Technometrics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/utch20

Tensor-Based Temporal Control for Partially Observed High-Dimensional Streaming Data

Zihan Zhang, Shancong Mou, Kamran Paynabar & Jianjun Shi

To cite this article: Zihan Zhang, Shancong Mou, Kamran Paynabar & Jianjun Shi (16 Oct 2023): Tensor-Based Temporal Control for Partially Observed High-Dimensional Streaming Data, Technometrics, DOI: 10.1080/00401706.2023.2271060

To link to this article: https://doi.org/10.1080/00401706.2023.2271060

TECHNOMETRICS
2023, VOL. 00, NO. 0, 1–13
https://doi.org/10.1080/00401706.2023.2271060

Taylor & Francis
Taylor & Francis Group

Tensor-Based Temporal Control for Partially Observed High-Dimensional Streaming Data

Zihan Zhang, Shancong Mou, Kamran Paynabar, and Jianjun Shi

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA

ABSTRACT

In advanced manufacturing processes, high-dimensional (HD) streaming data (e.g., sequential images or videos) are commonly used to provide online measurements of product quality. Although there exist numerous research studies for monitoring and anomaly detection using HD streaming data, little research is conducted on feedback control based on HD streaming data to improve product quality, especially in the presence of incomplete responses. To address this challenge, this article proposes a novel tensor-based automatic control method for partially observed HD streaming data, which consists of two stages: offline modeling and online control. In the offline modeling stage, we propose a one-step approach integrating parameter estimation of the system model with missing value imputation for the response data. This approach (i) improves the accuracy of parameter estimation, and (ii) maintains a stable and superior imputation performance in a wider range of the rank or missing ratio for the data to be completed, compared to the existing data completion methods. In the online control stage, for each incoming sample, missing observations are imputed by balancing its low-rank information and the one-step-ahead prediction result based on the control action from the last time step. Then, the optimal control action is computed by minimizing a quadratic loss function on the sum of squared deviations from the target. Furthermore, we conduct two sets of simulations and one case study on semiconductor manufacturing to validate the superiority of the proposed framework.

ARTICLE HISTORY
Received November 2022
Revised September 2023

KEYWORDS
Feedback control; High dimension; Partial observation; Streaming data; Tensor

1. Introduction

In advanced manufacturing processes, high-dimensional (HD) streaming data (e.g., sequential images or videos) have been widely used to measure and inspect the product quality. Examples include overlay measurements in semiconductor manufacturing (Zhong, Paynabar, and Shi 2023) and dimensional deformation profiles of fuselages in the aircraft assembly process (Zhong et al. 2022). Although numerous research has been done for monitoring and anomaly detection using HD streaming data (Thudumu et al. 2020), little research has been conducted on feedback control to improve HD quality response, especially when data is partially observed. For example, in semiconductor manufacturing, overlay errors induced by the pattern misalignment between adjacent layers in the lithography process can be controlled by adjusting the position/orientation of wafers, and the lens height, as indicated in Figure 1(a). Specifically, the entire wafer is comprised of identical rectangular fields, and each field has one chip fabricated through one exposure. After each exposure, the wafer stage moves horizontally to enable another time of exposure. As shown in Figure 1(b), the grids represent the boundaries of the cells; and the vectors are the measurement 2D vector, whose value on each axis denotes the overlay error on the corresponding axis, that is, the relative locational difference between two adjacent layers with the starting point on the previous layer and the endpoint on the current layer. The collection of all overlay measurements gives a sketch of the entire overlay vector field of the wafer. However, only limited fields in each wafer can be monitored by using a costly high-resolution micro camera, which results in incomplete measurements. Developing a control strategy to reduce the overlay errors is challenging due to the high dimensionality, complex spatio-temporal structure, and incomplete measurements. To address these challenges, this article proposes a new tensor-based automatic control method for HD streaming data with missing observations.

To create a feedback control model, the first step is to build a predictive model to quantify the relationship between the HD response and control variables. To address the high-dimensionality issue, traditional dimension reduction methods such as principal component regression (PCR) (Wold, Esbensen, and Geladi 1987) and partial least square (PLS) (Zhao et al. 2013) are widely used. Those methods first, extract features from the raw input data, and then, study the correlation between the features and HD response. Although PCR and PLS can reduce the data dimension based on the vectorized data, they fail to explore the spatial structure within high-order tensors such as images (Gahrooei et al. 2021). Recently, tensor analysis and multilinear algebra techniques have been used in HD streaming data modeling and analysis and provided promising results in many applications (Gaw, Yousefi, and Gahrooei 2021). For example, Yan, Paynabar, and Pacella (2019) propose a tensor regression model to link the HD response with the scalar input

CONTACT Kamran Paynabar *kamran.paynabar@isye.gatech.edu* H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA 30332.

Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/TECH*.

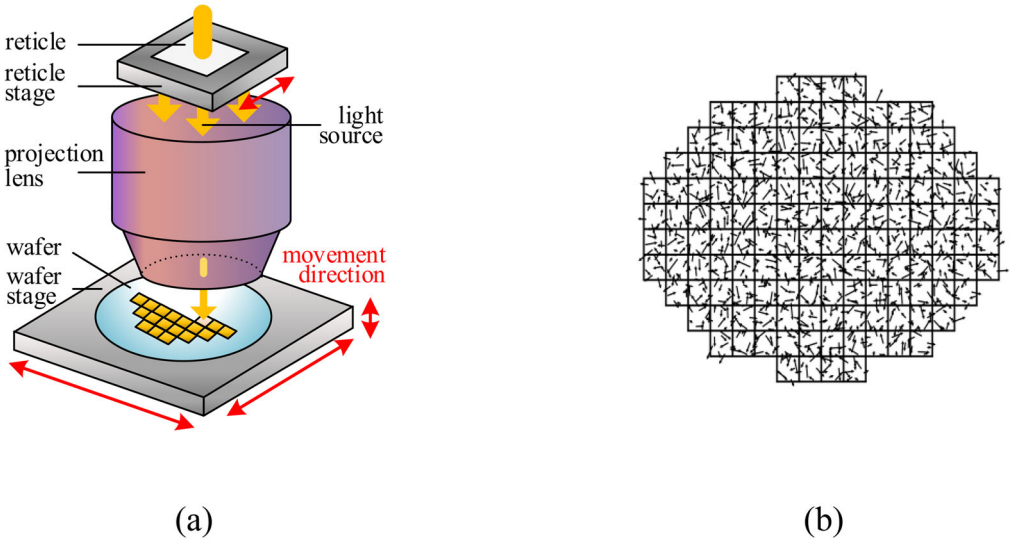

Figure 1. The overlay errors and the photolithography process. (a) a simulator for photolithography process (b) overlay errors.

through a tensor coefficient, in which Tucker decomposition (Kolda and Bader 2009) is adopted to represent the coefficient in a low-dimensional (LD) space. Lock (2018) develops a tensor-on-tensor regression (TOT) model, in which both inputs and outputs are in tensor format and approximated using CP decomposition (Kolda and Bader 2009). To relax the data homogeneity assumption for covariates in the TOT model, Gahrooei et al. (2021) propose a multiple tensor-on-tensor regression method. Besides, Llosa-Vite and Maitra (2023) propose a tensor-variate linear model with tensor-variate normal (TVN) errors, in which they render four types of low-rank structure on the model coefficient and allow the errors to follow a TVN distribution. Based on independent responses, they can estimate both model coefficients and covariance matrices for the TVN from the data.

Because these regression methods assume no temporal correlation in the response sequence, they cannot properly model HD streaming data, and hence, they cannot be used for online process control. To address this issue, Zhong, Paynabar, and Shi (2023) develop a tensor-based time series (TTS) model under the autoregressive with exogenous variables (ARX) framework to consider both spatial and temporal structures of responses, and further execute a one-step-ahead predictive control. Although their proposed model has effective process modeling capability, it assumes that the HD responses are completely observable without missing observations, which may not be true in many manufacturing processes for the following reasons: (i) The sensing capability may be insufficient. For example, due to budget limitations, sensing cost, and/or space constraints, the number of sensors is typically less than the number of variables to be monitored in a fuselage assembly process (Zhong et al. 2022); (ii) It may not be feasible to fully transfer data from high-resolution sensors to the data center or computing unit, such as the case in the solar flare detection problem (Gómez, Li, and Paynabar 2022), where partial solar images are transmitted back to the data center on earth, due to bandwidth restrictions. (iii) Faulty sensors may not be replaced timely at their early degradation stages (Zhang and Yang 2021). Thus, not all data will be collected or measured with high quality, which leads to missing values.

When HD responses involve incomplete entries, further research is required to handle the missing information. An intuitive solution is to first recover the incomplete responses using tensor completion (e.g., decomposition (Xue et al. 2021) or rank minimization (Gandy, Recht, and Yamada 2011)), and then construct the regression model using restored responses. However, due to the strong assumption on low-rank, tensor completion only works in a limited low-rank range. Additionally, this approach does not take the correlation between the response and control variables into account, when imputing the missing data. To address this issue, a popular approach is to impose low-rank assumptions directly on the data and shift the modeling of temporal dynamics to temporal factors. For example, Xiong et al. (2010) propose a probabilistic tensor factorization for temporal relational data, in which they assume that each time feature vector depends only on its immediate predecessor, and thus ignores the long-range dependency among responses. Besides, they only focus on the prediction (extrapolation) rather than imputing the embedded missing information. Yu, Rao, and Dhillon (2016) propose a temporal regularized matrix factorization framework, in which temporal dependencies are incorporated into matrix factorization models. However, their work (i) only involves the latent temporal vector embeddings rather than higher-order tensors, (ii) ignores the involvement of control actions to adjust the process, and (iii) ignores the correlation among noise terms by assuming a Gaussian distribution. Later, Chen and Sun (2020) propose a Bayesian temporal factorization framework for multi-dimensional time series prediction, but they (i) ignore the spatial correlation among the response since they assume that the observed entries are independent and each entry follows a Gaussian distribution, and (ii) do not mention the potential to involve the control execution in their future work, which makes it harder for readers to conduct control actions. In addition, although these two works consider the data completion, they only validate their performance under a relatively low missing ratio (i.e., less than 60%), which limits their applicability. To enable the data imputation in a wider range of missing ratio, Wang et al. (2021) propose an augmented tensor regression framework to recover incomplete responses by introducing the

system equation to provide additional information for the tensor response completion based on the tensor nuclear norm. Their work achieves good completion performance under high missing ratios, such as 90%. Furthermore, by imposing the low-rank property, sparsity, and fusion constraints, Zhou et al. (2021) derive the finite-sample error bound of the predictor for the tensor-based autoregressive model whose coefficients admit a CP-decomposition structure. For multivariate time-series forecasting, Chen and Sun (2020) propose a low-rank autoregressive tensor completion method. However, the proposed model cannot be applied to online estimation since it must be retrained for every prediction window, and this method fails to consider the potential relationship between the incomplete response and process inputs.

To achieve the feedback control for incomplete HD streaming data, this article will address imputation challenges for building a tensor-based time-series model with exogenous variables and an online control model using partial data. These challenges include that the accuracy of parameter estimation is impacted by partially observed autocorrelated responses; and incomplete online observations reduce the efficiency of control actions.

