actor-critic* algorithm [27] in which the action-value function $Q_{\pi_\theta}(\mathbf{s}_k^\star, \mathbf{a}_k)$ can be replaced by a class of compatible

function approximator $Q^{\mathbf{w}}(\mathbf{s}_k^\star, \mathbf{a}_k)$ such that the policy gradient is preserved. Therefore, the compatible function for a deterministic policy π_θ delivered by the parameterized MHE-MPC scheme can be expressed as follows:

$$Q^{\mathbf{w}} = (\mathbf{a}_k - \pi_\theta)^\top \Xi^\top \mathbf{w} + V^\nu(\mathbf{s}_k^\star) \tag{70}$$

The first term in the above compatible function as the critic part is an estimation for the advantage function and the second term estimates a value function for the history of the measurements delivered as a summarized variable $\mathbf{s}_k^\star$ by the MHE scheme. Both functions can be computed by the linear function approximators as follows:

$$V^\nu(\mathbf{s}_k^\star) = \boldsymbol{\Upsilon}(\mathbf{s}_k^\star)^\top \nu, \tag{71a}$$

$$A^{\mathbf{w}}(\mathbf{s}_k^\star, \mathbf{a}_k) = \boldsymbol{\Psi}(\mathbf{s}_k^\star, \mathbf{a}_k)^\top \mathbf{w} \tag{71b}$$

where $\boldsymbol{\Upsilon}(\mathbf{s}_k^\star)$ is the summarized measurement feature vector in order to constitute all monomials of the history of the measurements with degrees less than or equal to 2. The vector $\boldsymbol{\Psi}(\mathbf{s}_k^\star, \mathbf{a}_k) := \Xi(\mathbf{a}_k - \pi_\theta(\mathbf{s}_k^\star))$ includes the state-action features. Considering (70), the policy gradient (68) is then rewritten as follows:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\Xi\,\Xi^\top \mathbf{w}\right] \tag{72}$$

In this paper, the parameterized policy in the context of policy gradient RL is proposed to be captured by the MPC scheme (63). To evaluate the policy gradient (72), one needs to calculate some sensitivities upon the MPC and MHE schemes in order to compute the Jacobian matrix Ξ. Hence, the Jacobian matrices $\nabla_\theta \pi_\theta$ and $\nabla_{\mathbf{s}_k^\star} \pi_\theta$ can be computed by the sensitivity analysis for the parameterized MPC scheme while the gradient $\nabla_\theta \mathbf{s}_k^\star$ and the Jacobian matrices $\nabla_\theta \Pi_k, \nabla_{\Pi_k} \mathbf{s}_k^\star$ are obtained as sensitivity terms for the parameterized MHE scheme.

5.2. Sensitivity analysis and LSTD-based DPG

5.2.1. Sensitivity computation

We describe next how to compute the sensitivities (gradients) needed in the proposed policy gradient RL framework based on MHE-MPC. To that end, let us define the Lagrange functions $\hat{\mathcal{L}}_\theta, \mathcal{L}_\theta$ associated to the MHE and MPC schemes (59), (63) as follows:

$$\hat{\mathcal{L}}_\theta(\hat{z}) = \hat{\Lambda}_\theta + \hat{\lambda}^\top \hat{G}_\theta \tag{73}$$

$$\mathcal{L}_\theta(z) = \Lambda_\theta + \lambda^\top G_\theta + \mu^\top H_\theta \tag{74}$$

where Λ_θ and $\hat{\Lambda}_\theta$ are the total parameterized costs of the MPC and MHE schemes, respectively. The inequality constraints of (63) are collected by H_θ while G_θ and $\hat{G}_\theta$ gather, respectively, the equality constraints in the MPC and MHE schemes. We then label $\lambda, \hat{\lambda}$ the Lagrange multipliers associated with the equality constraints $G_\theta, \hat{G}_\theta$ of the MPC and MHE, respectively. Variables μ are the Lagrange multipliers associated with the inequality constraints of the MPC scheme. Let us label $\boldsymbol{\Gamma} = \{\mathbf{x}, \mathbf{u}, \sigma\}$ and $\hat{\boldsymbol{\Gamma}} = \hat{\mathbf{s}}$ the primal variables for the MPC and MHE, respectively. The associated primal-dual variables then read as $\mathbf{z} = \{\boldsymbol{\Gamma}, \lambda, \mu\}$ and $\hat{\mathbf{z}} = \left\{\hat{\boldsymbol{\Gamma}}, \hat{\lambda}\right\}$.

The sensitivity of the policy delivered by the MPC scheme (63) w.r.t policy parameters and the sensitivity of the estimated state associated with the MHE scheme (59) can be obtained via using the Implicit Function Theorem (IFT) on the Karush Kuhn Tucker (KKT) conditions underlying the parametric NLP. Assuming that Linear Independence Constraint Qualification (LICQ) and Second Order Sufficient Condition (SOSC) hold [20] at $\mathbf{z}^\star$ and $\hat{\mathbf{z}}^\star$, then, the following holds:

$$\frac{\partial \mathbf{z}^\star}{\partial \theta} = -\frac{\partial \kappa_\theta}{\partial \mathbf{z}}^{-1} \frac{\partial \kappa_\theta}{\partial \theta}, \tag{75a}$$

$$\frac{\partial \hat{\mathbf{z}}^\star}{\partial \theta} = -\frac{\partial \hat{\kappa}_\theta}{\partial \hat{\mathbf{z}}}^{-1} \frac{\partial \hat{\kappa}_\theta}{\partial \theta} \tag{75b}$$

where

$$\kappa_\theta = \begin{bmatrix} \nabla_{\boldsymbol{\Gamma}} \mathcal{L}_\theta \\ G_\theta \\ \mathrm{diag}(\mu)\mathbf{H}_\theta \end{bmatrix}, \quad \hat{\kappa}_\theta = \begin{bmatrix} \nabla_{\hat{\boldsymbol{\Gamma}}} \hat{\mathcal{L}}_\theta \\ \hat{G}_\theta \end{bmatrix} \tag{76}$$

are the KKT conditions associated with the MPC and MHE schemes, respectively. As π_θ and $\mathbf{s}_k^\star$ are, respectively, part of $\mathbf{z}^\star$ and $\hat{\mathbf{z}}^\star$. Then, the sensitivity of the MPC policy $\nabla_\theta \pi_\theta$ and the sensitivity of the MHE solution $\nabla_\theta \mathbf{s}_k^\star$ required in (72) can be extracted from gradients $\frac{\partial \mathbf{z}^\star}{\partial \theta}$ and $\frac{\partial \hat{\mathbf{z}}^\star}{\partial \theta}$, respectively.

5.2.2. LSTD-based policy gradient

In the context of compatible DPG, one can evaluate the optimal parameters $\mathbf{w}$ and ν of the action-value function approximation (70) as solutions of the following Least Squares (LS) problem:

$$\min_{\mathbf{w},\nu} \mathbb{E}\left[\left(Q_{\pi_\theta}(\mathbf{s}_k^\star, \mathbf{a}_k) - Q^{\mathbf{w}}(\mathbf{s}_k^\star, \mathbf{a}_k)\right)^2\right], \tag{77}$$

In the context of RL, the Least-Squares Temporal Difference (LSTD) algorithms offer efficient use of data and tend to converge faster than other methods [9]. The LSTD update rules for a policy gradient RL are then obtained as follows:

$$\nu = \Omega_\nu^{-1} b_\nu, \tag{78a}$$

$$\mathbf{w} = \Omega_{\mathbf{w}}^{-1} b_{\mathbf{w}}, \tag{78b}$$

$$\theta \leftarrow \theta - \alpha b_\theta \tag{78c}$$

where the matrices $\Omega_{(\cdot)}$ and the vectors $b_{(\cdot)}$ are calculated by taking expectation ($\mathbb{E}_m$) over m episodes as follows:

$$\Omega_\nu = \mathbb{E}_m\left[\sum_{k=1}^{T_f}\left[\boldsymbol{\Upsilon}(\mathbf{s}_k^\star)\left(\boldsymbol{\Upsilon}(\mathbf{s}_k^\star) - \gamma \boldsymbol{\Upsilon}(\mathbf{s}_{k+1}^\star)\right)^\top\right]\right] \tag{79a}$$

$$\Omega_{\mathbf{w}} = \mathbb{E}_m\left[\sum_{k=1}^{T_f}\left[\boldsymbol{\Psi}(\mathbf{s}_k^\star, \mathbf{a}_k)\boldsymbol{\Psi}(\mathbf{s}_k^\star, \mathbf{a}_k)^\top\right]\right], \tag{79b}$$

$$b_\nu = \mathbb{E}_m\left[\sum_{k=1}^{T_f} \boldsymbol{\Upsilon}(\mathbf{s}_k^\star) L(\mathbf{y}_k, \mathbf{a}_k)\right], \tag{79c}$$

$$b_{\mathbf{w}} = \mathbb{E}_m\left[\sum_{k=1}^{T_f}\left[\left(L(\mathbf{y}_k, \mathbf{a}_k) + \gamma V^\nu(\mathbf{s}_{k+1}^\star) - V^\nu(\mathbf{s}_k^\star)\right)\boldsymbol{\Psi}(\mathbf{s}_k^\star, \mathbf{a}_k)\right]\right], \tag{79d}$$

$$b_\theta = E_m\left[\sum_{k=1}^{T_f} \Xi\,\Xi^\top \mathbf{w}\right] \tag{79e}$$

where T_f is the final time instant at the end of each episode.

