12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Data-predictive Control of Multi-Timescale Nonlinear Processes ⋆

Jun Wen Tang * **Yitao Yan** * **Jie Bao** * **Biao Huang** **

* *School of Chemical Engineering, UNSW Sydney, NSW 2052, Australia. (e-mail: jun.tang@unsw.edu.au; y.yan@unsw.edu.au; j.bao@unsw.edu.au).*
** *Department of Chemical and Materials Engineering, University of Alberta, 116 St. and 85 Ave., Edmonton, AB, Canada T6G 2R3. (e-mail: biao.huang@ualberta.ca)*

Abstract: A novel big data-predictive control approach for nonlinear multi-timescale processes is presented in this paper. Multiple Dynamical Latent Variable Autoencoders (DLVAEs) are employed to approximate multi-timescale dynamics, utilizing timescale-based low-pass filtering and resampling of historical input-output data. The encoder in each DLVAE projects the nonlinear physical variable space onto a linear latent variable space, represented by a kernel space in behavioral system theory. During training, we not only impose kernel spaces and reconstruct data but also establish connections among latent variables from different DLVAEs at matching time-steps. Collectively, these multi-level latent variables span a wide prediction time horizon with limited (non-uniformly spaced) steps encompassing the current, near, and distant future. In online tracking control, we guide the latent variables from each DLVAE to their respective setpoints (derived from physical variable setpoints) while maintaining consistent physical variable values at matching time-steps, all within a linear framework.

Keywords: Data-based control, nonlinear processes, multi-timescale dynamics, behavioral systems theory, autoencoder

1. INTRODUCTION

Multi-timescale dynamics properties are common in various chemical processes due to the coexisting mechanisms of different timescales (e.g., Lévine and Rouchon (1991)). Additionally, the dynamics of boundary conditions in most distributed parameter systems typically span a wide range of time constants. For instance, industrial thickener processes, governed by consolidation, convection, and sedimentation, exhibit dynamics with a continuum of time constants ranging from seconds to 10 hours (Tan et al., 2016). A similar scenario can be observed in fuel cells and flow batteries with time constants ranging from milliseconds to seconds (Radisavljević-Gajić et al., 2019). To design control system for multi-timescale processes, it is imperative to consider the dynamics of different timescales to prevent deterioration in control performance and the process instability (Christofides and Daoutidis, 1996).

In model-based control, the focus often revolves around processes with explicitly separable dual timescales, where methods employ various techniques, including singular perturbation theory (Christofides and Daoutidis, 1996; Kumar et al., 1998) and, more recently, neural networks (Jian et al., 2023). However, these approaches can yield intricate nonlinear models that are prone to issues such as convergence problems and numerical stiffness (Ma et al., 2018). To address these challenges, one approach is to mitigate ill-conditioning by partitioning models into reduced-order subsystems and designing multi-stage controllers (e.g., Ellis et al. (2013)). Additionally, model predictive control (MPC) encounters difficulties with multi-timescale dynamics, necessitating extended prediction horizons to accommodate slow dynamics, which can lead to computational expenses. To tackle this issue, strategies like hierarchical MPC (Zhang et al., 2022) and MPC with non-uniform optimization horizons (Tan et al., 2016) have been proposed. It's important to note that these model-based control approaches require dynamic process models, which can be costly to develop and may introduce biases and modeling errors due to predetermined constraints (Huang and Kadali, 2008).

With the widespread use of sensors and control systems, modern chemical processes generate large volumes of data (e.g., Huang and Kadali (2008)). Data-based control has gained attention in process industries due to its direct use of process data for control, contrasting model-based approaches like MPC (Coulson et al., 2019; Markovsky and Rapisarda, 2008). Data-based control is particularly advantageous when process models are challenging to develop due to complex dynamics or a lack of process understanding. Data Predictive Control (DPC) utilizes process data directly, employing Hankel matrices (Coulson et al., 2019; Wei et al., 2020) or equivalently a kernel space (Zhao et al., 2022), grounded in the system behavioral framework (Willems, 1986a,b). A novel control method based on the behavioral systems framework, employing distributed big

⋆ This work was supported in part by the Australian Research Council under Grant DP220100355. *(Corresponding author: Jie Bao)*

data analysis, was recently introduced by Yan et al. (2023). However, designing data-based control for multi-timescale processes is challenging due to unknown timescales, similar to the system identification problem (Yang and Zhu, 2021). Noise and quantization errors make capturing slow dynamics within short horizons difficult, requiring, long trajectories that makes DPC computationally complex. Furthermore, extended prediction horizons may lead to error accumulation. Recently, a big data-driven control approach was developed by the authors for linear multi-timescale processes Tang et al. (2022, 2023) to use a multi-level DPC with each level capturing dynamics of different timescales using resampling of filtered process data. Notably, addressing data-based control for nonlinear multi-timescale processes remains an open challenge.