The proposed framework is illustrated in Figure 2. As shown in this figure, the proposed framework involves two stages: (i) offline modeling, and (ii) online control. For offline modeling, we propose a one-step method that learns model coefficients for the tensor-based ARX model and recovers missing entries of responses simultaneously. For online control, an optimal control law is derived based on the trained model and is employed to minimize the predicted output deviation from a target. In the online control stage, for each new sample, we first need to complete partial observations. To this end, an online completion strategy is proposed to impute the missing values considering its low-rank information and the one-step-ahead prediction result from the control action in the last time step.

The rest of the article is organized as follows: Section 2 elaborates the tensor-based automatic control model for incomplete HD streaming data and discusses its advantages. In Section 3, we propose a one-step algorithm for offline modeling, including response completion and parameter estimation. Based on the trained model, we make the one-step-ahead predictive control and propose an online completion algorithm to impute new missing observations in Section 4. Next, we validate the proposed methodology and compare it with some benchmark methods in terms of offline completion and online control performance using simulations and a case study of overlay errors in Sections 5 and 6, respectively. Finally, Section 7 concludes the article.

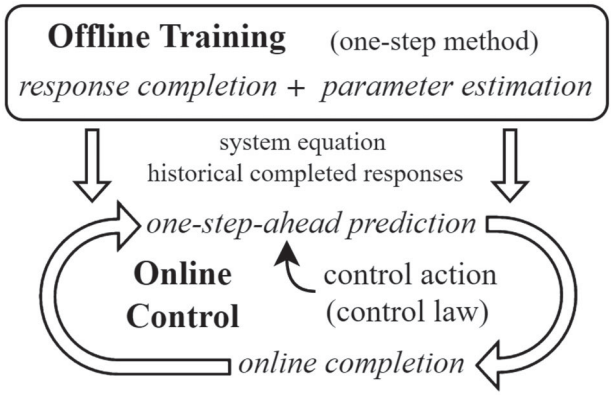

Figure 2. The proposed tensor-based control framework with partially observed HD streaming data.

2. Partially Observed Tensor-Based Automatic Control Model

In this section, we introduce tensor notations, basic assumptions, and the problem formulation.

2.1. Problem Setup

Consider a set of training data of size m, which contains a sequence of historical incomplete tensor responses $\mathcal{Y}_t^- \in \mathbb{R}^{Q_1 \times \cdots \times Q_d} (t = 1, \ldots, m)$ (corresponding to unknown complete $\mathcal{Y}_t$), and input control variables $\mathbf{X}_t \in \mathbb{R}^P (t = 1, \ldots, m)$ collected over time. Here, we assume a uniformly-at-random missing pattern (Liu et al. 2013). To model both the spatio-temporal structure of $\mathcal{Y}_t$ using partially observed $\mathcal{Y}_t^-$, and its relationship with input $\mathbf{X}_t$, we propose a partially observed tensor-based automatic control (poTAC) method based on the TTS model in Zhong, Paynabar, and Shi (2023). Specifically, the relationship among the current response tensor $\mathcal{Y}_t$, the previous response tensors $\mathcal{Y}_{t-j}$, $j = 1, \ldots, p$, as well as the control variable $\mathbf{X}_t$ is modeled using the following tensor-based ARX as

$$\begin{aligned} \mathcal{Y}_t &= \Sigma_{j=1}^{p} \mathcal{Y}_{t-j} * \mathcal{A}_j + \mathbf{X}_{t-1} * \mathcal{B} + \mathcal{E}_t \\ \text{with} \quad & P_{\Omega_t}(\mathcal{Y}_t) = P_{\Omega_t}\left(\mathcal{Y}_t^-\right), \end{aligned} \tag{1}$$

where p is the autoregressive order, which can be selected either based on domain knowledge or using cross validation (or AIC/BIC metrics); $\mathcal{E}_t \in \mathbb{R}^{Q_1 \times \cdots \times Q_d}$ represents the tensor of random noises; $\mathcal{A}_j \in \mathbb{R}^{Q_1 \times \cdots \times Q_d \times Q_1 \times \cdots \times Q_d} (j = 1, \ldots, p)$ and $\mathcal{B} \in \mathbb{R}^{P \times Q_1 \times \cdots \times Q_d}$ are the coefficients of corresponding inputs, which reflect the temporal correlation within HD streaming data; the operator $*$ is the contraction product of two tensors defined as $(\mathbf{X}_{t-1} * \mathcal{B})_{q_1,\ldots,q_d} = \Sigma_p (\mathbf{X}_{t-1})_p (\mathcal{B})_{p,q_1,\ldots,q_d}$ (Gahrooei et al. 2021); a projection function projects the tensor $\mathcal{R} \in \mathbb{R}^{Q_1 \times \cdots \times Q_d}$ onto the observed set Ω_t at time step t, such that $[P_{\Omega_t}(\mathcal{R})]_{(q_1,\ldots,q_d)} = \mathcal{R}_{(q_1,\ldots,q_d)}$ when $(q_1, \ldots, q_d) \in \Omega_t$.

To simplify the notations, we fold m complete tensor observations over time t into a higher-order tensor denoted by $\tilde{\mathcal{Y}} \in \mathbb{R}^{m \times Q_1 \times \cdots \times Q_d} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_m\}$. As illustrated in Figure 3, $\tilde{\mathcal{Y}}_{(-r)}^- (r = 0, \ldots, p)$ represent tensor responses within $[p - r + 1, m - r]$ time steps from $\tilde{\mathcal{Y}}^-$. Similarly, $\mathbf{X}_{(-1)} \in \mathbb{R}^{(m-p) \times P}$ and $\mathcal{E} \in \mathbb{R}^{(m-p) \times Q_1 \times \cdots \times Q_d}$ are augmented control variables and noises, respectively. Consequently, (2.1) can be written as

$$\begin{aligned} \tilde{\mathcal{Y}}_{(0)} &= \Sigma_{j=1}^{p} \tilde{\mathcal{Y}}_{(-j)} * \mathcal{A}_j + \mathbf{X}_{(-1)} * \mathcal{B} + \mathcal{E} \\ \text{with} \quad & P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right), \end{aligned} \tag{2}$$

where $\Omega = \{\Omega_1, \ldots, \Omega_m\}$. Besides, $\mathcal{E}$ is assumed to follow a tensor normal distribution as $\mathcal{E} \sim \mathcal{N}(0, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2, \ldots, \mathbf{\Sigma}_d, \mathbf{\Sigma}_b)$,

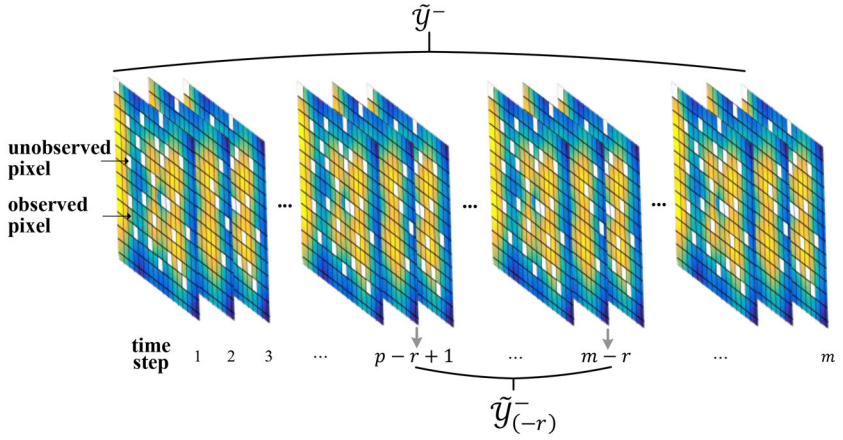

Figure 3. Image response notation representation.

or equivalently $\text{vec}(\mathcal{E}) \sim N(0, \mathbf{\Sigma}_{\mathcal{E}})$, where $\text{vec}(\mathcal{R})$ is the vectorization of a tensor $\mathcal{R}$. It is assumed that $\mathbf{\Sigma}_{\mathcal{E}} = \mathbf{\Sigma}_d \otimes \cdots \otimes \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_b$, where $\mathbf{\Sigma}_k (k = 1, \ldots, d)$ represents the spatial correlation of the noise, which are assumed to be defined by $\mathbf{\Sigma}_{k|i_1,i_2} = \exp\left(-\varpi \|r_{i_1} - r_{i_2}\|^2\right), k = 1, \ldots, d$ with the bandwidth ϖ controlling the strength of the spatial correlation and $\left(r_{i_1} - r_{i_2}\right)$ measuring the distance between data points i_1 and i_2; $\mathbf{\Sigma}_b$ represents the between-sample (temporal) covariance. Here, ϖ can be estimated using cross validation. The learning of $\mathbf{\Sigma}_b$ can refer to Algorithm 2 in Zhong, Paynabar, and Shi (2023).

However, estimating parameters in such a HD setting using traditional ARX estimation methods is impossible due to the overfitting issue and its computational and storage complexities. Moreover, the missing information also hinders the accuracy of parameter estimation. In order to handle these challenges and consider the spatial-temporal correlation among HD responses, we assume that the response data and model coefficients are low rank (Zhong, Paynabar, and Shi 2023). Thus, we can take advantage of the low-rank structure by representing the model using a LD core tensor and a set of factorizing matrices as defined in Tucker decomposition (Kolda and Bader 2009). That is,

$$\mathcal{B} = \mathcal{C}_B \times_1 \mathbf{U}_B \times_2 \mathbf{V}_{B1} \times_3 \cdots \times_{d+1} \mathbf{V}_{Bd}, \tag{3}$$

$$\mathcal{A}_j = \mathcal{C}_j \times_1 \mathbf{U}_{j1} \times_2 \cdots \times_d \mathbf{U}_{jd} \times_{d+1} \mathbf{V}_{j1} \times_{d+2} \cdots \times_{2d} \mathbf{V}_{jd}, \quad j = 1, \ldots, p, \tag{4}$$

where $\mathcal{C}_B \in \mathbb{R}^{P \times \tilde{Q}_1 \times \cdots \times \tilde{Q}_d}$ and $\mathcal{C}_j \in \mathbb{R}^{\tilde{Q}_1 \times \cdots \times \tilde{Q}_d \times \tilde{Q}_1 \times \cdots \times \tilde{Q}_d}$ are core tensors with $\tilde{Q}_i \ll Q_i (i = 1, \ldots, d)$. $\mathbf{U}_{ji} \in \mathbb{R}^{Q_i \times \tilde{Q}_i}$ $\left(j = 1, \ldots, p; i = 1, \ldots, d\right)$ and $\mathbf{U}_B \in \mathbb{R}^{P \times P}$ are basis (factorizing) matrices spanning the input space; and $\mathbf{V}_{ji} \in \mathbb{R}^{Q_i \times \tilde{Q}_i}$ $\left(j = 1, \ldots, p; i = 1, \ldots, d\right)$ and $\mathbf{V}_{Bi} \in \mathbb{R}^{Q_i \times \tilde{Q}_i}$ $(i = 1, \ldots, d)$ are basis matrices spanning the output space. To simplify the parameter estimation, we assume that $\mathbf{U}_B$ is an identity matrix. Besides, we set $\left\{\mathbf{U}_{1i} = \mathbf{U}_{2i} = \ldots = \mathbf{U}_{pi} = \mathbf{U}_i\right\}$ $(i = 1, \ldots, d)$ to capture the intercorrelation among the input data. Thanks to the flexibility of the core tensor $\left\{\mathcal{C}_j\right\}$ and basis matrices $\left\{\mathbf{V}_{ji}\right\}$, they could provide sufficient degrees to learn the high dimensional coefficients even if we set $\left\{\mathbf{U}_{ji}\right\}$ to be the same for all j (Zhong, Paynabar, and Shi 2023).