6. Simulation results

In this section, we illustrate the performance of the proposed learning-based control and estimation algorithm to deal with three types of problems. In the first test case, we consider a linear system evaluating a model mismatch problem where the MHE model in a combined MHE-MPC scheme is wrong and cannot capture the real system. In the second test case, we show that the proposed

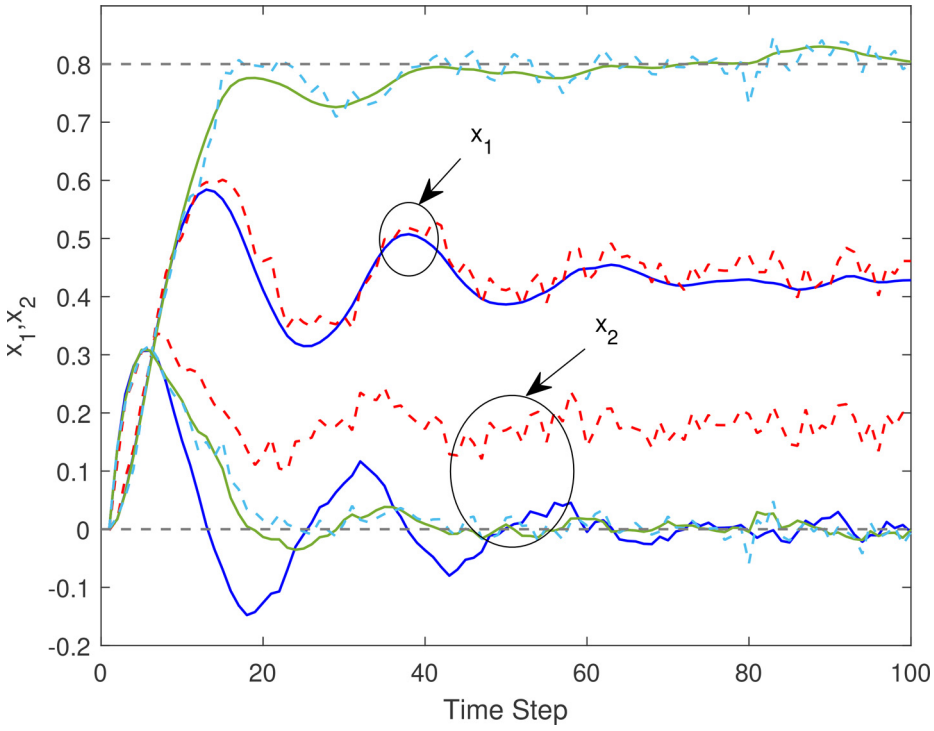

Fig. 1. Real system behavior and state estimations for a set-point tracking ($x_{d_1} = 0.8$ and $x_{d_2} = 0$) in the presence of the model mismatch on the MHE. The solid lines of blue color indicate the states while the estimations are indicated as dashed lines of red color. The correct states and estimations without model mismatch are shown in green.

framework achieves a better closed-loop performance for the control of systems using inaccurate models where a reduced model is used for both the control and estimation goals. We implement our algorithm for a smart building in order to maintain the room temperature in its comfort range even if there is no sufficient knowledge about the building dynamics. Finally, we investigate the proposed learning-based framework applied to a Continuous Stirred Tank Reactor (CSTR) as an example with nonlinear dynamics.

6.1. Test case 1

In this case study, we consider a model mismatch upon the MHE and evaluate a set-point tracking using an MHE-MPC for a two states linear system $\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k$ where x_1 is selected as measurement. The real system and MPC model are chosen as:

$$A = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0312 \\ 0.25 \end{bmatrix}, \tag{80}$$

while the MHE model is selected as:

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} 0.9 & 0.35 \\ 0 & 1.1 \end{bmatrix} \hat{\mathbf{x}}_k + \begin{bmatrix} 0.0813 \\ 0.2 \end{bmatrix} u_k \tag{81}$$

We then use the MHE scheme (59) where the arrival cost is adjusted based on the updating rule (52) and the stage cost is approximated based on the parameterization (54). The input convex NN has two hidden layers with 15 neurons and both the MHE and MPC horizons are set to 8. We use a smooth version of ReLU as an activation function g_j in ICNN (56).

$$g_j(x) = \log\left(1 + \exp(x)\right) \tag{82}$$

Note that in this example only the MHE scheme is learned by RL and the MPC scheme is not parameterized. Fig. 1 shows that the model mismatch on the MHE scheme can affect both the estimation performance and the set-point tracking performance. Indeed, the MHE model mismatch causes a large estimation error on x_2 and the set-point 0.8 on x_1 cannot be tracked. As it is shown in Fig. 2, the mentioned problems due to model mismatch have been solved and a correct state estimation is delivered where the proposed modification of the MHE cost is implemented. Fig. 3 shows the learning progress including the system states x_1, x_2 and their estimations during 60 RL steps such that the closed-loop performance $J(\pi_\theta)$ is improved by the MHE cost modification, and the correct state estimations shown in Fig. 2 are delivered.

6.2. Test case 2

6.2.1. Building model

Let us select a model of the real system of a house floor heating system connected to a ground source-based heat pump shown in Fig. 4. We consider a dynamical model with four states for the building as the real system under control, which is described by a set of ordinary differential equations as follows [24]:

$$C_{wa}\frac{dT_{wa}}{dt} = K_{wa,a}(T_a - T_{wa}) + K_{wa,r}(T_r - T_{wa}) \tag{83a}$$

$$C_r\frac{dT_r}{dt} = K_{wa,r}(T_{wa} - T_r) + K_{f,r}\left(T_f - T_r\right) \tag{83b}$$

$$C_f\frac{dT_f}{dt} = K_{f,r}\left(T_r - T_f\right) + K_b\left(T_w - T_f\right) \tag{83c}$$

$$C_w\frac{dT_w}{dt} = K_b\left(T_f - T_w\right) + \eta W_c \tag{83d}$$

where the control input $u = W_c$ is the power used by the heat pump. The states of the real system $\mathbf{x}^r = \left[T_{wa}, T_r, T_f, T_w\right]^\top$ are labeled the wall temperature, the room temperature, the floor (pavement) temperature, and the water pipeline temperature, respectively. The coefficients C_{wa}, C_r, C_f, and C_w read as the corresponding heat capacities of the above-mentioned temperatures.

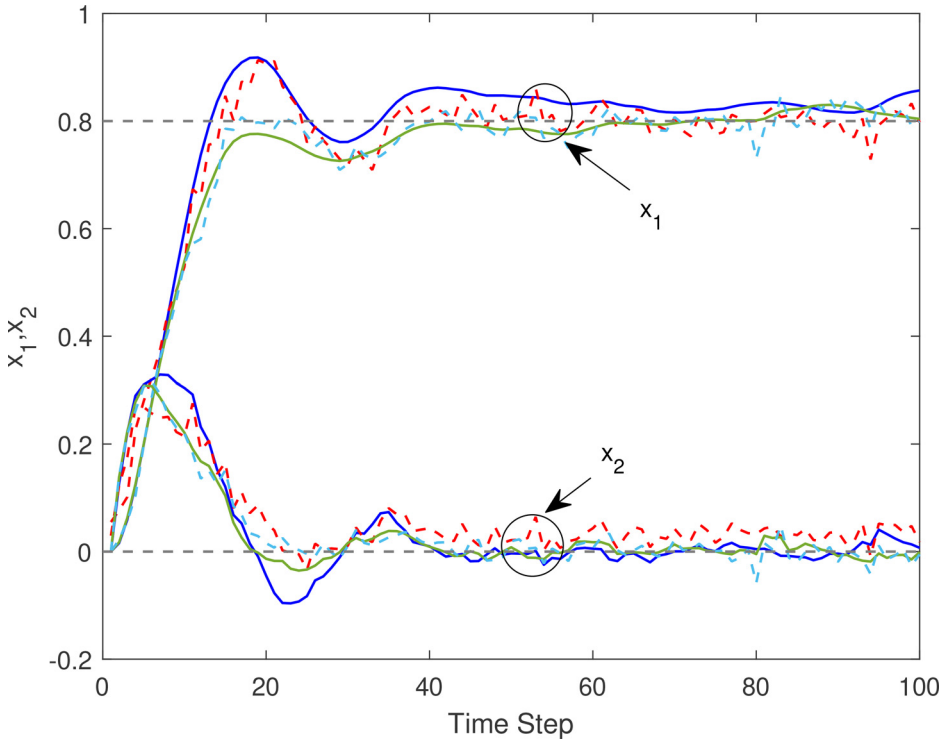

Fig. 2. Real system behavior and state estimations for a set-point tracking ($x_{d_1} = 0.8$ and $x_{d_2} = 0$) where the MHE scheme is modified. The solid lines of blue color indicate the states while the estimations are indicated as dashed lines of red color. The correct states and estimations without model mismatch are shown in green.