In this work, we propose a novel big data-predictive control method for nonlinear multi-timescale processes using Dynamic Latent Variable Autoencoders (DLVAEs) (similar to Zhao et al. (2022)) in a system behavioral framework. DLVAEs approximate multi-timescale dynamics by employing timescale-based filtering and resampling of historical input-output data. Using the encoder of each DLVAE, the physical variable nonlinear space is projected onto a linear latent variable space that is described by a kernel representation in the behavioral system theory. During training, we discover kernel spaces, reconstruct data, and establish connections among latent variables from different DLVAEs at corresponding time-steps. The latent variables span the prediction horizon with non-uniformly spaced steps, encompassing current, near, and distant future values. In online tracking control, we guide latent variables from DLVAEs to their respective setpoints, ensuring consistent (equivalent) physical variable values at matching time-steps, all within a linear framework.

The structure of the paper is as follows. Section 2 outlines the preliminaries regarding behavioral systems theory and DLVAE. Section 3 showcases the control structure of DPC with multi-level latent variable optimization horizons. An illustrative example using this control framework can be found in Section 4. The paper ends with a conclusion outlining its implications in Section 5.

2. NOTATIONS AND PRELIMINARIES

2.1 Notation

$\mathbb{R}$ and $\mathbb{Z}$ represent the sets of real numbers and integers, respectively, with $\mathbb{Z}_+$ denoting the set of positive integers. The space $\mathbb{V}$ is associated with the variable v with dimension v. For a positive definite symmetric matrix P, $\|x\|_P$ is the weighted Euclidean norm of x, i.e., $\|x\|_P = \sqrt{x^\top P x}$; and $\mathrm{col}(A, B) \coloneqq \begin{bmatrix} A^\top & B^\top \end{bmatrix}^\top$ denotes the concatenation of matrices A and B.

2.2 Behavioral Systems Theory

In this section, we provide a summary of behavioral systems theory, with more comprehensive insights available in Willems (1986a,b), and the references therein. In the behavioral systems framework, dynamical system denoted as Σ is defined as $\Sigma = (\mathbb{T}^+, \mathbb{V}, \mathfrak{B})$ (Polderman and Willems, 1998), where $\mathbb{T}$ represents the time axis, $\mathbb{V}$ is a signal space, and $\mathfrak{B}$ corresponds to the system's behavior. According to this definition, a dynamical system is defined by its admissible trajectories, in line with the rationale of data-driven analysis. Given that process data is often discrete-time, we usually assume $\mathbb{T} = \mathbb{Z}^+$. The set of collected trajectories with a finite length L, denoted as $\tilde{v}_{|[1,L]}$, defines the behavior within the interval $[1, L]$:

$$\mathfrak{B}_{|[1,h]} \coloneqq \{\tilde{v} \in \mathbb{V} \mid \exists \tilde{v}' \in \mathfrak{B}, \tilde{v} = \tilde{v}'_{|[1,h]}\}, \tag{1}$$

where $\tilde{v}_{|[1,h]} \coloneqq \mathrm{col}(v(1), v(2), \ldots, v(h))$.

One key feature of the behavioral framework is the absence of causality presumption among the elements within the variable v. Nevertheless, it is feasible to establish an input/output partition for v, taking into account the 'freeness' of its elements. Consider the system's variable $v = \mathrm{col}(u, y) \in \mathfrak{B}$ as a partition of v. For each trajectory of u, there is a corresponding y such that $(u, y) \in \mathfrak{B}$, classifying u as a free variable. An input/output partition is defined when all elements of the vector variable u are free, while none of y are. We also denote $\mathbf{m}(\mathfrak{B})$ as the number of inputs and $\mathbf{n}(\mathfrak{B})$ as the state cardinality, which is the smallest state-space dimension among various state representations.

The behavior $\mathfrak{B}$ is considered time-invariant if it satisfies the condition $\sigma\mathfrak{B} \subset \mathfrak{B}$, where $\sigma v(k) = v(k+1)$. In the context of time-invariant systems, a finite *lag* represents the minimum number of time steps required for the past and future trajectories to become independent. This *lag* is denoted as $\mathbf{l}(\mathfrak{B})$ (Maupong et al., 2017). This property allows for the construction of a trajectory based on the behavior within a shorter time interval, as demonstrated by the following lemma.

Lemma 1. (Weaving Response (Markovsky et al., 2005)). Let Σ be a time-invariant system with lag $\mathbf{l}(\mathfrak{B})$ and $\tilde{v}_1, \tilde{v}_2 \in \mathfrak{B}_{|[1,L]}$ be two L-step trajectories. If $\tilde{v}_{1|[L-\mathbf{l}(\mathfrak{B})+1,L]} = \tilde{v}_{2|[1,\mathbf{l}(\mathfrak{B})]}$, then

$$\tilde{v}' \coloneqq \mathrm{col}\left(\tilde{v}_1, \tilde{v}_{2|[\mathbf{l}(\mathfrak{B})+1,L]}\right) \in \mathfrak{B}_{|[1,2L-\mathbf{l}(\mathfrak{B})]}. \tag{2}$$

This insight is important for receding horizon control algorithms that require past steps larger than $\mathbf{l}(\mathfrak{B})$.