2.2. Problem Formulation

In the offline modeling stage, we propose the following loss function to complete the partially observed response tensor and estimate regression coefficients simultaneously:

$$\underset{\left\{\tilde{\mathcal{Y}}_{(-j)}\right\}, \mathcal{C}_j, \mathcal{C}_B, \{\mathbf{U}_i\}, \{\mathbf{V}_{ji}\}, \{\mathbf{V}_{Bi}\}, \mathbf{\Sigma}_{\mathcal{E}}}{\text{argmin}} \left\{ \gamma \sum_{j=0}^{p} \text{rank}\left(\tilde{\mathcal{Y}}_{(-j)}\right) + \text{vec}(\mathcal{H})^T \mathbf{\Sigma}_{\mathcal{E}}^{-1} \text{vec}(\mathcal{H}) \right\},$$

subject to

$$\begin{aligned}
&P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right), \\
&\mathbf{V}_{ji}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{ji} = \mathbf{I}_i, \mathbf{V}_{B_n i}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{B_n i} = \mathbf{I}_i, \\
&\qquad j = 1, \ldots, p; i = 1, \ldots, d, \\
&\mathcal{H} = \tilde{\mathcal{Y}}_{(0)} - \sum_{j=1}^{p} \tilde{\mathcal{Y}}_{(-j)} * \mathcal{A}_j - \mathbf{X}_{(-1)} * \mathcal{B}, \\
&\mathcal{A}_j = \mathcal{C}_j \times_1 \mathbf{U}_1 \times_2 \cdots \times_d \mathbf{U}_d \times_{d+1} \mathbf{V}_{j1} \times_{d+2} \cdots \times_{2d} \mathbf{V}_{jd}, \\
&\qquad j = 1, \ldots, p, \\
&\mathcal{B} = \mathcal{C}_B \times_1 \mathbf{U}_B \times_2 \mathbf{V}_{B1} \times_3 \cdots \times_{d+1} \mathbf{V}_{Bd},
\end{aligned} \tag{5}$$

where $\text{vec}(\mathcal{H}) = \text{vec}\left(\tilde{\mathbf{Y}}_{(0)}^{(1)}\right) - \sum_{j=1}^{p} \left(\left(\mathbf{V}_{jd} \otimes \cdots \otimes \mathbf{V}_{j1}\right) \otimes \mathbf{Z}_{(-j)}\right) \text{vec}\left(\mathbf{C}_j\right) - \left(\left(\mathbf{V}_{Bd} \otimes \cdots \otimes \mathbf{V}_{B1}\right) \otimes \mathbf{X}_{(-1)}\right) \text{vec}\left(\mathbf{C}_B\right)$. Specifically, in $\text{vec}(\mathcal{H})$, $\tilde{\mathbf{Y}}_{(0)}^{(1)}$ and $\tilde{\mathbf{Y}}_{(-j)}^{(1)}$ are the transpose of the mode-1 matricization of complete response tensors $\tilde{\mathcal{Y}}_{(0)}$ and $\tilde{\mathcal{Y}}_{(-j)}$, respectively; $\mathbf{Z}_{(-j)} = \tilde{\mathbf{Y}}_{(-j)}^{(1)} \left(\mathbf{U}_d \otimes \cdots \otimes \mathbf{U}_1\right)$; $\mathbf{C}_j \in \mathbb{R}^{\tilde{Q} \times \tilde{Q}}$ and $\mathbf{C}_B \in \mathbb{R}^{P \times \tilde{Q}}$ are the unfoldings of $\mathcal{C}_j$ and $\mathcal{C}_B$ with $\tilde{Q} = \prod_{i=1}^{d} \tilde{Q}_i$, respectively; and $\mathbf{I}_i$ is a $\tilde{Q}_i \times \tilde{Q}_i$ identity matrix.

Here, we apply the weighted constraint given by $\mathbf{V}_{ji}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{ji} = \mathbf{I}_i$, and $\mathbf{V}_{B_n i}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{B_n i} = \mathbf{I}_i$, which guarantees a similar spatial covariance structure for the estimated basis matrices and gives a closed-form solution. Besides, γ is a user-defined tuning parameter that can be selected using cross-validation.

2.3. Model Interpretation and Discussion

There are two terms in the objective function, that is, (5), where the first term exploits the low-rank structure of the output tensor for data completion and the second term minimizes the negative likelihood for parameter estimation. By combining these two terms, the data completion performance is improved through additional guidance provided by system dynamics embedded in the negative likelihood, which consequently improves parameter estimation accuracy.

Specifically, if $\gamma \to \infty$, (5) reduces to low-rank tensor completion algorithm (HALRTC) in Liu et al. (2013), a traditional tensor completion problem, that is

$$\underset{\left\{\tilde{\mathcal{Y}}_{(-j)}\right\}}{\operatorname{argmin}} \left\{ \gamma \sum_{j=0}^{p} \operatorname{rank}\left(\tilde{\mathcal{Y}}_{(-j)}\right) \right\},$$

subject to

$$P_{\Omega}\left(\tilde{\mathcal{Y}}\right) = P_{\Omega}\left(\tilde{\mathcal{Y}}^{-}\right). \tag{6}$$

However, Huang et al. (2015) shows that (6) accurately recovers missing entries only under a restricted relationship between missing ratio and rank, which limits its performance in more complex scenarios. On the contrary, the proposed method can boost the completion performance under a wider range of missing ratio and rank by incorporating system dynamics.

Furthermore, if $\gamma = 0$, (5) reduces to a regression problem with incomplete response:

$$\underset{\mathcal{C}_j,\mathcal{C}_B,\{\mathbf{U}_i\},\{\mathbf{V}_{ji}\},\{\mathbf{V}_{Bi}\},\boldsymbol{\Sigma}_{\mathcal{E}}}{\operatorname{argmin}} \left\{\operatorname{vec}(\mathcal{H})^T \boldsymbol{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}(\mathcal{H})\right\},$$

subject to

$$\begin{aligned}
&P_{\Omega}\left(\tilde{\mathcal{Y}}\right) = P_{\Omega}\left(\tilde{\mathcal{Y}}^{-}\right),\\
&\mathbf{V}_{ji}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{V}_{ji} = \mathbf{I}_i, \mathbf{V}_{B_n i}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{V}_{B_n i} = \mathbf{I}_i,\\
&\qquad j = 1,\ldots,p; i = 1,\ldots,d,\\
&\mathcal{H} = \tilde{\mathcal{Y}}_{(0)} - \sum_{j=1}^{p} \tilde{\mathcal{Y}}_{(-j)} * \mathcal{A}_j - \mathbf{X}_{(-1)} * \mathcal{B},\\
&\mathcal{A}_j = \mathcal{C}_j \times_1 \mathbf{U}_1 \times_2 \cdots \times_d \mathbf{U}_d \times_{d+1} \mathbf{V}_{j1} \times_{d+2} \cdots \times_{2d} \mathbf{V}_{jd},\\
&\qquad j = 1,\ldots,p,\\
&\mathcal{B} = \mathcal{C}_B \times_1 \mathbf{U}_B \times_2 \mathbf{V}_{B1} \times_3 \cdots \times_{d+1} \mathbf{V}_{Bd}.
\end{aligned} \tag{7}$$

It can be expected that the incomplete response will decrease the accuracy of parameter estimation and deteriorate the online control performance, especially under a high missing ratio. Therefore, it will be beneficial to incorporate the information of imputed entries in parameter estimation.

3. Offline Modeling

Solving (5) is challenging, since it involves both data imputation and model estimation. In this section, we propose an offline modeling algorithm to complete missing entries in the response and estimate the model coefficients.

Rather than adopting the NP-hard nonconvex rank penalty of a tensor $\mathcal{R}$, that is, $\operatorname{rank}(\mathcal{R})$, we use its convex relaxation (i.e., tensor nuclear norm) to make optimization tractable. The nuclear norm of a tensor $\mathcal{R}$, denoted by $\|\mathcal{R}\|_*$, is defined as the weighted average of nuclear norms of its matricizations along each mode (Wang et al. 2021), that is, $\|\mathcal{R}\|_* = \sum_{i=1}^{d} \alpha_i \|\mathbf{R}_{(i)}\|_*$, where $\mathbf{R}_{(i)} \in \mathbb{R}^{I_i \times I_{-i}} (I_{-i} = I_1 \times I_2 \cdots \times I_{i-1} \times I_{i+1} \times \cdots \times I_n)$ is the mode-i matricization of a tensor $\mathcal{R} \in \mathbb{R}^{I_1 \times \cdots \times I_i \times \cdots \times I_n}$, and $\|\mathbf{R}\|_* = \sum_j \lambda_j(\mathbf{R})$ is the nuclear norm of the matrix $\mathbf{R}$ with $\lambda_j(\mathbf{R})$ to be its corresponding jth largest singular value. Here, we assign equal weights to each of the tensor modes in the nuclear norm, that is, $\alpha_i = \frac{1}{d+1} (i = 1,\ldots,d+1)$. Let $\boldsymbol{\Theta}$ be the set of parameters including $\mathcal{C}_j, \mathcal{C}_B, \{\mathbf{U}_i\}, \{\mathbf{V}_{ji}\}, \{\mathbf{V}_{Bi}\}$, and $\boldsymbol{\Sigma}_{\mathcal{E}}$, where $j = 1,\ldots,p; i = 1,\ldots,d$. We can reformulate (5) as

$$\begin{aligned}
&\underset{\left\{\tilde{\mathcal{Y}}_{(-j)}\right\},\mathcal{C}_j,\mathcal{C}_B,\{\mathbf{U}_i\},\{\mathbf{V}_{ji}\},\{\mathbf{V}_{Bi}\},\boldsymbol{\Sigma}_{\mathcal{E}}}{\operatorname{argmin}}\\
&\left\{ \sum_{i=1}^{d+1} \sum_{j=0}^{p} \gamma \alpha_i \|\tilde{\mathbf{Y}}_{(-j)}^{(i)}\|_* + \operatorname{vec}(\mathcal{H})^T \boldsymbol{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}(\mathcal{H}) \right\},
\end{aligned}$$

subject to

$$\begin{aligned}
&P_{\Omega}\left(\tilde{\mathcal{Y}}\right) = P_{\Omega}\left(\tilde{\mathcal{Y}}^{-}\right),\\
&\mathbf{V}_{ji}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{V}_{ji} = \mathbf{I}_i, \mathbf{V}_{B_n i}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{V}_{B_n i} = \mathbf{I}_i,\\
&\qquad j = 1,\ldots,p; i = 1,\ldots,d,\\
&\mathcal{H} = \tilde{\mathcal{Y}}_{(0)} - \sum_{j=1}^{p} \tilde{\mathcal{Y}}_{(-j)} * \mathcal{A}_j - \mathbf{X}_{(-1)} * \mathcal{B},\\
&\mathcal{A}_j = \mathcal{C}_j \times_1 \mathbf{U}_1 \times_2 \cdots \times_d \mathbf{U}_d \times_{d+1} \mathbf{V}_{j1} \times_{d+2} \cdots \times_{2d} \mathbf{V}_{jd},\\
&\qquad j = 1,\ldots,p,\\
&\mathcal{B} = \mathcal{C}_B \times_1 \mathbf{U}_B \times_2 \mathbf{V}_{B1} \times_3 \cdots \times_{d+1} \mathbf{V}_{Bd}.
\end{aligned} \tag{8}$$

To realize offline modeling, we adopt the block coordinate descent (BCD) algorithm to update and iteratively as shown in Algorithm 1, in which the convergence is satisfied when (i) the difference of the objective function is less than the tolerance threshold or (ii) the algorithm reaches the maximal iteration number.