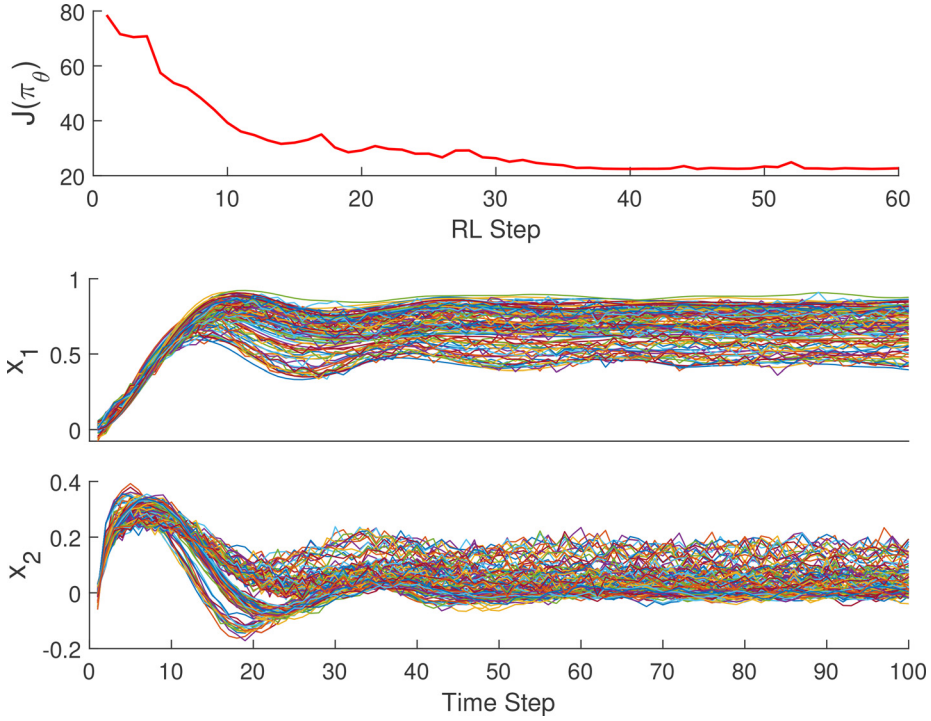

Fig. 3. Closed-loop performance and evolution of states and their estimations during reinforcement learning.

We label $K_{wa,a}$, $K_{wa,r}$, $K_{f,r}$ and K_b the overal heat transfer coefficients between the $\{T_{wa}, T_a\}$ wall-ambient, $\{T_{wa}, T_r\}$ wall-room, $\{T_f, T_r\}$ floor-room and $\{T_f, T_w\}$ floor-water pipeline, respectively.

The Coefficient of Performance (COP) η for heat pumps varies with type, outdoor ground temperature, and condenser temperature. In this paper, we then adopt a stochastic COP shown in Fig. 5 to make the simulations more realistic.

To implement a POMDP scenario, we assume that the building dynamics can be modeled by a reduced model considering the room and water pipeline temperatures (T_r, T_w) as the only measurable states used in the state-space model. Hence, the dynamics of wall inertia and floor are removed from the real system (83), and then a partially observable model with two states is adopted for both the MHE and MPC schemes. To reduce the order of a state-

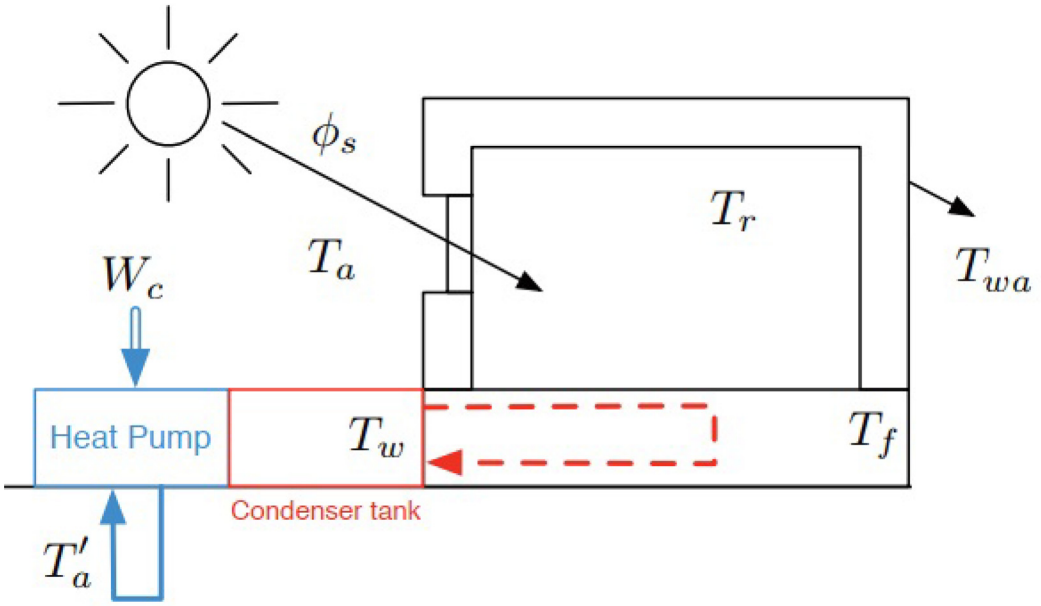

Fig. 4. Building climate control [13] using a heat pump floor heating system. The dashed line represents the floor heating pipelines.

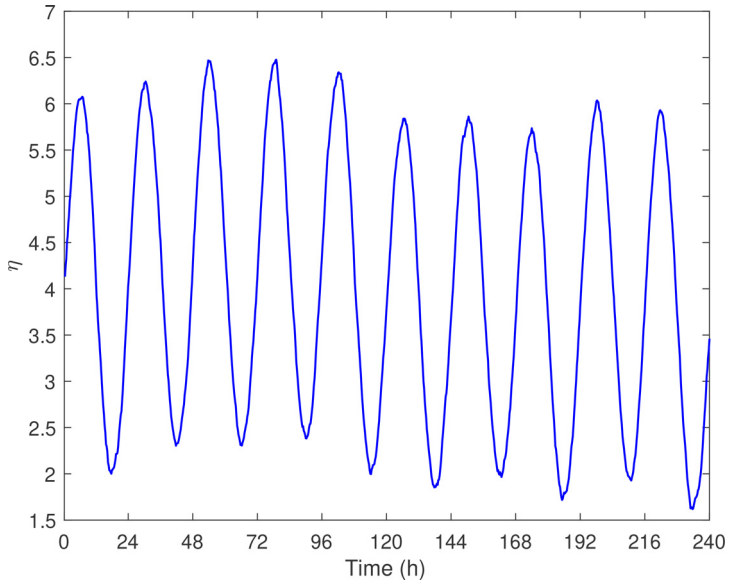

Fig. 5. Stochastic COP of heat pump sampled from the last RL step.

space model captured from the real model of the building (83), we propose to use ÿ*modred*ÿ with option ÿ*MatchDC*ÿ as a built-in function in MATLAB. By eliminating the states T_{wa}, T_f from the real system, the frequency response of the reduced model is affected so that it can no longer follow the real response shown in Fig. 6.

The parameters of the building model adopted in this simulation are given in the following table.