We also explore linear time-invariant systems. A dynamical system is linear if $\mathbb{V}$ is a vector space, e.g., $\mathbb{V} = \mathbb{R}^{\mathrm{v}}$ and $\mathfrak{B}$ is a linear subspace of $\mathbb{V}^{\mathbb{T}}$ and closed in the topology of pointwise convergence (Willems et al., 2005). Alongside the time-invariant property, the behavior $\mathfrak{B}$ of a linear time-invariant (LTI) system can always be expressed using a kernel representation $\mathfrak{B} = \{v : \mathcal{K}(\sigma)v = 0\}$, where $\mathcal{K}$ is a polynomial matrix. The matrix $\tilde{\mathcal{K}} = [\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_N]$ is the coefficient matrix, and N is the order of the kernel representation. The behavior with such a representation is denoted as $\mathfrak{B} = \ker(\mathcal{K})$. A system within Σ is *controllable* if, for any $v \in \mathfrak{B}$, there exists a corresponding $\bar{v} \in \mathfrak{B}$ such that $\bar{v}(t) = v(t)$ for $t \leq t_1$ and $t \geq t_2$.

The L-step behavior of an LTI system can be represented by the column span of the Hankel matrix constructed from one of its T-step trajectories:

$$\mathfrak{H}_L(\tilde{v}) = \begin{bmatrix} v(1) & v(2) & \dots & v(T-L+1) \\ v(2) & v(3) & \dots & v(T-L+2) \\ \vdots & \vdots & \dots & \vdots \\ v(L) & v(L+1) & \dots & v(T) \end{bmatrix} \quad (3)$$

if the following lemma is satisfied:

Lemma 2. ((Markovsky and Dörfler, 2022)). Let the data $\tilde{v}$ generated by a system, i.e., $\tilde{v} \in \mathfrak{B}_{|[1,L]}$. Then, $\text{colspan}(\mathfrak{H}_L(\tilde{v})) = \mathfrak{B}_{|[1,L]}$ for $L > \mathbf{l}(\mathfrak{B})$ if and only if $\text{rank}(\mathfrak{H}_L(\tilde{v})) = \mathbf{m}L + \mathbf{n}$.

Lemma 2 generalizes the fundamental lemma in Willems et al. (2005) by relaxing the persistent excitation requirement on the input.

2.3 Dynamical Latent Variable Autoencoder (DLVAE)

A DLVAE (similar to Zhao et al. (2022)) has a structured latent variable with distinct time-step partitions as shown in Figure 1. In the context of DLVAE, process trajectories of length h, denoted as $\tilde{w}_{|[k-h+1,k]}$, are reconstructed. Each step, represented as $w(k)$, corresponds to an encoder ϕ, a latent variable $\ell(k)$, a decoder ψ, and a reconstructed $w(k)$ denoted as $\hat{w}(k)$. Notably, the same ϕ and ψ are consistently applied to all h steps of $w(k)$, $\ell(k)$, and $\hat{w}(k)$. The encoder ϕ maps $w(k)$ to $\ell(k)$ through an M-layer neural network, defined as $\ell(k) = (\phi_M \circ \phi_{M-1} \circ \cdots \circ \phi_1)(w(k)) := \phi(w(k))$. Each intermediate layer in ϕ follows the structure $x_m = \phi_m(W_m X_{m-1} + b_m)$, where x_m represents the set of variables in the m-th layer, while ϕ_m, W_m, and b_m correspond to the activation function, weighting matrix, and bias vector of the m-th layer, respectively. The decoder ψ which maps $\ell(k)$ to $\hat{w}(k)$, i.e. $\hat{w}(k) = \psi(\ell(k))$, employs a similar set of activation functions, weighting matrices, and bias vectors.

To maintain linearity within the hidden layer comprising $\tilde{\ell} = \text{col}(\ell(k-h+1), \dots, \ell(k))$, a Kernel Representation layer is introduced. It ensures $\tilde{\mathcal{K}}\tilde{\ell}_{|[k-h,k]} = \epsilon_{\mathcal{K}}$, where $\tilde{\mathcal{K}}$ is the coefficient matrix of the kernel space $\mathcal{K}$, and $\epsilon_{\mathcal{K}}$ represents the error in the kernel representation of linear latent variable dynamics. During DLVAE training, the following loss function is used:

$$\mathcal{L} = \lambda_1 \|\tilde{\mathcal{K}}\tilde{\ell}_{|[k-h,k]}\|^2 + \lambda_2 \sum_{i=k-h}^{k} \|w(i) - \psi(\phi(w(i)))\|^2. \quad (4)$$

Both error terms, each with its respective weighting constants λ_1 and λ_2, are optimized simultaneously. This minimizes the kernel space error $\epsilon_{\mathcal{K}}$ at time step k and the reconstruction error ϵ_A across all h time steps.