Algorithm 1 Offline Modeling Algorithm for Solving (5).

1: **Inputs**: $\tilde{\mathcal{Y}}^{-}$, $\mathbf{X}_{(-1)}$, and Ω.
2: Initialize $\tilde{\mathcal{Y}}$ by solving (6).
3: Initialize $\boldsymbol{\Theta}$ by solving (7) based on the initialized $\tilde{\mathcal{Y}}$ in Step 2.
4: **Loop**
5: ($\tilde{\mathcal{Y}}$-update) Given $\boldsymbol{\Theta}^k$, update $\tilde{\mathcal{Y}}^k$ to $\tilde{\mathcal{Y}}^{k+1}$.
6: ($\boldsymbol{\Theta}$-update) Given completed $\tilde{\mathcal{Y}}^{k+1}$, update $\boldsymbol{\Theta}^k$ to $\boldsymbol{\Theta}^{k+1}$.
7: Let $\tilde{\mathcal{Y}}^k = \tilde{\mathcal{Y}}^{k+1}$ and $\boldsymbol{\Theta}^k = \boldsymbol{\Theta}^{k+1}$.
8: **End** until convergence.

In the following sections, we will introduce the $\tilde{\mathcal{Y}}$-update (Step 5) and $\boldsymbol{\Theta}$-update (Step 6) procedures in detail. For notational convenience, we omit the iteration number k in the following discussion.

3.1. $\tilde{\mathcal{Y}}$-update

Assuming that $\mathbf{\Theta}^k$ is given at the $(k+1)$th iteration of the algorithm, we update $\tilde{\mathcal{Y}}$ by solving the following sub-problem:

$$\underset{\tilde{\mathcal{Y}}}{\operatorname{argmin}} \left\{ \operatorname{vec}(\mathbf{H}_s)^T \mathbf{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}(\mathbf{H}_s) + \sum_{i=1}^{d+1} \gamma\alpha_i \|\tilde{\mathbf{Y}}_{(i)}\|_* \right\},$$

subject to

$$P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right), \tag{9}$$

where $\mathbf{H}_s = \mathbf{S}_0\tilde{\mathbf{Y}}_{(1)} - \sum_{j=1}^{p} \mathbf{S}_j\tilde{\mathbf{Y}}_{(1)}\mathbf{A}_j - \mathbf{X}_{(-1)}\mathbf{B}_{(1)}$, $\mathbf{A}_j \in \mathbb{R}^{\tilde{Q}\times\tilde{Q}}$ is an unfolding of $\mathcal{A}_j$ with $\tilde{Q} = \prod_{i=1}^{d}\tilde{Q}_i$, $\mathbf{B}_{(1)}$ is mode-1 matricization of $\mathcal{B}$, $\tilde{\mathbf{Y}}^{(1)}_{(-j)} = \mathbf{S}_j\tilde{\mathbf{Y}}_{(1)}$ with the sampling matrix $\mathbf{S}_j \in \mathbb{R}^{(m-p)\times m}$ selecting response slides from $\tilde{\mathbf{Y}}_{(1)}$ as defined by

$$\mathbf{S}_j(i,r) = \begin{cases} 1, & \text{if } r = i+p-j, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Next, we use ADMM algorithm to solve (9). If we regard $\tilde{\mathcal{Y}}$ as a global variable, the problem becomes a consensus problem. To tackle the interdependent nuclear norm terms, we introduce auxiliary local tensors $\tilde{\mathcal{Y}}^i$, where $i = 1, \ldots, d+2$. Thus, we can decompose (9) into problems with independent local variables, that is

$$\underset{\left\{\tilde{\mathcal{Y}}^i\right\},\tilde{\mathcal{Y}}}{\operatorname{argmin}} \left\{ \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right)^T \Sigma_{\mathcal{E}}^{-1} \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right) + \sum_{i=1}^{d+1} \gamma\alpha_i \|\tilde{\mathbf{Y}}^i_{(i)}\|_* \right\},$$

subject to

$$P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right), \tilde{\mathcal{Y}} = \tilde{\mathcal{Y}}^i, i = 1, \ldots, d+2, \tag{11}$$

where $\mathbf{H}_s^{d+2} = \mathbf{S}_0\tilde{\mathbf{Y}}^{d+2}_{(1)} - \sum_{j=1}^{p}\mathbf{S}_j\tilde{\mathbf{Y}}^{d+2}_{(1)}\mathbf{A}_j - \mathbf{X}_{(-1)}\mathbf{B}_{(1)}$. To solve (11), we first address the equality constraints $\tilde{\mathcal{Y}} = \tilde{\mathcal{Y}}^i$ by defining the augmented Lagrangian as follows:

$$L_\rho\left(\left\{\tilde{\mathcal{Y}}^i\right\}, \tilde{\mathcal{Y}}, \{\mathcal{U}^i\}\right) = \left\{ \sum_{i=1}^{d+1} \gamma\alpha_i \|\tilde{\mathbf{Y}}^i_{(i)}\|_* + \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right)^T \times \mathbf{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right) + \sum_{i=1}^{d+2} \left(\langle \mathcal{U}^i, \tilde{\mathcal{Y}}^i - \tilde{\mathcal{Y}} \rangle + \frac{\rho}{2}\|\tilde{\mathcal{Y}}^i - \tilde{\mathcal{Y}}\|_F^2 \right) \right\}, \tag{12}$$

where $\{\mathcal{U}^i\}_{i=1,\ldots,d+2}$ are dual variables, ρ is the step size which can be chosen according to Section 3.4 in Boyd et al. (2011), $\langle\cdot,\cdot\rangle$ represents the inner product of tensors, and $\|\mathcal{R}\|_F^2$ is the Frobenius norm of a tensor $\mathcal{R}$, which is defined as the 2-norm of its vectorization, that is, $\|\mathcal{R}\|_F^2 = \|\operatorname{vec}(\mathcal{R})\|_2^2$. Correspondingly, the problem is represented as

$$\underset{\left\{\tilde{\mathcal{Y}}^i\right\},\tilde{\mathcal{Y}},\{\mathcal{U}^i\}}{\operatorname{argmin}} \; L_\rho\left(\left\{\tilde{\mathcal{Y}}^i\right\}, \tilde{\mathcal{Y}}, \{\mathcal{U}^i\}\right),$$

subject to

$$P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right). \tag{13}$$

Next, we will update local variables, global variables, and dual variables iteratively. To update the local variable, we solve the following two unconstrained problems:

$$\underset{\mathcal{Y}^i}{\operatorname{argmin}} \left\{ \gamma\alpha_i\|\tilde{\mathbf{Y}}^i_{(i)}\|_* + \langle \mathbf{u}^i_{(i)}, \tilde{\mathbf{Y}}^i_{(i)} - \tilde{\mathbf{Y}}_{(i)}\rangle + \frac{\rho}{2}\|\tilde{\mathbf{Y}}^i_{(i)} - \tilde{\mathbf{Y}}_{(i)}\|_F^2 \right\}, \quad i = 1, \ldots, d+1. \tag{14}$$

$$\underset{\tilde{\mathcal{Y}}^{d+2}}{\operatorname{argmin}} f\left(\tilde{\mathcal{Y}}^{d+2}\right) = \underset{\tilde{\mathcal{Y}}^{d+2}}{\operatorname{argmin}} \left\{ \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right)^T \mathbf{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}\left(\mathbf{H}_s^{d+2}\right) + \langle \mathbf{u}^{d+2}_{(1)}, \tilde{\mathbf{Y}}^{d+2}_{(1)} - \tilde{\mathbf{Y}}_{(1)}\rangle + \frac{\rho}{2}\|\tilde{\mathbf{Y}}^{d+2}_{(1)} - \tilde{\mathbf{Y}}_{(1)}\|_F^2 \right\}. \tag{15}$$

Equations (14) and (15) can be solved using Propositions 1 and 2, respectively.

Proposition 1. The closed-form solution of (14) is

$$\tilde{\mathbf{Y}}^i_{(i)} = \mathbf{U}_r\mathbf{\Lambda}_\lambda\mathbf{V}_r^T, \tag{16}$$

where $\mathbf{\Lambda}_\lambda$ is a diagonal matrix with $(\mathbf{\Lambda}_\lambda)_{ii} = \max\left(0, \Lambda_{\mathbf{p},ii} - \frac{\gamma\alpha_i}{1+\rho}\right)$, $\mathbf{U}_r$ and $\mathbf{V}_r$ are the first r columns of $\mathbf{U}_\mathbf{p}$ and $\mathbf{V}_\mathbf{p}$ from the SVD decomposition of $\mathbf{P}^i = \tilde{\mathbf{Y}}^i_{(i)} - \frac{1}{\rho}\mathbf{u}^i_{(i)}$, with $\mathbf{P}^i = \mathbf{U}_\mathbf{p}\mathbf{\Lambda}_\mathbf{p}\mathbf{V}_\mathbf{p}^T$, and $r = rank\left(\mathbf{P}^i\right)$.

The proof of Proposition 1 can be found in Appendix of supplementary materials.

Proposition 2. By setting the gradient of (15) to be zero, we can get the corresponding closed-form solution as follows:

$$\begin{aligned} \operatorname{vec}\left(\tilde{\mathbf{Y}}^{d+2}_{(1)}\right) = \Bigg(& \rho\mathbf{I} + 2\mathbf{\Sigma}_{d1} \otimes \left(\mathbf{S}_0^T\Sigma_s\mathbf{S}_0\right) \\ & -2\sum_{j=1}^{p}\begin{pmatrix} \left(\mathbf{\Sigma}_{d1}\mathbf{A}_j^T\right) \otimes \left(\mathbf{S}_0^T\mathbf{\Sigma}_s\mathbf{S}_j\right) + \\ \left(\mathbf{A}_j\mathbf{\Sigma}_{d1}\right) \otimes \left(\mathbf{S}_j^T\mathbf{\Sigma}_s\mathbf{S}_0\right) \end{pmatrix} \\ & + \sum_{j=1}^{p}\sum_{k=1}^{p}\begin{pmatrix} \left(\mathbf{A}_k\mathbf{\Sigma}_{d1}\mathbf{A}_j^T\right) \otimes \left(\mathbf{S}_k^T\mathbf{\Sigma}_s\mathbf{S}_j\right) + \\ \left(\mathbf{A}_j\mathbf{\Sigma}_{d1}\mathbf{A}_k^T\right) \otimes \left(\mathbf{S}_j^T\mathbf{\Sigma}_s\mathbf{S}_k\right) \end{pmatrix}\Bigg)^{-1} \\ & \times \operatorname{vec}\left(2\mathbf{S}_0^T\mathbf{\Sigma}_s\mathbf{X}_{(-1)}\mathbf{B}_{(1)}\mathbf{\Sigma}_{d1}\right. \\ & -2\sum_{j=1}^{p}\mathbf{S}_j^T\mathbf{\Sigma}_s\mathbf{X}_{(-1)}\mathbf{B}_{(1)}\mathbf{\Sigma}_{d1}\mathbf{A}_j^T \\ & \left. - \mathbf{u}^{d+2}_{(1)} + \rho\mathbf{Y}_{(1)}\right), \end{aligned} \tag{17}$$

where $\mathbf{\Sigma}_{d1} = \mathbf{\Sigma}_d^{-1} \otimes \cdots \otimes \mathbf{\Sigma}_2^{-1} \otimes \mathbf{\Sigma}_1^{-1}$, $\mathbf{\Sigma}_s = \mathbf{\Sigma}_b^{-1}$, and $\mathbf{I}$ is an $(mQ_1 \ldots Q_d) \times (mQ_1 \ldots Q_d)$ identity matrix.