6.2.2. Simulation settings

As we propose to adopt a reduced model of the real system (83) as a POMDP scenario, we label $\mathbf{x}^m = [T_r, T_w]^\top$ the model states (measurements) used in both the MHE and MPC schemes. We then use a parameterized MHE scheme as (59) to estimate the model states from the noisy measurements $\bar{\mathbf{y}} = \mathbf{x}^m$. The stage cost $\hat{L}_\theta\left(\hat{\mathbf{s}}_i, \mathbf{o}_i^H\right)$ in this MHE scheme consists of two cost terms expressed in (54) so that $L_{NN}(\mathbf{Y}_i, \theta_{NN})$ is approximated using an input convex neural network defined in (56). This NN consists of two hidden layers and each layer has 26 neurons where a smooth version of ReLU is used. The cost term L_{θ_0} is selected as a least square problem parameterized as follows:

$$L_{\theta_0} = \left\|\bar{\mathbf{y}}_i - \mathbf{h}\left(\hat{\mathbf{x}}_i^m\right)\right\|_{Q_\theta}^2 + \mathcal{G}_\theta^\top \hat{\mathbf{x}}_i^m \tag{84}$$

Note that the second term in the cost above reads as a gradient modification term. The adjustable weighting matrix Q_θ in the equation above is tuned using RL. As a requirement, this weighting matrix must be symmetric and positive semidefinite. However, the RL steps delivered by the LSTD-based DPG do not necessarily respect this requirement, and we need to enforce it via constraints on the RL steps throughout the learning process. To address this requirement, we then formulate a Semidefinite Programming (SDP) as the following least squares optimization problem:

$$\min_{\Delta\theta} \quad \frac{1}{2}\|\Delta\theta\|^2 - d^\top \Delta\theta \tag{85a}$$

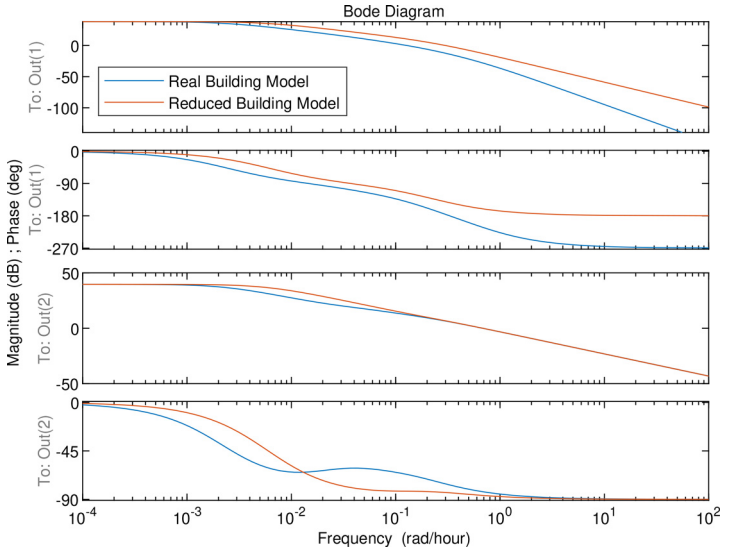

Fig. 6. Bode plot of the frequency response.

$$\text{s.t.} \quad Q_\theta(\theta + \Delta\theta) \succeq 0, \tag{85b}$$

$$W^{(z)}_{1:l-1} \geq 0 \tag{85c}$$

where $\theta = \{Q_\theta, W^{(z)}_{1:l-1}\}$ and $d = -\alpha b_\theta$. We assume that the weighting matrix Q_θ is a linear function of θ. Then, it is updated at every RL step (epoch) due to updating $\Delta\theta$, which is a solution of the above SDP scheme. Note that the second term in the objective function (85a) ensures that $\Delta\theta$ is obtained in the direction of the policy gradient at every RL step.

To keep the room temperature in a comfortable range, we formulate an economic MPC scheme as follows:

$$\min_{\mathbf{x}^m, u, \sigma} \quad \gamma^{k+N_{\text{MPC}}}\left(w_f \sigma_{k+N_{\text{MPC}}}\right) + \sum_{i=k}^{k+N_{\text{MPC}}-1} \gamma^i (p_{u,i} u_i + w\sigma_i) \tag{86a}$$

$$\text{s.t.} \quad \mathbf{x}^m_{i+1} = \hat{f}^{\text{MPC}}(\mathbf{x}^m_i, u_i), \tag{86b}$$

$$\mathbf{x}^m_k = \hat{\mathbf{x}}^{\star,m}_k, \tag{86c}$$

$$\underline{\theta} + T^{\min}_{r,i} - \sigma_i \leq T_{r,i} \leq \bar{\theta} + T^{\max}_{r,i} + \sigma_i, \tag{86d}$$

$$\Delta u_{\min} \leq \Delta u_i \leq \Delta u_{\max}, \tag{86e}$$

$$u_{\min} \leq u_i \leq u_{\max}, \tag{86f}$$

$$\sigma_{k,\dots,k+N_{\text{MPC}}} \geq 0 \tag{86g}$$

where $\hat{\mathbf{x}}^{\star,m}_k$ is the current state estimation delivered by the approximate MHE scheme, p_u is the cost coefficient for the electricity prices, and $\hat{f}^{\text{MPC}}$ is captured from a model reduction approach. To adjust the constraints upon the room temperature, we consider two parameters $(\underline{\theta}, \bar{\theta})$ and let RL tune them. As a result of the theorems developed in this paper, we propose to modify the stage cost of the MHE scheme with a reduced model to tackle POMDPs. To that end, we let RL adjust the NN weights θ_{NN} and some parameters of the first stage cost term L_{θ_0} including inverse of the covariance matrix Q_θ and gradient term $\mathcal{G}_\theta$ in (84). Hence, all RL parameters $\theta = \{\theta_{NN}, Q_\theta, \mathcal{G}_\theta, \underline{\theta}, \bar{\theta}\}$ are adjusted by the proposed LSTD-based DPG reinforcement learning. We adopt a baseline stage cost in the proposed LSTD-based RL algorithm as follows:

$$L(y_k, a_k) = p_{u,k}.a_k + w.\max\left(0, h\left(T_{r,k}\right)\right) \tag{87}$$

where $a_k = \pi_\theta\left(\hat{\mathbf{x}}^{\star,m}_k\right) = \mathbf{u}^\star_0\left(\hat{\mathbf{x}}^{\star,m}_k, \theta\right)$ with the possible addition of occasional random exploratory moves. Note that $\mathbf{u}^\star_0$ is the first element of the control input sequence $\mathbf{u}^\star$ delivered by the MPC scheme (86). We use the weight $w = 100$ where $h\left(T_{r,k}\right)$ collects the inequality constraints upon indoor temperature $T^{\min}_{r,k} \leq T_{r,k} \leq T^{\max}_{r,k}$.

We choose a sampling time $15min$ and a forecast $24h$ for the ambient disturbances and electricity prices. Therefore, the prediction and estimation horizons $(N_{\text{MPC}}, N_{\text{MHE}})$ are set to 96. The ambient temperature and electricity prices are forecasted for 10 days starting from the first day of January 2021 in Trondheim, Norway where the data used in this simulation is provided by Nord Pool Spot as an electricity market operator.

6.2.3. Discussion

In practice, it is very difficult to make an accurate model of a building for the model-based control approaches, i.e., an MPC scheme since there are some complex dynamics and uncertainties that may not be captured. To address this complexity, a common solution is to adopt some simplified and reduced models in this context. Although these simplified models are useful to be used in an MPC scheme in order to reduce computational complexity, they can affect the control performance in a building climate control system. In this simulation, we use a super-simplified and realistic building model where its dynamics include only two measurable states T_r, T_w while the aim is to control the indoor temperature in a real model (83). As it is shown in Fig. 7, the first evolution (No learning is used) of T_r in blue color cannot perfectly respect the lower variable constraint and there is a heavy violation since the model is not truly captured.