3. DPC WITH MULTI-LEVEL LATENT VARIABLE OPTIMIZATION HORIZON

To approximate multi-timescale dynamics, we employ multiple DLVAEs with varying sampling periods Δt, while limiting the number of time steps. Initially, historical data undergoes filtering and downsampling to align with these Δt for DLVAE training. Regarding prediction time horizons, the DLVAE with the largest Δt captures long-term dynamics in just a few steps, with this pattern persisting as Δt progressively decreases. This arrangement of DLVAEs

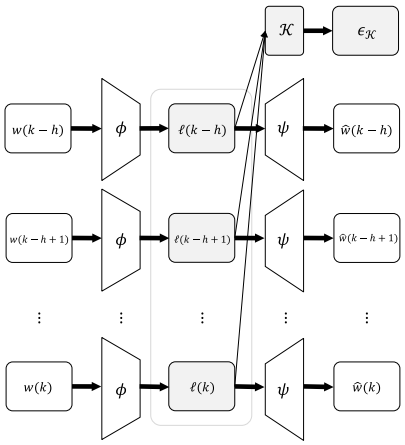

Fig. 1. The structure of a DLVAE.

in ascending order of Δt establishes a hierarchical structure for prediction horizons, exponentially extending the time span from top to bottom. Notably, this hierarchical pattern extends not only to the physical variables but also to the latent variables within the DLVAEs.

During setpoint tracking control, optimization is carried out by simultaneously accounting for the effects of all approximated dynamics. Our approach involves controlling the all i-level of latent variable ℓ_i to reach an equivalent setpoint $\ell_{r_i} = \phi(w_r)$ where w_r represents the setpoint of physical variables. By optimizing ℓ_i to match this setpoint, and subsequently decoding it to obtain the control action u_i through $\hat{w}_i = \psi(\ell_i)$, we achieve control over nonlinear processes within a linear framework.

3.1 Approximating Multi-timescale Dynamics from Resampled Process Data

Assuming an adequate original sampling rate that can capture the fastest process dynamics requiring control, we can approximate these dynamics at different timescales by re-sampling time-series process data. This technique enables the creation of a multi-level structure for predictive control. Specifically, each level's horizon, denoted as Δt_i, can be a multiple of the previous level's, such as $\Delta t_i = n_i \Delta t_{i-1}$. For instance, in a 3-level configuration shown in Figure 2, where sampling periods adhere to $\Delta t_3 = 4\Delta t_2 = 16\Delta t_1$, each level optimizes for four future steps. Consequently, although the optimization within each horizon extends effectively to $64\Delta t_1$ into the future, the number of steps that require optimization across all levels remains limited to just 12 (specifically, $\tilde{w}_{1|[k+1,k+4]}$, $\tilde{w}_{2|[k+1,k+4]}$, and $\tilde{w}_{3|[k+1,k+4]}$). Moreover, as more levels are added, the length (time span) of the optimized future trajectory grows exponentially, while the computational complexity increases linearly. At each horizon level, we obtain the latent variable with corresponding kernel space by training a DLVAE using data downsampled to its sampling period. In the first level, the DLVAE is trained on data with basic noise filtering. However, starting from the second level, we apply a low-pass filter before downsampling to eliminate frequency components higher than half of the reduced sampling frequency, thus preventing aliasing

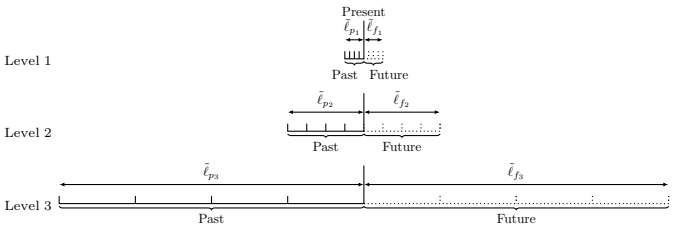

Fig. 2. Three-level horizon.

(Proakis, 2001). The low-pass filter for the i-th level can be represented as follows:

$$w_i(k+1) = \sum_{j=0}^{n_i-1} \alpha_{ij} w_{i-1}(k-j), \tag{5}$$

where n_i is the order and α_{ij} are the coefficients of the filter respectively. As the scale of horizon increases, low-pass filtering gradually removes the fast dynamics, revealing the slow dynamics that require long horizon to show significant changes. As the horizon scale increases, the low-pass filtering gradually attenuates fast dynamics, unveiling the slower dynamics that exhibit significant changes over longer horizons. Then, we obtain the i-th level latent variable ℓ_i where $\tilde{\mathcal{K}}_i \tilde{\ell}_i = \epsilon_i$ by training DLVAE$_i$ from the filtered downsampled data in the subsequent levels.