The proof of Proposition 2 is provided in Appendix II of supplementary materials.

Although (17) provides a closed-form solution, it could only be used in small-scale problems when the computation resource is limited since it will result in computation challenges due to multiple Kronecker products and matrix inversion. To address this issue, we can use stochastic gradient descent algorithm

or its variants (e.g., Nesterov's accelerated gradient descent (AGD) algorithm) to solve (15) for large-scale problems and summarize details in Algorithm 2, where $\{\mathbf{D}^{(l)}\}$ are auxiliary variables at the lth iteration and ϵ is the user-defined error tolerance (Nesterov 1983).

Algorithm 2 Alternative Solution (for large-scale problems) to (15).

Inputs: $\mathbf{\Theta}^k$, $\tilde{\mathbf{Y}}_{(1)}$, $\mathbf{X}_{(-1)}$, $\{\mathbf{S}_j\}$, ρ, and Ω.
Initialize $\tilde{\mathbf{Y}}_{(1)}^{d+2^{(0)}} = \mathbf{D}^{(0)} = \tilde{\mathbf{Y}}_{(1)}$.
While $\|\nabla f\left(\mathbf{D}^{(l)}\right)\|_F \geq \varepsilon$:
  $\tilde{\mathbf{Y}}_{(1)}^{d+2^{(l+1)}} = \mathbf{D}^{(l)} - \mu\nabla f\left(\mathbf{D}^{(l)}\right)$.
  $\mathbf{D}^{(l+1)} = \tilde{\mathbf{Y}}_{(1)}^{d+2^{(l+1)}} + \frac{l-1}{l+2}\left(\tilde{\mathbf{Y}}_{(1)}^{d+2^{(l+1)}} - \tilde{\mathbf{Y}}_{(1)}^{d+2^{(l)}}\right)$.
  $l \leftarrow l+1$.
End while.
Return $\tilde{\mathbf{Y}}_{(1)}^{d+2^{(l)}}$.

To update the global variable, the following constrained optimization problem is formulated:

$$\underset{\tilde{\mathcal{Y}}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{d+2} \left(\langle \mathcal{U}^i, \tilde{\mathcal{Y}}^i - \tilde{\mathcal{Y}} \rangle + \frac{\rho}{2} \|\tilde{\mathcal{Y}}^i - \tilde{\mathcal{Y}}\|_F^2 \right) \right\},$$

subject to

$$P_\Omega\left(\tilde{\mathcal{Y}}\right) = P_\Omega\left(\tilde{\mathcal{Y}}^-\right). \tag{18}$$

According to Wang et al. (2021), Equation (18) can be solved by

$$\tilde{\mathcal{Y}} = \begin{cases} \tilde{\mathcal{Y}}^-, & \text{if } (i, q_1, \ldots, q_d) \in \Omega, \\ \frac{1}{d+2}\sum_{i=1}^{d+2}\left(\tilde{\mathcal{Y}}^i + \frac{1}{\rho}\mathcal{U}^i\right), & \text{otherwise.} \end{cases} \tag{19}$$

Finally, we update the dual variable via $\mathcal{U}^i \leftarrow \mathcal{U}^i + \rho\left(\tilde{\mathcal{Y}}^i - \tilde{\mathcal{Y}}\right), i = 1, \ldots, d+2$.

The details of $\tilde{\mathcal{Y}}$-update are shown in Algorithm 3, where (t) represents the iteration number. The stopping criteria follow Algorithm 1.

Algorithm 3 Response Completion Algorithm by Solving (9).

Inputs: $\mathbf{\Theta}^k$, $\tilde{\mathcal{Y}}^-$, $\mathbf{X}_{(-1)}$, $\{\mathbf{S}_j\}$, ρ, and Ω.
Loop
  Update local variable by solving (14) and (15): $\tilde{\mathcal{Y}}^{i^{(t+1)}} \leftarrow \tilde{\mathcal{Y}}^{i^{(t)}}$.
  Update global variable using (19): $\tilde{\mathcal{Y}}^{(t+1)} \leftarrow \tilde{\mathcal{Y}}^{(t)}$.
  Update dual variable: $\mathcal{U}^{i^{(t+1)}} \leftarrow \mathcal{U}^{i^{(t)}} + \rho\left(\tilde{\mathcal{Y}}^{i^{(t)}} - \tilde{\mathcal{Y}}^{(t)}\right)$.
  Let $\tilde{\mathcal{Y}}^{i^{(t)}} = \tilde{\mathcal{Y}}^{i^{(t+1)}}$, $\tilde{\mathcal{Y}}^{(t)} = \tilde{\mathcal{Y}}^{(t+1)}$, and $\mathcal{U}^{i^{(t)}} = \mathcal{U}^{i^{(t+1)}}$.
End until convergence.

3.2. Θ-update

When $\tilde{\mathcal{Y}}^{k+1}$ is updated from $\tilde{\mathcal{Y}}^k$, $\mathbf{\Theta}^k$ can be updated by solving the following sub-problem:

$$\underset{\mathbf{\Theta}}{\operatorname{argmin}} \left\{ \operatorname{vec}(\mathcal{H})^T \mathbf{\Sigma}_{\mathcal{E}}^{-1} \operatorname{vec}(\mathcal{H}) \right\}$$

$$\text{subject to } \mathbf{V}_{ji}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{ji} = \mathbf{I}, \mathbf{V}_{Bi}^T \mathbf{\Sigma}_i^{-1} \mathbf{V}_{Bi} = \mathbf{I}. \tag{20}$$

To efficiently optimize (20), following Zhong, Paynabar, and Shi (2023), we adopt the alternating least square with block coordinate descent (ALS-BCD) method to update $\mathbf{\Theta}$ iteratively as shown in Algorithm 4, where ε is the error tolerance. The details are summarized in Appendix III of supplementary materials, in which the stopping criteria follow Algorithm 1.

Algorithm 4 Parameter Estimation Algorithm for Solving (20).

Inputs: $\mathbf{\Theta}$ (initialization) and $\tilde{\mathcal{Y}}$.
Estimate $\mathbf{\Sigma}_{\mathcal{E}}$ (Part 1 of Appendix III).
Estimate $\{\mathbf{U}_i\}$ using Tucker decomposition (Part 2 of Appendix III).
Calculate the initial value of the likelihood term in (20).
Loop
  Loop
    Estimate $\{\mathbf{V}_{ji}\}$ and $\{\mathbf{V}_{Bi}\}$ iteratively (Part 3 of Appendix III).
  End until convergence.
  Estimate $\{\mathcal{C}_j\}$ and $\mathcal{C}_B$ (Part 4 of Appendix III).
End until convergence.

Figure 4 summarizes the parameter estimation and response completion procedure used for offline modeling. Next, we present the predictive control model.

4. One-Step-Ahead Predictive Control

Given estimated parameters $\hat{\mathbf{\Theta}}$ and completed historical responses $\left\{\mathcal{Y}_{t-j}^{cp}\right\}_{j=1,\ldots,p}$ obtained from the offline modeling stage, the optimal control can be achieved at time step $t-1$ by minimizing the expected difference between the one-step-ahead predicted response and the target value, which can be formulated as

$$\psi(\mathbf{X}_{t-1}) = \min_{\mathbf{X}_{t-1}} E\|\hat{\mathcal{Y}}_{t|\{t-j\}}(\mathbf{X}_{t-1}) - \mathcal{Y}_t^{\text{target}}\|_F^2 \tag{21}$$

where $\mathcal{Y}_t^{\text{target}}$ is the target tensor, $E(\cdot)$ is the expectation operator, and $\hat{\mathcal{Y}}_{t|\{t-j\}}(\mathbf{X}_{t-1})\,(j = 1, \ldots, p)$ is the one-step-ahead predicted tensor at time step $t-1$ based on imputed historical responses $\left\{\mathcal{Y}_{t-j}^{cp}\right\}_{j=1,\ldots,p}$ and estimated model coefficients $\hat{\mathcal{A}}_j$ and $\hat{\mathcal{B}}$, and the control action $\mathbf{X}_{t-1}$, that is,

$$\hat{\mathcal{Y}}_{t|\{t-j\}}(\mathbf{X}_{t-1}) = \Sigma_{j=1}^p \hat{\mathcal{A}}_j * \mathcal{Y}_{t-j}^{cp} + \hat{\mathcal{B}} * \mathbf{X}_{t-1}, \tag{22}$$

The optimal control law is given by the closed-form solution of (21):

$$\mathbf{X}_{t-1} = \mathbf{C}_B^{-1}(\mathbf{V}_{Bd} \otimes \cdots \otimes \mathbf{V}_{B1})^T \times \operatorname{vec}\left(\mathcal{Y}_t^{\text{target}} - \sum_{j=1}^p \mathcal{Y}_{t-j}^{cp} * \mathcal{A}_j\right), \tag{23}$$

where $\mathbf{C}_B \in \mathbb{R}^{\tilde{P}\times\tilde{Q}}$ is the folded core tensor $\mathcal{C}_{\mathcal{B}}$ with $\tilde{Q} = \prod_{i=1}^d \tilde{Q}_i$. The detailed derivations can be found in Appendix IV of supplementary materials.

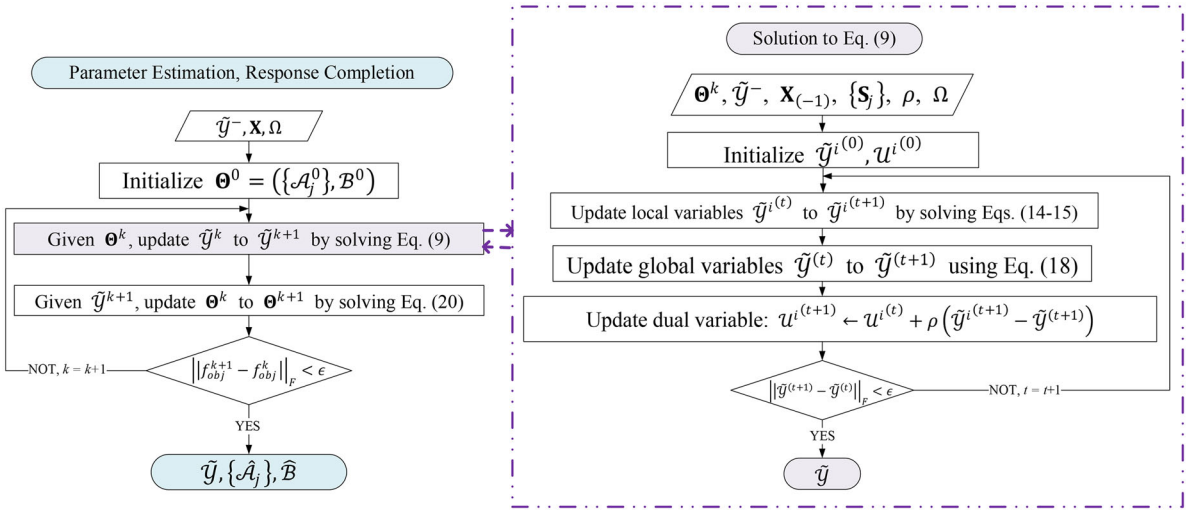

Figure 4. Flowchart of offline modeling algorithm.