The evolution of estimated T_r is depicted in Fig. 8 in red color and it can be observed that the first evolution of this estimation is not able to follow the first evolution of the real T_r in blue color. This estimation error, where there is still no adopted learning mechanism upon MHE and MPC, can be clearly observed in Fig. 10. To address these problems induced by adopting a reduced

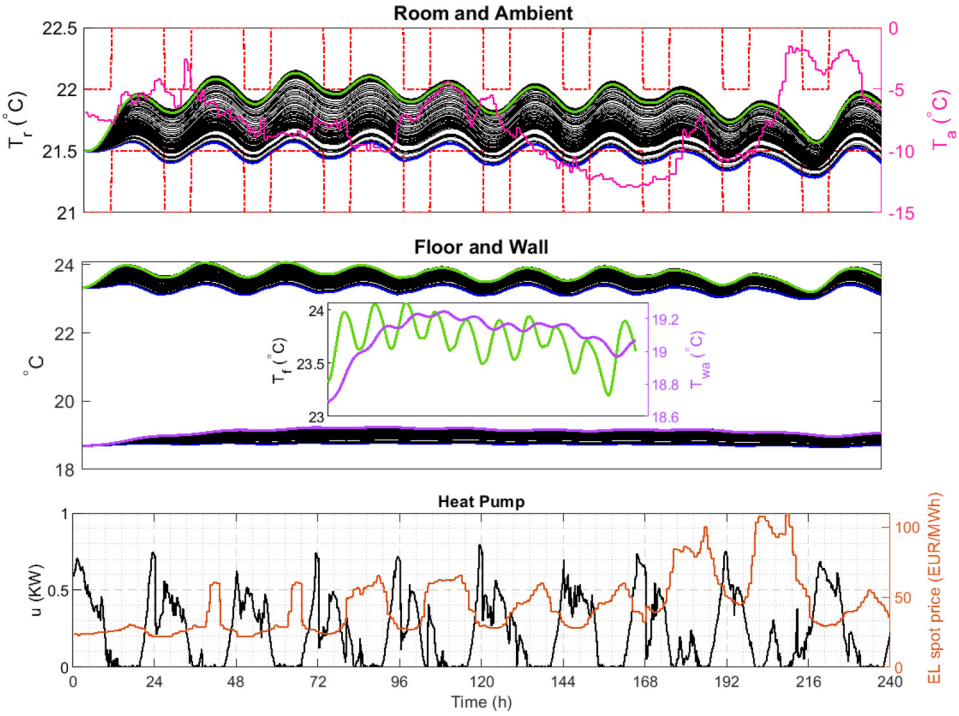

Fig. 7. Evolution of the building temperatures T_r, T_f, T_{wa} (black color) and trained optimal policy u where both the estimator (MHE) and controller (MPC) use an imperfect model. The comfort T_r is captured (green color) after 185 learning steps (epoch) for the adjustment of MHE and MPC schemes.

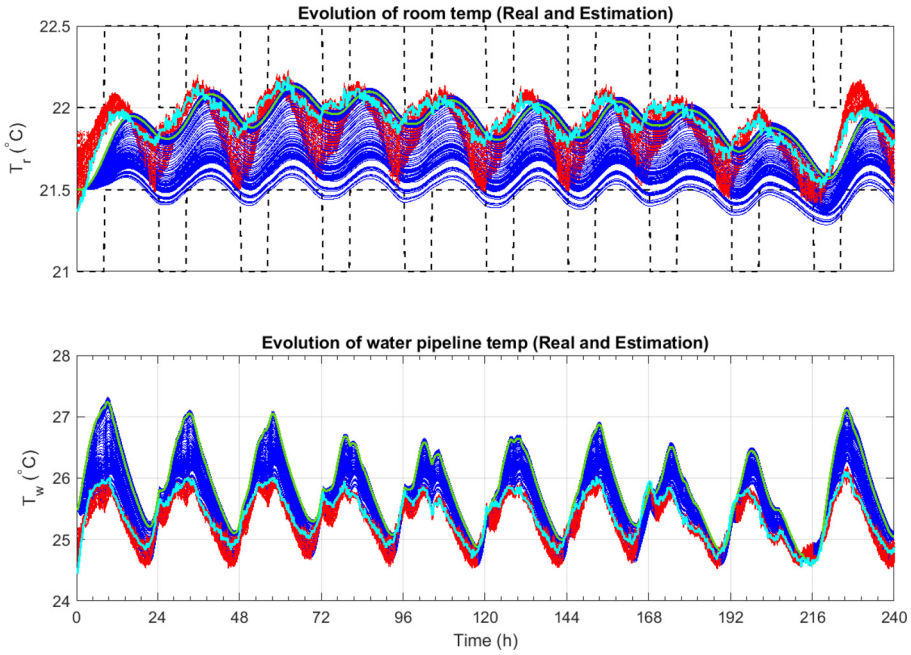

Fig. 8. Evolution of the real states as measurements (temperatures T_r, T_w in blue color) and their estimations (red color) used in the inaccurate models of MHE and MPC as POMDPs. The estimations in light blue color are captured from the trained MHE estimator.

model, we let an LSTD-based DPG reinforcement learning adjust the parameters of both the MHE (cost modification) and MPC (constraint adjustment) schemes shown in Fig. 9 in order to capture a correct state estimation and deliver a learned policy to tackle this model inaccuracy.

To conclude the proposed learning-based state estimation and control, we can observe that the proposed theorem of cost modification in an MHE scheme with imperfect model works since we achieve a perfect closed-loop performance by applying that theorem in order to modify the MHE cost depicted in Fig. 9. It is worth noting that, the learned policy is optimally captured from the MPC scheme so that the heat pump power has its highest value in lower electricity prices and it has a minimum peak for times that the electricity is expensive shown in Fig. 7.

6.3. Test case 3

6.3.1. CSTR Nonlinear model

In this section, the proposed learning-based MHE-MPC scheme is applied to a Continuous Stirred Tank Reactor (CSTR), where the

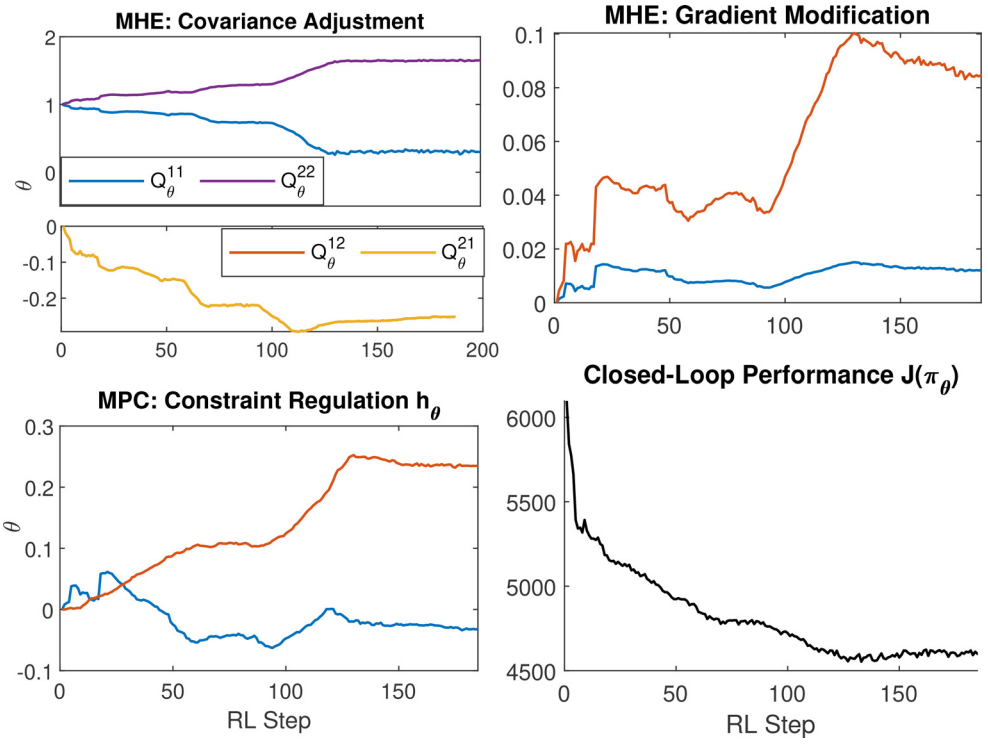

Fig. 9. Some parameters of the MHE/MPC and the closed-loop performance $J(\pi_\theta)$ over reinforcement learning steps.

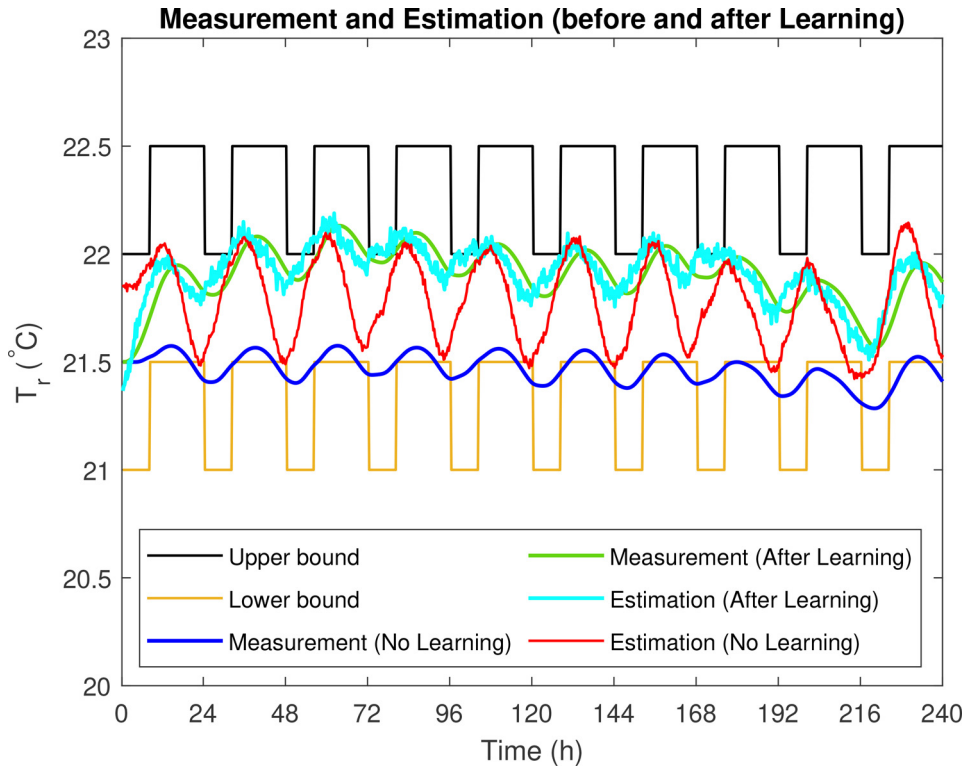

Fig. 10. The building indoor temperature before and after learning the estimator and controller.