While the relationship between the w_i of adjacent levels can be represented by the filtering-downsampling equation, it is not the case for ℓ_i. Here, we can establish a linear map M_i between the ℓ_i of adjacent levels equivalent to (5), i.e.

$$\ell_i = M(\ell_{i-1}) \Leftrightarrow (5), i = 2, \ldots, L. \tag{6}$$

The linear map M_i can be found by linking the ℓ_i of the adjacent levels during the DLVAEs training such that

$$\ell_i(k+1) = \sum_{j=0}^{n_i-1} \beta_{ij} \ell_{i-1}(k-j), \tag{7}$$

where n_i is the order of the lowpass filter in (5) and β_{ij} is coefficient of the linear mapping at i-th level.

During the DLVAEs training, the weightings and biases of ϕ_i and ψ_i, $\tilde{\mathcal{K}}_i$, as well as β_{ij} in (7), are optimized (simultaneously) by minimizing the following loss function:

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{L} \left(\lambda_{1_i} \|\epsilon_{\mathcal{K}_i}\|_M^2 + \lambda_{2_i} \sum_{v=k-h_i}^{k} \|w_i(v) - \psi(\phi(w_i(v)))\|_M^2 \right) \\ &+ \sum_{i=2}^{L} \left(\lambda_{3_i} \sum_{v=k-\frac{h_{i-1}}{\Delta t_i}}^{k} \left\| \ell_i(v) - \sum_{j=0}^{n_i-1} \beta_{ij} \ell_{i-1}(v-1-j) \right\|_M^2 \right),\end{aligned} \tag{8}$$

where $\epsilon_{\mathcal{K}_i} = \tilde{\mathcal{K}}_i \tilde{\ell}_{i_{|[k-h_i,k]}}$, L is the number of levels (or DLVAEs), h_i is the trajectory length of $\tilde{w}_{i_{|[k-h_i,h_i]}}$, and λ_{1_i}, λ_{2_i}, and λ_{3_i} are the scalar weightings for the reconstruction errors, kernel space errors and linear mapping error derived from (7) at i-th level, respectively so that the mean squared errors denoted as $\|\bullet\|_M^2$ are within similar ranges. The third term of (8) shows that only certain steps of $\tilde{\ell}_i$ that has the matching time steps with the $\tilde{\ell}_{i-1}$ is linked. As an illustration, the structure of the linked DLVAEs used to extract $\tilde{\ell}_i$ for three-level horizon in the previous example is depicted in Figure 3. The multi-level structure allows distant future predictions within each receding horizon, serving as trend guidance. Since only the first step of the first level is executed per time step, approximation errors have minimal impact on control. We aim to further reduce these errors by integrating dissipativity-based design techniques, as suggested by Zhao et al. (2022).

3.2 Multi-level DPC Control Algorithm

After the construction of multi-level horizons, we can perform online tracking control through optimization (as shown in (9)). During the online tracking control of w, we are controlling $\tilde{\ell}_i$ of all levels simultaneously. According to Lemma 1, each horizon need to have sufficient number of "past" steps to ensure the the optimized trajectory belongs to the behavior. By separating the total number of steps (trajectory length) h_i into p_i past and f_i future steps ($p_i + f_i = h_i$), then p_i must satisfy $p_i \geq \mathbf{l}(\mathfrak{B}_i)$ where $\mathfrak{B}_i = \ker(\mathcal{K}_i)$. At the k-th step, $\tilde{\ell}_i$ can be partitioned into the "past" and "future" trajectories as $\tilde{\ell}_{i_{|[k-p_i+1,k+f_i]}} = \text{col}(\tilde{\ell}_{p_i}, \tilde{\ell}_{f_i}) = \text{col}(\tilde{\ell}_{i_{|[k-p_i+1,k]}}, \tilde{\ell}_{i_{|[k+1,k+f_i]}})$, and the same applies to $\tilde{w}$ (as well as $\tilde{\hat{w}}$) where $\tilde{w}_{i_{|[k-p_i+1,k+f_i]}} = \text{col}(\tilde{w}_{p_i}, \tilde{w}_{f_i})$. To achieve tracking control such that $w(k)$ approaching the setpoint w_r, we can first obtain the setpoint trajectories $\text{col}(\tilde{\ell}_{p_{r_i}}, \tilde{\ell}_{f_{r_i}})$ by encoding the $\text{col}(\tilde{w}_{p_i}, \tilde{w}_r)$ using ϕ_i where $\tilde{w}_r = \text{col}(w_r, \ldots, w_r)$ is a reference trajectory with a constant setpoint w_r. Then, during the online control, $\tilde{\ell}_k$ is optimized such that $\tilde{\ell}_{p_i} = \tilde{\ell}_{p_{r_i}}$ to fulfill Lemma 1 while $\tilde{\ell}_{f_i}$ is approaching $\tilde{\ell}_{f_{r_i}}$ as reinforced in (9b). At the same time, the linear mapping (7) between adjacent $\tilde{\ell}_{f_i}$ is applied so that upon decoding, $\hat{w}_i$ at corresponding time steps align with the filtering-downsampling processes (5). The optimization problem at k-th time step can be formulated as follows:

$$\min_{\tilde{\ell}_{f_i}, \ldots, \tilde{\ell}_{f_L}} \sum_{i=1}^{L} \|\tilde{\ell}_{f_i} - \tilde{\ell}_{f_r}\|_{Q_i}^2 + \lambda_i \|\epsilon_{\mathcal{K}_i}\|^2 \tag{9a}$$

$$\text{s.t. } \tilde{\mathcal{K}}_i \begin{bmatrix} \tilde{\ell}_{p_{r_i}} \\ \tilde{\ell}_{f_i} \end{bmatrix} = \epsilon_{\mathcal{K}_i}, \qquad i = 1, \ldots, L \tag{9b}$$

$$\ell_i(k+1) = \sum_{j=0}^{n_i-1} \beta_{ij} \ell_{i-1}(k-j), \qquad i = 2, \ldots, L \tag{9c}$$

$$\tilde{\ell}_{f_i} \in \mathbb{L}_i^{[k+1,k+f_i]}, \qquad i = 1, \ldots, L \tag{9d}$$

where (9c) is incorporated from (7), Q_i is the weighting matrix for $\tilde{\ell}_{f_i}$, and λ_i is a scalar weighting for $\epsilon_{\mathcal{K}_i}$. (9d) includes additional constraints necessary for the system or controller. The first level optimized future trajectory $\tilde{\ell}_{f_1}^*$ can then be encoded to produce $\tilde{w}_{f_1}^*$, and the first step of the input trajectory $u^*(k+1)$ of $\tilde{u}_f^*$, where $\tilde{w}_{f_1}^* = \text{col}(\tilde{y}_f^*, \tilde{u}_f^*)$, is implemented.

Remark 3. As ℓ_i lacks inherent physical properties, its controllability can be verified post-DLVAE training. We can use the trajectory $\tilde{\ell}_{i_{|[1,T]}}$, obtained by encoding historical data $\tilde{w}_{i_{|[1,T]}}$ as $\ell_i(k) = \phi_i(w_i(k))$, to construct $\mathfrak{H}_{1i} = \mathfrak{H}(\tilde{\ell}_{i_{|[1,T-1]}})$ and $\mathfrak{H}_{2i} = \mathfrak{H}(\tilde{\ell}_{i_{|[2,T]}})$. Then, we can apply the controllability test for ℓ_i proposed in Mishra et al. (2020). Besides, the partitioning of $\tilde{\ell}_i$ into "past" and "future" ($\tilde{\ell}_i = \text{col}(\tilde{\ell}_{p_i}, \tilde{\ell}_{f_i})$) is valid as long as Lemma 2

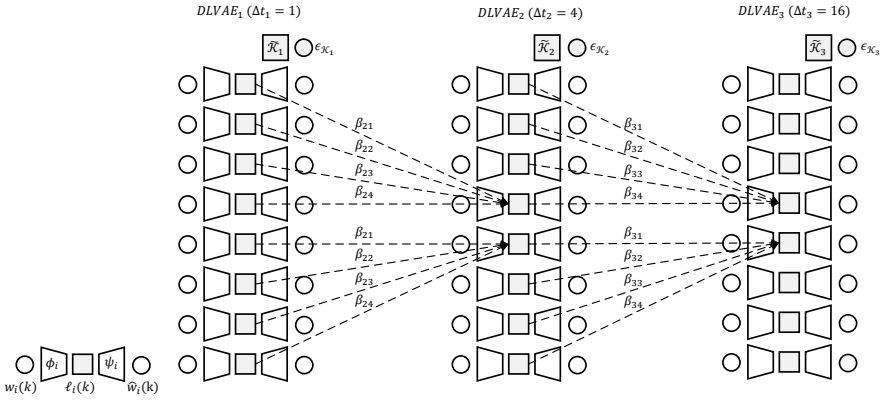

Fig. 3. The structure of linked DLVAEs for the construction of DPC with three-level horizon.

Model parameters		ρ	0.9342 kgL^{-1}
c_1	30.828 h^{-1}	C_p	3.01 $kJkg^{-1}K^{-1}$
c_2	86.688 h^{-1}	V	101
c_3	0.1 KkJ^{-1}	Operating point	
k_{10}, k_{20}	1.287e12 h^{-1}	C_{A_s}	1.235 $molL^{-1}$
k_{30}	9.043e9 $m^3mol^{-1}h^{-1}$	C_{B_s}	0.9 $molL^{-1}$
ΔH_{AB}	4.2 $kJmol^{-1}$	T_s	134.14 $°C$
ΔH_{BC}	-11.0 $kJmol^{-1}$	T_{c_s}	128.95 $°C$
ΔH_{AD}	-41.85 $kJmol^{-1}$	F_s	18.83 h^{-1}
T_0	130.0 $°C$	Q_s	-4495.7 kJh^{-1}
E_1	9758.3 K	C_{A0_s}	5.1 $molL^{-1}$
E_2	8560 K		

Table 1. Model parameters & operating points

holds. This negates the requirement to establish causality, highlighting one of the key benefits of behavioral systems theory.