Next, when a new observation $\mathcal{Y}_t^-$ is available, we need to impute its missing entries before moving to the next time step. Intuitively, we can just fill in the one-step-ahead prediction result in the missing entries of new observations, that is

$$\mathcal{Y}_t^{cp} = \begin{cases} \mathcal{Y}_t^-, & \text{if } (i, q_1, \ldots, q_d) \in \Omega_t, \\ \hat{\mathcal{Y}}_{t|\{t-j\}}\left(\mathbf{X}_{t-1}\right), & \text{otherwise.} \end{cases} \tag{24}$$

This strategy is easy to implement when the online computation resource is limited. However, (24) highly depends on the model trained offline, while ignoring the low-rank structure of the response, which may mislead the online completion when the system model is inaccurately estimated. To achieve a better performance, we propose the following low-rank prediction-oriented completion (LRPOC) strategy to impute the incomplete entries:

$$\begin{aligned} &\min_{\mathcal{Y}_t^{cp}} \left\{ \mu \|\mathcal{Y}_t^{cp}\|_* + \frac{1}{2}\|\mathcal{Y}_t^{cp} - \hat{\mathcal{Y}}_{t|\{t-j\}}\left(\mathbf{X}_{t-1}\right)\|_F^2 \right\} \\ &\text{subject to } P_{\Omega_t}\left(\mathcal{Y}_t^{cp}\right) = P_{\Omega_t}\left(\mathcal{Y}_t^-\right), \end{aligned} \tag{25}$$

where μ is a user-defined tuning parameter that can be selected using cross-validation.

Equation (25) has two-fold benefits for online control: (i) it makes use of the most of available information, that is, the true observations and the impact from control actions; and (ii) it explores the low-rank structure of the output response. For notational convenience, we call (24) simplified LRPOC (sLRPOC) strategy. Following a similar approach in solving (9), we apply ADMM to solve the problem by introducing auxiliary variables to reformulate (25) to be a global consensus problem:

$$\begin{aligned} \min_{\left\{\mathbf{Y}_{t(i)}^{cp}\right\}_{i=1,\ldots,d+1}} &\left\{ \sum_{i=1}^{d+1} \alpha_i \left(\mu \|\mathbf{Y}_{t(i)}^{cp}\|_* + \frac{1}{2}\|\mathbf{Y}_{t(i)}^{cp} \right.\right. \\ &\left.\left. - \left[\hat{\mathbf{Y}}_{t|\{t-j\}}\left(\mathbf{X}_{t-1}\right)\right]_{(i)} \|_F^2 \right) \right\} \end{aligned}$$

$$\begin{aligned} &\text{subject to } P_{\Omega_t}\left(\mathcal{Y}_t^{cp}\right) = P_{\Omega_t}\left(\mathcal{Y}_t^-\right), \mathcal{Y}_t^{cp} = \mathcal{Y}_t^i, \\ &\qquad i = 1, \ldots, d+1, \end{aligned} \tag{26}$$

where $\mathbf{Y}_{t(i)}^i$ represents the mode-i matricization of local tensors $\mathcal{Y}_t^i, i = 1, \ldots, d+1$. We first address the equality constraints $\mathcal{Y}_t^{cp} = \mathcal{Y}_t^i, i = 1, \ldots, d+1$ by defining the augmented Lagrangian as follows:

$$\begin{aligned} L_\tau\left(\mathcal{Y}_t^{cp}, \left\{\mathbf{Y}_{t(i)}^i\right\}, \left\{\mathcal{F}_t^i\right\}\right) = \sum_{i=1}^{d+1} \Big\{ &\mu\alpha_i \|\mathbf{Y}_{t(i)}^i\|_* + \frac{\alpha_i}{2}\|\mathbf{Y}_{t(i)}^i \\ &- \left[\hat{\mathbf{Y}}_{t|\{t-j\}}\left(\mathbf{X}_{t-1}\right)\right]_{(i)} \|_F^2 + \langle \mathcal{F}_t^i, \mathcal{Y}_t^{cp} - \mathcal{Y}_t^i \rangle + \frac{\tau}{2}\|\mathcal{Y}_t^{cp} - \mathcal{Y}_t^i\|_F^2 \Big\} \end{aligned} \tag{27}$$

where $\left\{\mathcal{F}_t^i\right\}$ denote dual variables, and τ is the step size. Then (25) is represented as

$$\begin{aligned} &\min_{\mathcal{Y}_t^{cp}, \left\{\mathbf{Y}_{t(i)}^i\right\}, \left\{\mathcal{F}_t^i\right\}} L_\tau\left(\mathcal{Y}_t^{cp}, \left\{\mathbf{Y}_{t(i)}^i\right\}, \left\{\mathcal{F}_t^i\right\}\right) \\ &\text{subject to } P_\Omega\left(\mathcal{Y}_t^{cp}\right) = P_\Omega\left(\mathcal{Y}_t^-\right). \end{aligned} \tag{28}$$

The solution details can be found in Appendix V of supplementary materials.

Based on the one-step-ahead prediction and online completion strategy, the one-step-ahead predictive control algorithm is summarized in Algorithm 5.

Algorithm 5 Online Control Algorithm.

1: **Inputs**: $\hat{\mathbf{\Theta}}$, $\left\{\mathcal{Y}_{t-j}^{cp}\right\}_{j=1,\ldots,p}$, and $\mathcal{Y}_t^{\text{target}}$.
2: Execute control action according to (23).
3: Calculate the one-step-ahead prediction based on (22).
4: Collect new observation $\mathcal{Y}_t^-$, and initialize $\mathcal{Y}_t^{cp^{(0)}} = \mathcal{Y}_t^-$.
5: Impute $\mathcal{Y}_t^-$ to get the recovered $\mathcal{Y}_t^{cp}$ by solving (25).

5. Performance Evaluation via Simulations

In this section, we conduct simulation studies to validate the proposed partially observed tensor-based automatic control method.

5.1. Data Generation

Following Zhong, Paynabar, and Shi (2023), we include two types of responses in our simulation study: (i) wave-shape surface control, and (ii) truncated cylinder control. For each case, we generate a complete data series, and then, select partially observed series of length N_{tr} and N_{te} to be the training and test data, respectively.

Case 1: Wave-shape surface point cloud simulation

Assuming $l = 1$ and $p = 2$, a sequence of waveform surface responses in a 3D Cartesian coordinate system is generated. Specifically, as shown in Figure 5(a), for the t-th sample, an $I_1 \times I_2$ matrix is simulated using the following model:

$$y_t = y_{t-1} * \mathcal{A}_1 + y_{t-2} * \mathcal{A}_2 + \mathcal{C}_B \times_2 V^1 \times_3 V^2 \times_1 \mathbf{X}_{t-1} + \mathcal{E}_t, \tag{29}$$

where $\times_i$ is the mode-i tensor product between a tensor $\mathcal{R}_1 \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ by a matrix $\mathbf{R}_2 \in \mathbb{R}^{M \times I_i}$, that is, $\mathcal{R}_1 \times_i \mathbf{R}_2 \in \mathbb{R}^{I_1 \times \cdots \times I_{i-1} \times M \times I_{i+1} \times \cdots \times I_n}$, matrix y_t includes the height information at the location $\left(\frac{i_1}{I_1}, \frac{i_2}{I_2}\right)$ $(i_1 = 1, \ldots, I_1; i_2 = 1, \ldots, I_2)$, $\mathcal{A}_1 = \mathcal{C}_1 \times_1 \mathbf{V}^1 \times_2 \mathbf{V}^2 \times_3 \mathbf{V}^1 \times_4 \mathbf{V}^2$, $\mathcal{A}_2 = \mathcal{C}_2 \times_1 \mathbf{V}^1 \times_2 \mathbf{V}^2 \times_3 \mathbf{V}^1 \times_4 \mathbf{V}^2$, $\mathcal{C}_B \in \mathbb{R}^{3\times3\times2}$ is randomly generated from a normal distribution $N(0.3, 0.5)$, and basis matrices $\mathbf{V}^{(k)} = \left[\boldsymbol{v}_1^{(k)}, \boldsymbol{v}_2^{(k)}, \boldsymbol{v}_3^{(k)}\right]$ with $\boldsymbol{v}_\alpha^{(k)} = \left[\sin\left(\frac{\pi\alpha}{I_k}\right), \sin\left(\frac{2\pi\alpha}{I_k}\right), \ldots, \sin\left(\frac{I_k\pi\alpha}{I_k}\right)\right]^T$ $(k = 1, 2; \alpha = 1, 2, 3)$. The elements of input matrices $\mathbf{X}_t \in \mathbb{R}^{4\times1}(t = 1, \ldots, N_{tr}$ or $N_{te})$ are randomly sampled from the standard normal distribution $N(0, 1)$. The noise is generated from the tensor normal distribution $\mathcal{N}(0, \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_b)$, where $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$.

Case 2: Truncated cone point cloud simulation

We simulate a truncated cone point cloud based on a set of scalars and simple profile data in a three-dimensional cylindrical coordinate system (r, ϕ, z), where $\phi \in [0, 2\pi]$, and $z \in [0, 1]$. An equidistant grid of size $I_1 \times I_2$ over the (ϕ, z) space is generated by setting $\phi_i = \frac{2\pi i}{I_1}(i = 1, \ldots, I_1)$ and $z_j = \frac{j}{I_2}(j = 1, \ldots, I_2)$. We simulate the mean patterns of the point cloud surface S_t such that $r(\phi, z) = 1$ for any pair of (ϕ, z). Next, the variational pattern is generated by the following tensor time-series sequence:

$$S_t = S_{t-1} * \mathcal{A}_1 + S_{t-2} * \mathcal{A}_2 + \mathcal{C}_B \times_2 \mathbf{V}^1 \times_3 \mathbf{V}^2 \times_1 \mathbf{X}_{t-1} + \mathcal{E}_t, \tag{30}$$

where $S_t \in \mathbb{R}^{I_1 \times I_2}$ represents the variational pattern at time t, and $\mathbf{X}_t \in \mathbb{R}^{4\times1}$ is a control vector. $\mathcal{C}_B$ is generated from a normal distribution $N(0.3, 0.5)$. We also generate $\mathcal{A}_1, \mathcal{A}_2, \mathbf{V}^1, \mathbf{V}^2$, and the noise term in a similar way to Case 1. Examples of truncated cone point cloud are given in Figure 5(b).

The target response for Case 1 is zero, while its Frobenius norm is 6.64 for Case 2. For both cases, we generate $N_{tr} = 50$ samples as training data and $N_{te} = 40$ as test data.