dynamical system is nonlinear and may not be modeled accurately. In this chemical reactor, the reaction $(A \to B)$ is accomplished by means of an irreversible and exothermic chemical reaction, and the aim is to control the concentration of A, C_a, and the reaction volume, V, by manipulating the output process flow rate, q_s, and the coolant flow rate, q_c, see [22]. The CSTR dynamics are described as follows: (Table 1)

$$\dot{V}(t) = q_o - q_s(t), \tag{88a}$$

$$\dot{C}_a(t) = \frac{q_o}{V(t)}(C_{a_o} - C_a(t)) - k_0 e^{\frac{-E}{RT(t)}} C_a(t), \tag{88b}$$

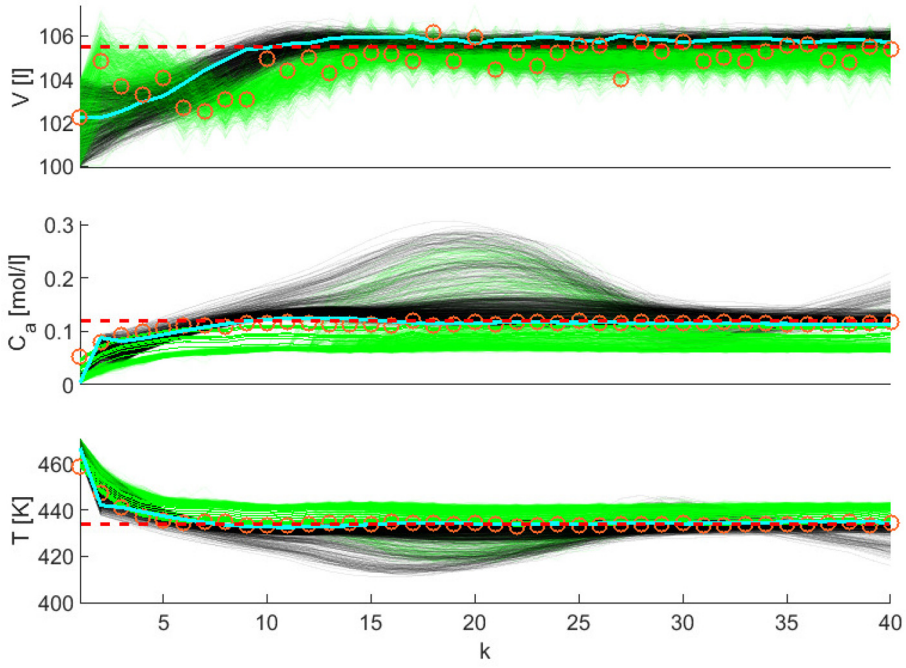

Fig. 11. Evolution of the CSTR states and their estimations during the learning progress. The system states and their estimations at the learning stage are shown as black and green lines, respectively. The orange circles and blue lines, respectively, represent the correct state estimations and the system states delivered after 800 RL steps. The set points are shown as red dashed lines.

Table 1
Building parameters.

C_{wa}	$24.2 \times 10^6[\frac{J}{K}]$	$K_{wa,a}$	$56[\frac{W}{K}]$
C_r	$6 \times 10^6[\frac{J}{K}]$	$K_{wa,r}$	$386[\frac{W}{K}]$
C_f	$24.8 \times 10^6[\frac{J}{K}]$	$K_{f,r}$	$594[\frac{W}{K}]$
C_w	$20.7 \times 10^6[\frac{J}{K}]$	K_b	$506[\frac{W}{K}]$

$$\dot{T}(t) = \frac{q_o}{V(t)}(T_o - T(t)) + k_1 e^{\frac{-E}{RT(t)}} C_a(t) + k_2 \frac{q_c(t)}{V(t)}\left(1 - e^{\frac{-k_3}{q_c(t)}}\right)(T_{co} - T(t)), \tag{88c}$$

$$k_1 = \frac{-\Delta H k_0}{\rho C_p}, \quad k_2 = \frac{\rho_c C_{pc}}{\rho C_p}, \quad k_3 = \frac{hA}{\rho_c C_{pc}} \tag{88d}$$

where $V(t), C_a(t), T(t)$ are the reaction volume, the concentration of A, and the reactor temperature, respectively. We consider $V(t)$ and $T(t)$ as measurable states since it is not usually easy to measure the concentration C_a directly. The measurement noises are selected as $\mathcal{N}(0, Q)$ with $Q = \text{diag}(2.5^2, 2.5^2)$. The control inputs are q_s and q_c. The constant parameters given in Table 2 are the process flow rate q_o, the feed concentration C_{a_o}, the reaction rate k_0, the activation energy term E/R, the feed temperature T_o, the inlet coolant temperature T_{co}, the heat of reaction ΔH, the heat transfer term hA, the liquid densities ρ, ρ_c and the specific heats C_p, C_{pc}.

To investigate the performance of the proposed modification of the MHE scheme, we adopt the correct model (88) in the MPC scheme while the MHE scheme is formulated using an imperfect model of the real system as follows:

Table 2
CSTR Model Parameters.

q_o	$100[\frac{l}{min}]$	C_{a_o}	$1[\frac{mol}{l}]$
T_o	$350[K]$	T_{co}	$350[K]$
ΔH	$-2 \times 10^5[\frac{cal}{mol}]$	ρC_p	$1000[\frac{cal}{lK}]$
k_0	$7.2 \times 10^{10}[\frac{1}{min}]$	E/R	$1 \times 10^4[K]$
$\rho_c C_{pc}$	$1000[\frac{cal}{lK}]$	hA	$7 \times 10^5[\frac{cal}{minK}]$

$$\dot{V}(t) = q_o - q_s(t), \tag{89a}$$

$$\dot{C}_a(t) = 0.93\frac{q_o}{V(t)}(C_{a_o} - C_a(t)) - 1.2k_0 e^{\frac{-E}{RT(t)}} C_a(t), \tag{89b}$$

$$\dot{T}(t) = 0.93\frac{q_o}{V(t)}(T_o - T(t)) + 1.3k_1 e^{\frac{-E}{RT(t)}} C_a(t) + 0.8k_2 \frac{q_c(t)}{V(t)}\left(1 - e^{\frac{-0.8k_3}{q_c(t)}}\right)(T_{co} - T(t)) \tag{89c}$$

The constraints on the states and control inputs are $90 \le V \le 110$, $0 \le C_a \le 0.35$, $400 \le T \le 480$, $80 \le q_s \le 120$ and $75 \le q_c \le 140$.

6.3.2. Simulation settings

In this simulation, both the modification step H and the horizon N are set to 10. The number of neurons in the hidden layers of the ICNN is set to 18, and we consider a sampling time 0.1min. In the reinforcement learning setting, the parameterized MHE scheme is adjusted during 800 episodes (RL steps), where each episode includes a 4min (40 time steps) of running the real system. Hence, in this simulation scenario, we use a total of 3.2×10^4 learning samples to modify the MHE scheme with the imperfect model (89), and improve the closed-loop performance. Note that the initial conditions are randomly selected for episodes of length 40. The set-point tracking goal is set as $V^d = 105\ l$, $C_a^d = 0.12\ mol/l$, $T^d = 433.72\ K$, $q_s^d = 100\ l/min$ and $q_c^d = 110\ l/min$.

6.3.3. Discussion

Fig. 11 depicts the evolution of the system states and their estimation during the learning progress. As it is observed, the correct state estimations (orange circles) are delivered after 800 RL steps so that the system states (blue lines) can track the corresponding set points accurately.