4. ILLUSTRATIVE EXAMPLE

We use a 2-input 2-output nonisothermal CSTR with the reaction $A \to B \to C$ and $A \to D$ as a case study. The CSTR can be described in the following equations (Klatt and Engell, 1998):

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A_0} - C_A) - k_1 C_A - k_3 C_A^2 \tag{10a}$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + k_1 C_A - k_2 C_B \tag{10b}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + c_1(T_c - T) - \frac{k_1 C_A \Delta H_{AB} + k_2 C_B \Delta H_{BC} + k_3 C_A^2 \Delta H_{AD}}{\rho C_p} \tag{10c}$$

$$\frac{dT_c}{dt} = c_2(T - T_c) + c_3 Q \tag{10d}$$

where T is the reactor temperature, T_c is the coolant temperature, T_0 is the inlet temperature, Q is the amount of heat removed by the coolant, and the reaction rate coefficients k_i, for $i = 1, 2, 3$, are described by the Arrhenius law $k_i = k_{i0} e^{\frac{-E_i}{T+273.15}}$. The values of the model parameters and operating point are given in Table 1. The model is presumed to be unknown during the control design.

Amongst the variables, F (Input 1, u_1) and Q (Input 2, u_2) are manipulated to control the C_B (Output 1, y_1) and T (Output 2, y_2). To begin the controller design, the model is employed to generate a noisy dataset $\tilde{w}$ comprising 25,000 data points. This dataset is created by applying random input trajectories of u_1 and u_2 to produce output trajectories of y_1 and y_2. The resulting process data, $\tilde{w}$, is then normalized. From the 25,000 data points, 20,000 are allocated for training, while the remaining 5,000 are reserved for validation. For training and validation, the linked three DLVAEs, as depicted in Figure 3 corresponding to the three-level horizon structure in Table 2, are employed with specifications outlined in Table 3. In this setup, the Encoder layer (ϕ_i) and the Decoder layer (ψ_i) utilize the leaky-ReLU activation function with a slope of 2. As shown in Figure 4, the process output is able to track the setpoint during online control. We also implement a single-level DPC with a sampling interval of Δt_1, featuring 12 future steps, which has an equivalent computational complexity as the three-level DPC. However, this approach leads to diverging trajectories, primarily because it struggles to account for the long-term inverse dynamics.

level	ΔT (min)	No. of Past Steps	No. of Future Steps
1	1	4	4
2	4	4	4
3	16	4	4

Table 2. Three-level horizons structure.

Parameter	Value
Number of steps (h_i)	8
Dimension of Input ($w_i(k)$) at i-th step	(4,1)
Dimension of Encoder (ϕ_i) per input	(8,1)
Dimension of Latent/Hidden Layer (ℓ_i) per Encoder	(4,1)
Dimension of Decoder (ψ_i) per Hidden Layer	(8,1)
Dimension of Output ($\hat{w}_i(k)$) per Decoder	(4,1)
Dimension of Kernel Layer ($\mathcal{K}_i$)	(8,32)
Dimension of Kernel Space Error ($\epsilon_{\mathcal{K}_i}$)	(8,1)
Weightings for $\epsilon_{\mathcal{K}_i}$ ($\lambda_{\ell_{1_i}}$)	$10/i$
Weightings for reconstruction error $\epsilon_{A_i}(\lambda_{2_i})$	$1/i$
Weightings for linear mapping error (λ_{3_i})	$5/i$

Table 3. Structure of DLVAE_i. (m, n) defines the dimension.

5. CONCLUSION

In this novel data-predictive control approach for nonlinear multi-timescale processes, we employ a series of DLVAEs to approximate the multi-timescale dynamics of nonlinear processes. Through training with low-pass filtering and resampling of historical input-output data, we create dynamical latent variables. These latent variables belong to LTI systems or kernel spaces, effectively restores the process nonlinearity through encoding and decoding. During the training phase, our focus extends beyond imposing

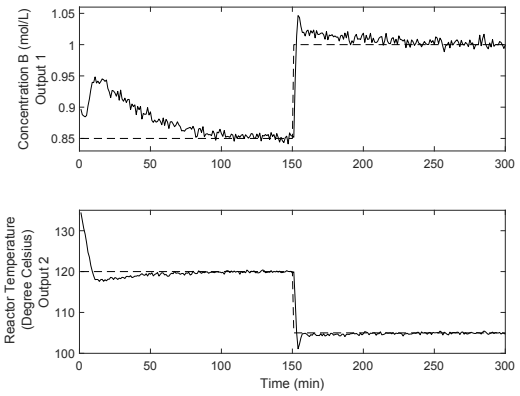

Fig. 4. Control performance of three-level DPC with outputs tracking the setpoint (dotted line)

kernel spaces and data reconstruction; we also establish connections among latent variables from different DLVAEs at matching time-steps. This collective approach allows us to span a wide prediction time horizon with limited, non-uniformly spaced steps, encompassing the current, near, and distant future. During online tracking control, our method guides latent variables from each DLVAE to their respective setpoints, derived from desired physical variable values, while ensuring consistency in physical variable values at matching time-steps. This approach enables us to control nonlinear multi-timescale systems effectively in a linear framework. Future work includes the design of stabilizing conditions for the proposed DPC method, such as incorporating trajectory-based dissipativity conditions.