5.2. Benchmark Methods in Comparative Study

In Study 1, we analyze the completion efficiency of the proposed poTAC method and compare it with the HALRTC algorithm (Liu et al. 2013) for low-rank tensor completion.

As for Study 2, an intuitive two-step CT method is chosen as a benchmark for comparison, in which the response completion (C) is first conducted by using the HALRTC algorithm (Liu et al. 2013), and then the regression model is established based on the TTS model (T) in Zhong, Paynabar, and Shi (2023).

By combining different strategies for either offline modeling or online completion, we will test the performance of four methods as summarized in Table 1. Specifically, they are: (i) poTAC (offline modeling using Algorithm 1, and online completion using LRPOC), (ii) CTC (CT followed by online completion still using HALRTC), (iii) spTAC (simplified poTAC, i.e., offline modeling using Algorithm 1, and online completion using sLRPOC), and (iv) CTLRC (CT followed by online completion still using sLRPOC). From Table 1, we find that poTAC and spTAC share the same proposed one-step offline training method, while spTAC uses the simplified version of the proposed online control algorithm. SpTAC and CTLRC employ the same online control algorithm, but CTLRC first completes the response using HALRTC and then builds the regression using TTS, which is a two-step method different from the proposed one-step method. As for CTLRC and CTC, they use the same two-step offline training algorithm, but CTC only considers the low-rank structure to complete the response while CTLRC imputes the response

Table 1. The methods in the comparative study.

Methods	Offline modeling	Online completion
poTAC	Algorithm 1	LRPOC
spTAC	(one-step method)	sLRPOC
CTLRC	HALRTC followed by TTS	
CTC	(two-step method)	HALRTC

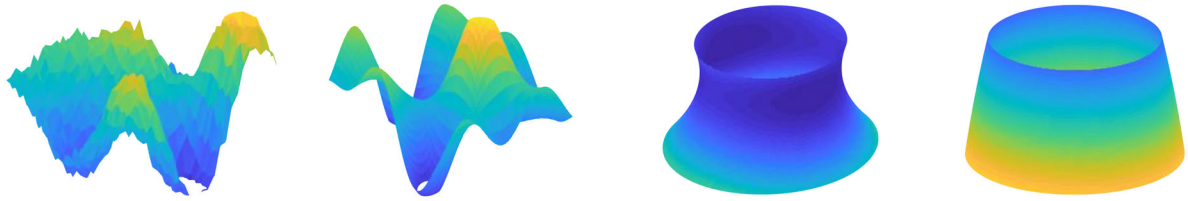

Figure 5. Examples of generated data in simulations. (a) Wave-shape surface point cloud. (b) Truncated cone point cloud.

considering the deviation of prediction from the target. Note that the same control law, that is, (24), is executed to get a new response.

5.3. Simulation Results

Study 1: Offline Completion Performance Evaluation

We generate data according to the described procedure with $I_1 = 20$ and $I_2 = 12$. In order to demonstrate the applicability of the proposed method, we test its performance under different settings including (i) five random missing ratios, that is, 50%, 60%, 70%, 80%, and 90%; (ii) four levels of signal-to-noise ratios (SNR, defined by $\sum_{i=1}^{N_{tr}} \|\mathcal{Y}_{tr,i}\|_F^2 / \sum_{i=1}^{N_{tr}} \|E_{tr,i}\|_F^2$), that is, 10^{13}, 10^{11}, 10^9, and 10^7; and (iii) three Tucker rank tuples, that is, 3, 6, and 9 for all modes, respectively.

Since this is a small-scale problem, we use the closed-form solution, that is, (16), for offline modeling. In this Study, we evaluate the offline performance using the mean squared completion error (MSCE), that is, $\frac{1}{N_{tr}} \sum_{t=1}^{N_{tr}} \|\mathcal{Y}_t^{cp} - \mathcal{Y}_t\|_F^2$. The simulation results are summarized in Appendix VI of supplementary materials, in which 30 experiments are replicated for each setting. From Appendix VI, we can find that the SNR level has a limited effect on the completion performance. Both poTAC and HALRTC have robust performance with respect to SNR values under considered SNR levels. Therefore, to have a clearer illustration of relationships among different settings, Figure 6 shows only the results for the highest SNR level.

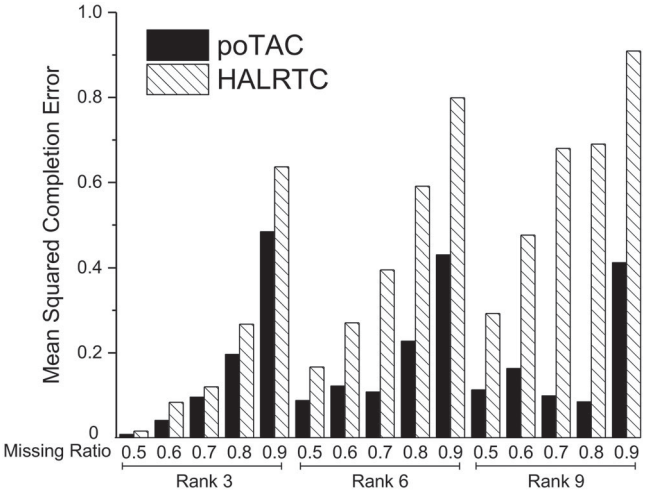

Figure 6. MSCE results for offline completion given true system model with SNR≈ 10^{13}.

From Figure 6, we can find that poTAC outperforms HALRTC in all considered scenarios. It is also evident that under the same missing ratio, unlike HALRTC, the performance of poTAC is not sensitive to the rank. Regardless of variations in the missing ratio and response rank, the completion performance of poTAC is relatively robust, while the completion error of HALRTC increases drastically under either higher response rank or higher missing ratio. These simulation results indicate that the completion performance benefits from the additional information provided by the ARX structure.

Study 2: Online Control Performance Evaluation

For Study 2, we generate observations of size $I_1 = 30$ by $I_2 = 20$ and randomly remove 80%, 85%, and 90% of training data. Under such a setting, the computation involves Kronecker product of size $30,000 \otimes 30,000$ if we use the closed-form solution of (15) in offline modeling, which is computationally intractable. Thus, we use the gradient descent algorithm to solve (15). Based on the trained model, we execute the one-step-ahead predictive control and impute the new observations using the proposed LRPOC and its simplified version, that is, sLRPOC. In this study, we consider three SNR levels for each setting and replicate it 30 times to evaluate the average performance with respect to the steady state relative mean squared deviation (RMSD) from the target, defined by $\sum_{i=1}^{N_c} \|\mathcal{Y}_i^{ct} - \mathcal{Y}^{\text{target}}\|_F^2 / \sum_{i=1}^{N_c} \|\mathcal{Y}_i^{wct} - \mathcal{Y}^{\text{target}}\|_F^2$ with

$$\mathcal{Y}_i^{ct} = \Sigma_{j=1}^{p} \mathcal{Y}_{t-j}^{ct} * \mathcal{A}_j + \mathbf{X}_{t-1} * \mathcal{B} + E_t, \tag{31}$$

$$\mathcal{Y}_i^{wct} = \Sigma_{j=1}^{p} \mathcal{Y}_{t-j}^{wct} * \mathcal{A}_j + E_t, \tag{32}$$

where $\mathcal{Y}_i^{ct}$ is the response with control, and $\mathcal{Y}_i^{wct}$ is the one without control. The RMSD results and their variance (as shown in brackets) are reported in Table II (Case 1) and Table III (Case 2) in Appendix VII of supplementary materials.

We also visualize all the RMSD results in Figure 7 (Case 1) and Figure 8 (Case 2). Note that CTC is not included in Figure 8 since its RMSD values are too large. From the figures, we can see

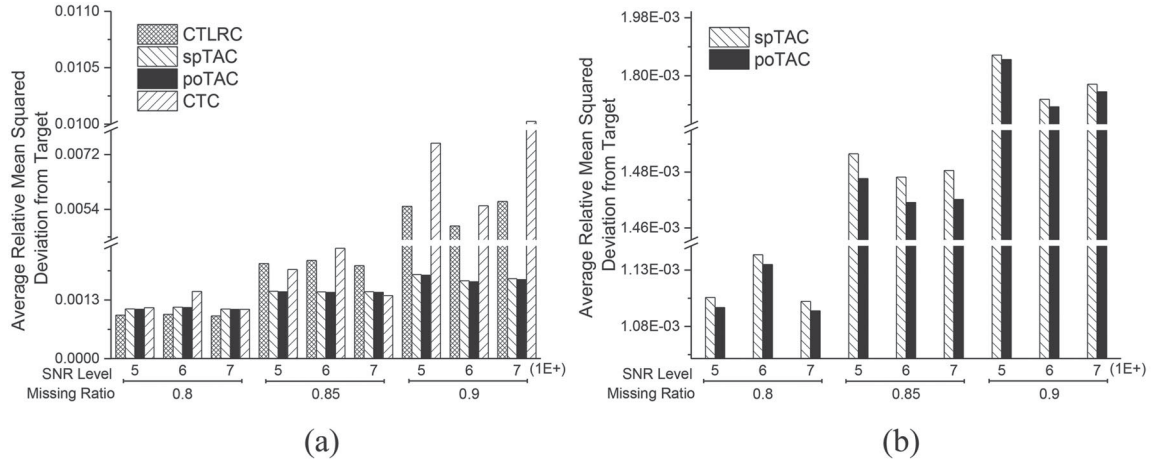

Figure 7. Online control performance for Case 1. (a) comparison among four methods (b) comparison between spTAC and poTAC.

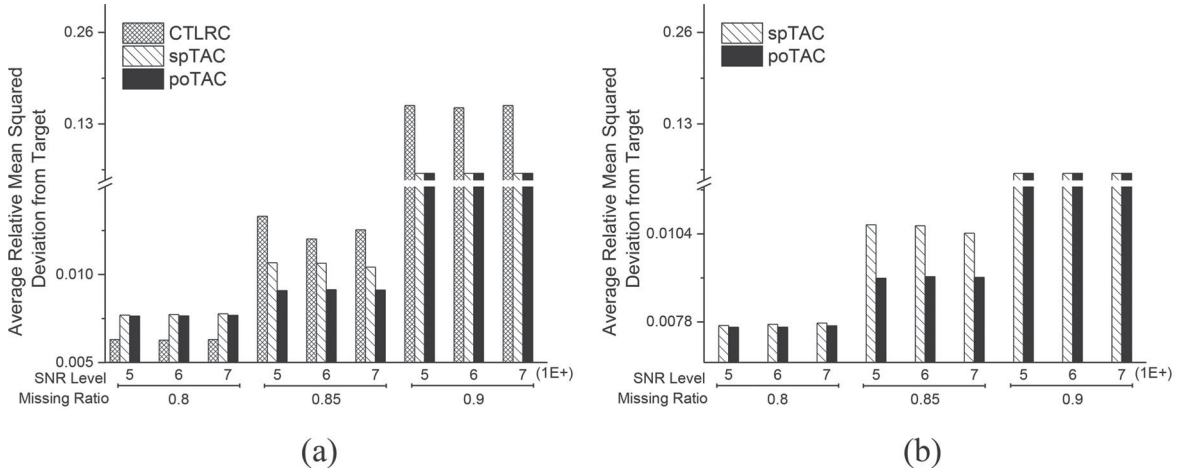

Figure 8. Online control performance for Case 2. (a) comparison among three methods (b) comparison between spTAC and poTAC.