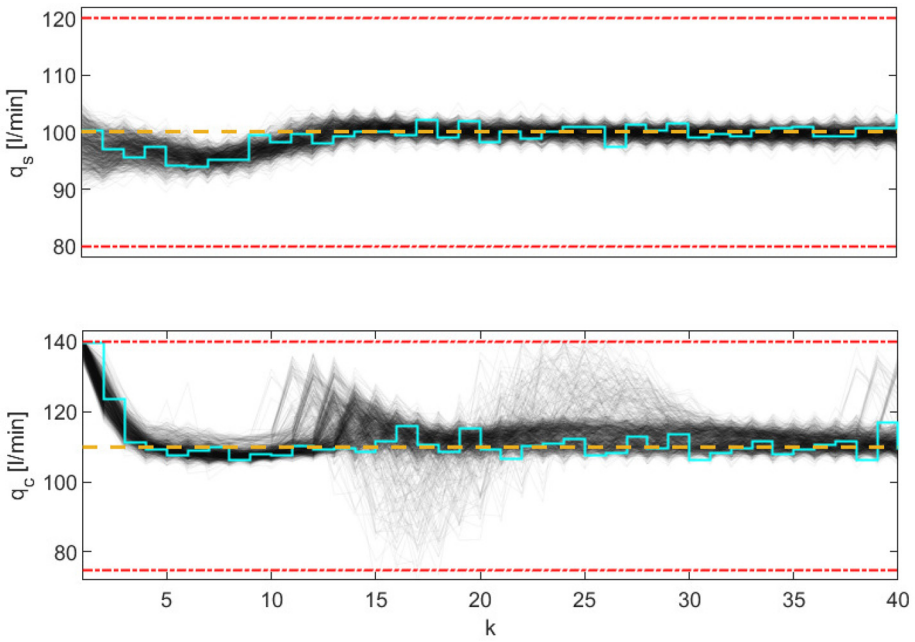

Fig. 12. Evolution of the CSTR control inputs during the learning progress. The control inputs at the learning stage are shown as black lines while the optimal control inputs delivered from the MHE-MPC after 800 RL steps are shown as blue stairs. The constraints and set points are shown as red-dashed and yellow-dashed lines, respectively.

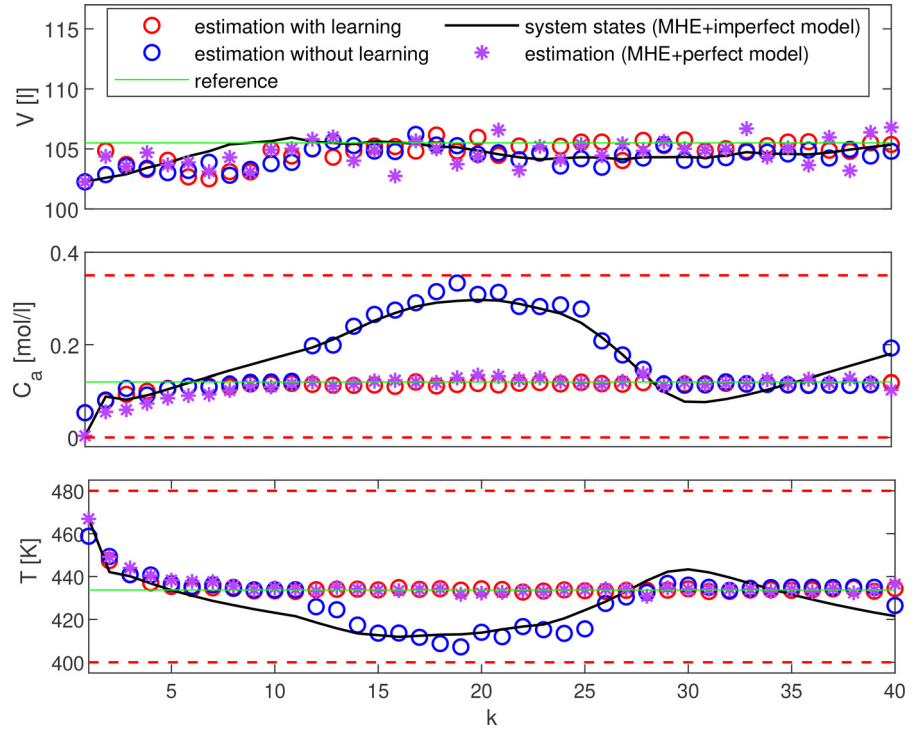

Fig. 13. Comparative analysis between the conventional MHE-MPC with an imperfect model and learning-based MHE-MPC.

Fig. 12 shows the evolution of the control inputs in black color during the learning progress, and it is observed that the optimal control inputs shown as stairs in blue color track the corresponding set-points while the constraints are guaranteed.

The results depicted in Fig. 13 provide a comparative analysis between the proposed learning-based MHE-MPC and one without learning. It can be observed that the system states in black color are struggling to track the references since the imperfect model (89) is used in the MHE scheme, and the wrong estimation is delivered, shown as blue circles. The proposed learning-based modification of the MHE scheme then adjusts the MHE stage cost function so that the state estimations (red circles) perfectly match the correct estimations. Note that the correct estimations are those captured from the MHE scheme with the perfect model (88).

The closed-loop performance $J(\pi_\theta)$ is illustrated in Fig. 14, and it is observed that the best performance is achieved after 500 episodes, and the norm of policy gradient steps $\nabla_\theta J$ moves towards zero since the policy parameters converge to the optimal parameters.

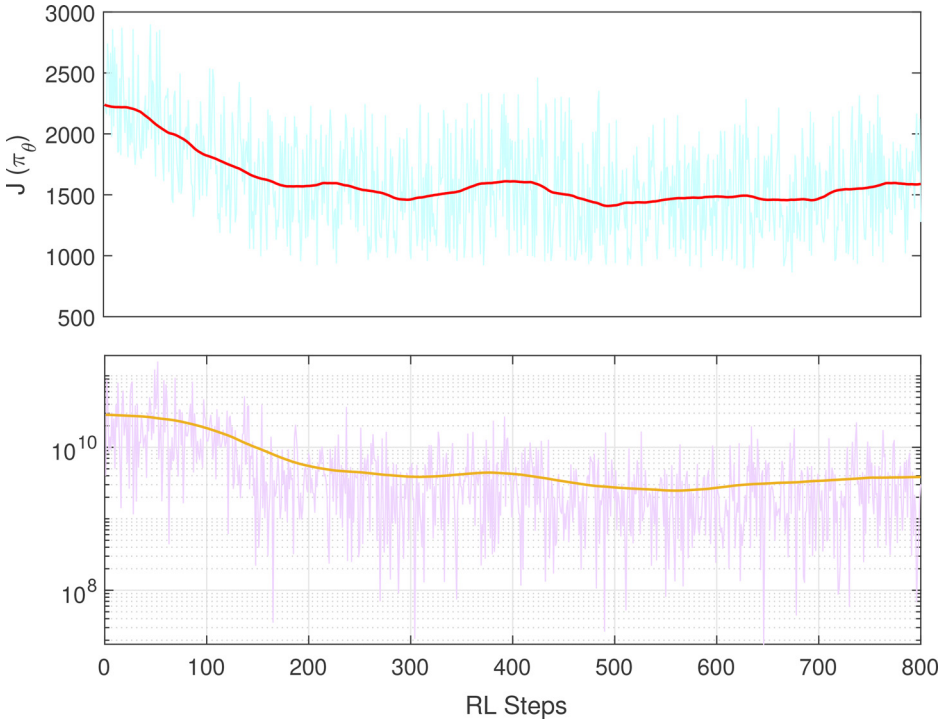

Fig. 14. Closed-loop performance and the norm of the policy gradient steps $||\nabla_\theta J(\pi_\theta)||_2$ in logarithmic scale. The noisy closed-loop performance is shown as a blue line while its moving average is shown as a red line. The noisy logarithmic scale of $||\nabla_\theta J(\pi_\theta)||_2$ is shown as a purple line while its moving average is shown as a yellow line.

Table 3
Computation time.