REFERENCES

Christofides, P.D. and Daoutidis, P. (1996). Feedback control of two-time-scale nonlinear systems. *Int. J. Control*, 63(5), 965–994.

Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th ECC*, 307–312. IEEE.

Ellis, M., Heidarinejad, M., and Christofides, P.D. (2013). Economic model predictive control of nonlinear singularly perturbed systems. *J. Process Control*, 23(5), 743–754.

Huang, B. and Kadali, R. (2008). *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*. Springer.

Jian, N.L., Zabiri, H., and Ramasamy, M. (2023). Control of the multi-timescale process using multiple timescale recurrent neural network-based model predictive control. *Ind. Eng. Chem. Res.*, 62(15), 6176–6195.

Klatt, K.U. and Engell, S. (1998). Gain-scheduling trajectory control of a continuous stirred tank reactor. *Comput. Chem. Eng.*, 22(4-5), 491–502.

Kumar, A., Christofides, P.D., and Daoutidis, P. (1998). Singular perturbation modeling of nonlinear processes with non-explicit time-scale multiplicity. *Chem. Eng. Sci.*, 53(8), 1491–1504.

Lévine, J. and Rouchon, P. (1991). Quality control of binary distillation columns via nonlinear aggregated models. *Automatica*, 27(3), 463–480.

Ma, R., Liu, C., Breaz, E., Briois, P., and Gao, F. (2018). Numerical stiffness study of multi-physical solid oxide fuel cell model for real-time simulation applications. *Appl. Energy*, 226, 570–581.

Markovsky, I. and Dörfler, F. (2022). Identifiability in the behavioral setting. *IEEE Trans. Automat. Control*, 68(3), 1667–1677.

Markovsky, I. and Rapisarda, P. (2008). Data-driven simulation and control. *Int. J. Control*, 81(12), 1946–1959.

Markovsky, I., Willems, J.C., Rapisarda, P., and De Moor, B.L. (2005). Algorithms for deterministic balanced subspace identification. *Automatica*, 41(5), 755–766.

Maupong, T., Mayo-Maldonado, J.C., and Rapisarda, P. (2017). On Lyapunov functions and data-driven dissipativity. *IFAC-PapersOnLine*, 50(1), 7783–7788.

Mishra, V.K., Markovsky, I., and Grossmann, B. (2020). Data-driven tests for controllability. *IEEE Control Syst. Lett.*, 5(2), 517–522.

Polderman, J.W. and Willems, J.C. (1998). *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media.

Proakis, J.G. (2001). *Digital signal processing: principles algorithms and applications*. Pearson Education India.

Radisavljević-Gajić, V., Milanović, M., and Rose, P. (2019). *Multi-stage and multi-time scale feedback control of linear systems with applications to fuel cells*. Springer.

Tan, C.K., Tippett, M.J., and Bao, J. (2016). Model predictive control with non-uniformly spaced optimization horizon for multi-timescale processes. *Comput. Chem. Eng.*, 84, 162–170.

Tang, J.W., Yan, Y., Bao, J., and Huang, B. (2022). Data-predictive control of multi-timescale processes. In *2022 IEEE International Symposium on AdCONIP*, 73–77.

Tang, J.W., Yan, Y., Bao, J., and Huang, B. (2023). Multi-level data-predictive control for linear multi-timescale processes with stability guarantee. *J. Process Control*, 130, 103083.

Wei, L., Yan, Y., and Bao, J. (2020). A data-driven predictive control structure in the behavioral framework. *IFAC-PapersOnLine*, 53(2), 152–157.

Willems, J.C. (1986a). From time series to linear system-Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5), 561–580.

Willems, J.C. (1986b). From time series to linear system-Part II. Exact modelling. *Automatica*, 22(6), 675–694.

Willems, J.C., Rapisarda, P., Markovsky, I., and De Moor, B.L. (2005). A note on persistency of excitation. *Syst. Control Lett.*, 54(4), 325–329.

Yan, Y., Bao, J., and Huang, B. (2023). Distributed data-driven predictive control via dissipative behavior synthesis. *IEEE Trans. Automat. Control*, 1–16. doi: 10.1109/TAC.2023.3298281.

Yang, C. and Zhu, Y. (2021). Two-time scaled identification for multi-energy systems. *Control Eng. Pract.*, 113, 104845.

Zhang, X., Jiang, W., Yu, S., Xu, X., and Li, Z. (2022). A dual-level model predictive control scheme for multi-timescale dynamical systems. *IEEE Trans. Syst. Man Cybern. Syst.*

Zhao, M., Yan, Y., Bao, J., and Wang, W. (2022). Data predictive control of nonlinear process feature dynamics through latent variable behaviours. *Comput. Chem. Eng.*, 107857.