Time	Baseline MHE-MPC	Modified MHE-MPC
Test Case 1	0.89 sec	1.68 sec
Test Case 2	10.2 min	19.47 min
Test Case 3	1.23 sec	2.15 sec

6.4. Computation time

To investigate the computation time of solving the proposed modified MHE-MPC scheme, the choice of the modification step H and the number of neuron used in the hidden layers of ICNN in the modified MHE scheme are regarded as crucial issues since they determine the number of the parameters required in the proposed modification. Although, the horizon H could be larger than N to approximate the modified stage cost accurately, we may choose a small size of H even smaller than N in order to provide an acceptable trade-off between the computational effort and the approximate value captured from the NN. To mitigate the computational efforts for all three numerical examples, the horizon length of the modification step H has been set to the same value as the prediction/estimation horizon N. Fortunately, the training stage of the proposed MHE/MPC-based RL can be accomplished offline, and it is worth mentioning that the MPC parameterization is quite flexible so that the MPC cost can be parameterized from a numerical perspective that makes the MPC implementation as tractable and effective as possible. However, the MHE/MPC-based RL combined with NN in the loop may struggle a bit in the real-time applications, in particular those cases with very small sampling times and large estimation/prediction horizons. Nonetheless, the progress in the optimization algorithms and in the computational hardware makes the deployment of real-time MHE/MPC possible for most of the real applications. The computation times for three test cases above are provided in the Table 3. Note that we do not use real-time solvers in the present paper.

7. Conclusion

In this paper, we have shown how an MHE scheme can be modified such that its performance degradation due to using an imperfect MHE model is tackled. The stage cost modification in both versions of the stochastic and deterministic MHE schemes is proposed so that a correct probability measure and state estimation can be delivered even if the underlying model cannot capture the real system. A practical implementation of the proposed approach upon the MHE cost modification is discussed. To achieve the best closed-loop performance for a combined MHE-MPC scheme using an imperfect model of the real system, we detail a parameterization method for both the deterministic MHE and MPC schemes and develop an MHE/MPC-based policy gradient reinforcement learning algorithm. The effectiveness of the proposed learning-based estimator/controller has been established for three examples including a model mismatch problem, a climate control of smart building where the building model used in the MHE-MPC is simplified and different from the real dynamics, and a CSTR estimation/control problem. Further work will aim to implement a modified stochastic MHE scheme proposed in this paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was funded by the Research Council of Norway (RCN) project Safe Reinforcement Learning using MPC (SARLEM).

References

[1] B. Amos, L. Xu, J. Z. Kolter, Input convex neural networks, 2017. 1609.07152

[2] K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath, Deep reinforcement learning: a brief survey, IEEE Signal Process. Mag. 34 (6) (2017) 26–38.

[3] J. Berberich, J. Köhler, M.A. Müller, F. Allgöwer, Data-driven model predictive control: closed-loop guarantees and experimental results, at - Automatisierungstechnik 69 (7) (2021) 608–618.

[4] S. Bezuglyi, P.E.T. Jorgensen, Transfer Operators, Endomorphisms, and Measurable Partitions, Springer International Publishing, 2018, pp. 105–111.

[5] F. Bünning, A. Schalbetter, A. Aboudonia, M.H. de Badyn, P. Heer, J. Lygeros, Input convex neural networks for building MPC, 2020, arXiv:2011.13227

[6] W. Cai, H.N. Esfahani, A.B. Kordabad, S. Gros, Optimal management of the peak power penalty for smart grids using MPC-based reinforcement learning, in: Proceeding of the 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021a, pp. 6365–6370.

[7] W. Cai, A.B. Kordabad, H.N. Esfahani, A.M. Lekkas, S. Gros, MPC-based reinforcement learning for a simplified freight mission of autonomous surface vehicles, in: Proceeding of the 60th IEEE Conference on Decision and Control (CDC), 2021b, pp. 2990–2995, doi:10.1109/CDC45484.2021.9683750.

[8] H.N. Esfahani, S. Gros, Policy gradient reinforcement learning for uncertain polytopic LPV systems based on MHE-MPC, IFAC-PapersOnLine 55 (15) (2022) 1–6. 6th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2022

[9] H.N. Esfahani, A.B. Kordabad, S. Gros, Approximate robust NMPC using reinforcement learning, in: Proceeding of the European Control Conference (ECC), 2021a, pp. 132–137, doi:10.23919/ECC54610.2021.9655129.

[10] H.N. Esfahani, A.B. Kordabad, S. Gros, Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics, in: Proceeding of the American Control Conference (ACC), 2021b, pp. 2121–2126.

[11] S. Gros, M. Zanon, Data-driven economic NMPC using reinforcement learning, IEEE Trans. Autom. Control 65 (2) (2020) 636–648.

[12] Z.D. Guo, M.G. Azar, B. Piot, B.A. Pires, T. Pohlen, R. Munos, Neural predictive belief representations, CoRR abs/1811.06407 (2018).

[13] R. Halvgaard, N.K. Poulsen, H. Madsen, J.B. Jørgensen, Economic model predictive control for building climate control in a smart grid, in: Proceeding of the IEEE PES Innovative Smart Grid Technologies (ISGT), 2012, pp. 1–6.

[14] M.J. Hausknecht, P. Stone, Deep recurrent Q-learning for partially observable MDPs, CoRR abs/1507.06527 (2015).

[15] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artific. Intell. 101 (1) (1998) 99–134.

[16] B. Karg, S. Lucia, Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning, Comput. Chem. Eng. 148 (2021) 107266.

[17] A.B. Kordabad, H.N. Esfahani, A.M. Lekkas, S. Gros, Reinforcement learning based on scenario-tree MPC for ASVs, in: Proceeding of the American Control Conference (ACC), 2021, pp. 1985–1990.

[18] S. Muntwiler, K.P. Wabersich, M.N. Zeilinger, Learning-based moving horizon estimation through differentiable convex optimization layers, 2021, arXiv:2109.03962

[19] X. Nian, A.A. Irissappane, D. Roijers, Dcrac: Deep conditioned recurrent actor–critic for multi-objective partially observable environments, in: Proceeding of the International Foundation for Autonomous Agents and Multiagent Systems, in: AAMAS '20, 2020, pp. 931–938.

[20] J. Nocedal, S. Wright, Numerical Optimization, 2 ed., Springer, 2006.

[21] P. Kuhl, M. Diehl, T. Kraus, J.P. Schloder, H.G. Bock, A real-time algorithm for moving horizon state and parameter estimation, Comput. Chem. Eng. 35 (1) (2011) 71–83.

[22] H.A. Pipino, C.A. Cappelletti, E.J. Adam, Adaptive multi-model predictive control applied to continuous stirred tank reactor, Comput. Chem. Eng. 145 (2021) 107195.

[23] C.V. Rao, J.B. Rawlings, Constrained process monitoring: moving-horizon approach, AIChE J. 48 (1) (2002) 97–109, doi:10.1002/aic.690480111.

[24] S. Rastegarpour, L. Ferrarini, S. Gros, Economic NMPC for multiple buildings connected to a heat pump and thermal and electrical storages, IFAC-PapersOnLine 53 (2) (2020) 17089–17094. 21st IFAC World Congress

[25] J.B. Rawlings, L. Ji, Optimization-based state estimation: current status and some new results, J. Process Control 22 (8) (2012) 1439–1444.

[26] J.B. Rawlings, D.Q. Mayne, M. Diehl, Model Predictive Control: Theory, Computation, and Design, 2, Nob Hill Publishing Madison, WI, 2017.

[27] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: Proceedings of the 31st International Conference on International Conference on Machine Learning, JMLR.org, 2014. I–387–I–395

[28] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[29] T. Gangwani, J. Lehman, Q. Liu, J. Peng, Learning belief representations for imitation learning in POMDPs, in: Proceeding of the 35th Conference on Uncertainty in Artificial Intelligence, 2019, pp. 1–14.

[30] M. Tenny, J. Rawlings, Efficient moving horizon estimation and nonlinear model predictive control, in: Proceedings of the American Control Conference (IEEE Cat. No.CH37301), 6, 2002, pp. 4475–4480 vol.6, doi:10.1109/ACC.2002.1025355.

[31] B. Wang, Z. Ma, S. Lai, L. Zhao, T.H. Lee, Differentiable moving horizon estimation for robust flight control, 2021, arXiv:2108.03212

[32] X. Xiang, S. Foo, Recent advances in deep reinforcement learning applications for solving partially observable Markov decision processes (POMDp) problems: part fundamentals and applications in games, robotics and natural language processing, Mach. Learn. Knowl. Extract. 3 (3) (2021) 554–581.

[33] Y. Wang, K. Velswamy, B. Huang, A novel approach to feedback control with deep reinforcement learning, IFAC-PapersOnLine 51 (18) (2018) 31–36. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018

[34] A.S. Zamzam, X. Fu, N.D. Sidiropoulos, Data-driven learning-based optimization for distribution system state estimation, IEEE Trans. Power Syst. 34 (6) (2019) 4796–4805.

[35] M. Zanon, S. Gros, Safe reinforcement learning using robust MPC, IEEE Trans. Autom. Control 66 (8) (2021) 3638–3652.

[36] X. Zhong, Z. Ni, Y. Tang, H. He, Data-driven partially observable dynamic processes using adaptive dynamic programming, in: Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2014, pp. 1–8.