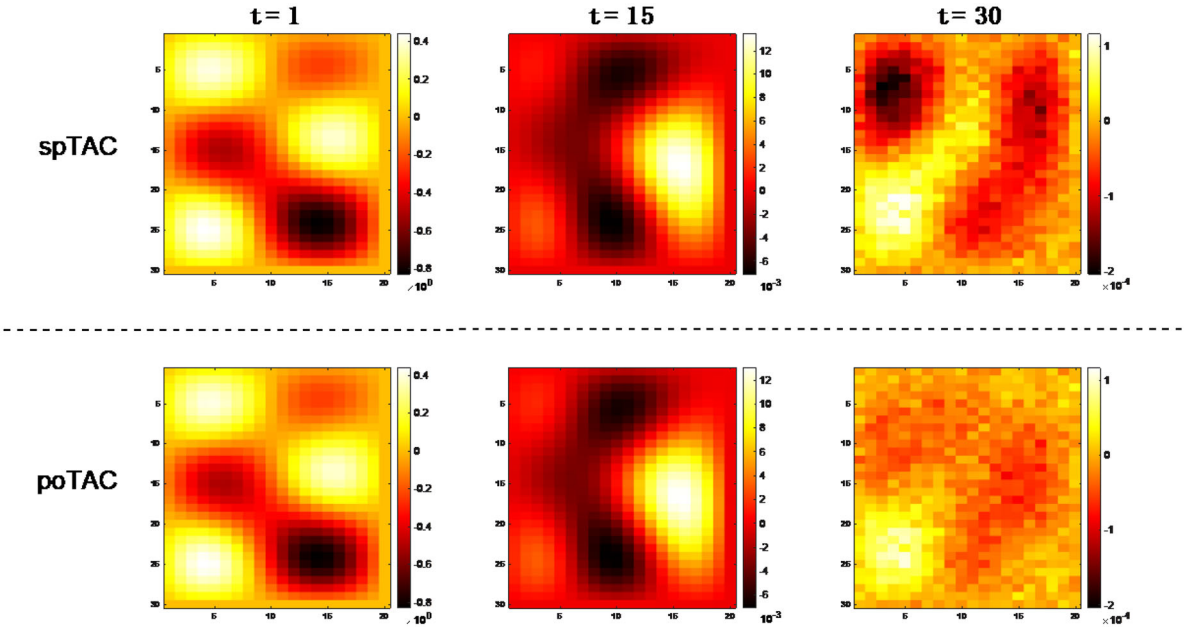

Figure 9. Comparison between spTAC and poTAC.

that the SNR level has greater impact on the control performance of CTLRC and CTC under the same missing ratio, while it has limited effect on spTAC and poTAC.

We can also evaluate the performance of our offline modeling approach by comparing spTAC and CTLRC based on the same online completion strategy (i.e., sLRPOC). As can be seen from Figures 7(a) and 8(a), in both cases, although CTLRC has slightly better control performance when missing ratio is low, spTAC clearly outperforms CTLRC in cases with high missing ratio (e.g., 85% and 90%). This is because the high rank or high missing ratio settings are beyond the capability of HALRTC, which validates our discussion on the impact of system dynamics in Section 2.3.

Regarding the online completion strategy, we can see that the proposed LRPOC is more effective than sLRPOC (HALRTC) by comparing poTAC and spTAC (CTLRC and CTC) under the same offline modeling setting. This is expected because the proposed LRPOC incorporates benefits from both sLRPOC (capturing the system dynamics) and HALRTC (considering the low-rank structure of the response). Having a closer look at Figures 7(b) and 8(b), we observe that sLRPOC achieves a competitive performance with LRPOC at some cases. Thus, sLRPOC is recommended under the limited computation resources.

To visualize the control performance of the proposed online control algorithm, we show the sample response of Case 1 under 80% missing ratio and the lowest SNR level at time $t = 1, 15, 30$ for poTAC and spTAC, in Figure 9. As time moves forward, the RMSD continually decrease and the process reaches a relative steady state with small deviation around the target (i.e., zero). Moreover, by comparing the images for $t = 30$, we can see that the proposed LRPOC strategy achieves better control result compared with its simplified version.

Additionally, to compare the control performance of the poTAC and CTC over time, the Frobenius norms of squared deviation from the target are shown in Figure 10. To have a fair comparison, the sLRPOC strategy is executed for both methods. From Figure 10, we can see that the proposed poTAC has less deviation from the target than the CTC method since it learns

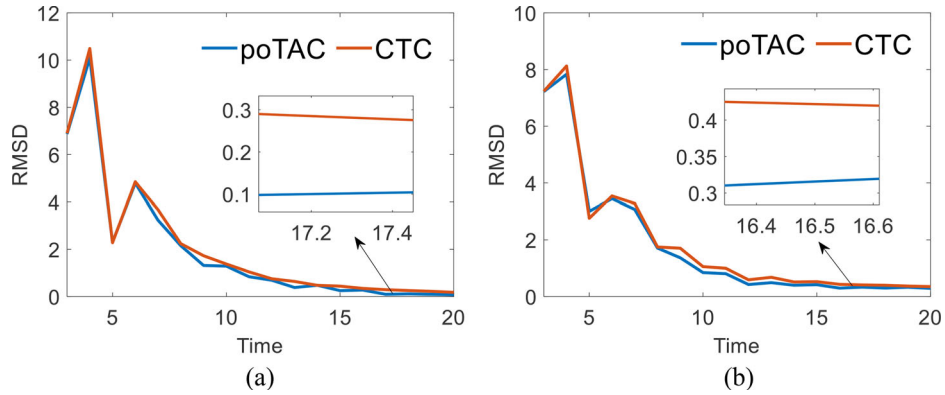

Figure 10. System performance deviation from target. (a) Case 1. (b) Case 2.

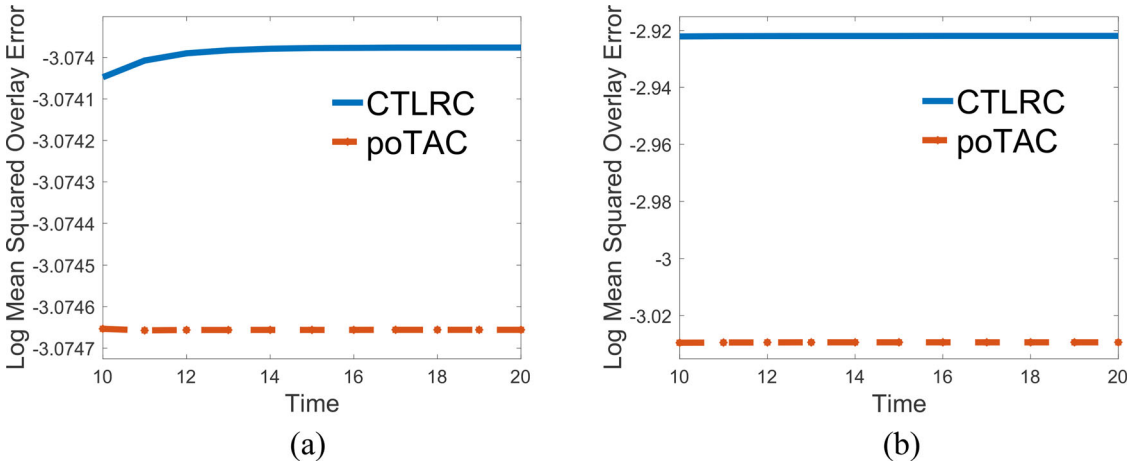

Figure 11. Log mean squared overlay error over time. (a) under 70% missing ratio (b) under 80% missing ratio.

more about system dynamics during the offline modeling stage, which results in a better tracking ability for the online control.

6. Case Study

Photolithography is an important process in semiconductor manufacturing, in which the overlay error is a critical quality characteristic that should be controlled online (Figure 1(b)). To model the overlay errors, we represent them as images, which includes all overlay errors from a single wafer. To control the wafer quality, we can control the important settings of the production machine, such as the wafer position and lens height.

In this case study, we generate the overlay data from a simulator endorsed by a well-known semiconductor company (Figure 1(a)). The detailed procedure of data generation can be found in Zhong, Paynabar, and Shi (2023). Using the simulator, we generate 50 training samples and 50 test samples to validate the control performance of the proposed poTAC method. According to the expert experience, we set $p = 1$ in our analysis. The basis ranks are set to be 7 and 2 using the AIC criterion. The target value in the online control model for the overlay error is set to zero. We first apply the proposed method to estimate model coefficients and restore the incomplete response. Next, we use the trained model for online control. Here, we randomly remove 70% and 80% of data in each observation, and each setting will be replicated for 30 times. The RMSD results are summarized in Appendix VIII of supplementary materials and illustrated in Figure 11. As expected, results show that the proposed poTAC method provides lower RMSD than the CTLRC method since it incorporates the system dynamics during the offline modeling, especially under a higher missing ratio setting.

7. Conclusions

In this article, we proposed a tensor-based control framework for autocorrelated HD streaming data in the presence of partially observed responses. First, we developed a one-step offline modeling method, which simultaneously incorporates both model parameter estimation and incomplete data imputation. Compared with the intuitive two-step method (i.e., completion first and then estimation), our developed method has two-fold benefits: (i) the system dynamic equation provides additional information for data completion, and (ii) parameter estimation benefits from the restored missing entries. We further employed the estimated regression model to execute the one-step-ahead predictive control, in which an online completion strategy, that is, LRPOC, was proposed to impute new missing observations by balancing the low-rank structure of the response and the estimation errors. To validate the effectiveness of the proposed framework, we conducted two sets of simulations and a case study in the semiconductor manufacturing process. The results showed that the proposed method outperforms the benchmarks in both offline completion and online control in a wider range of ranks and missing ratios for the data to be completed. In future

works, we can consider (i) the control model in a fully tensor format (including a tensor-based moving-average component), (ii) more general missing patterns and (iii) texture images for the process control. Besides, more efficient solutions to the proposed framework can be explored to make the online computation faster.

Supplementary Materials

In the online supplementary materials of this article, we provide a PDF file "poTAC-Appendix.pdf" to contain technical details. Additionally, we provide a folder "codes" containing a "ReadMe.txt" file and MATLAB codes for reproducing Figure 10 in this article.

Acknowledgments

We would like to thank the editor, associate editor, and referees for their constructive comments and suggestions that helped us considerably improve the article.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

The work of Kamran Paynabar was partially supported by the NSF grant CMMI1839591.

ORCID

Zihan Zhang http://orcid.org/0000-0002-5882-055X

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011), "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends®in Machine Learning*, 3, 1–122. [6]

Chen, X., and Sun, L. (2020), "Low-Rank Autoregressive Tensor Completion for Multivariate Time Series Forecasting," ArXiv: abs/2006.10436 [2,3]

Chen, X., and Sun, L. (2021), "Bayesian Temporal Factorization for Multidimensional Time Series Prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 4659–4673.

Gahrooei, M. R., Yan, H., Paynabar, K., and Shi, J. (2021), "Multiple Tensor-on-Tensor Regression: An Approach for Modeling Processes with Heterogeneous Sources of Data," *Technometrics*, 63, 147–159. [1,2,3]

Gandy, S., Recht, B., and Yamada, I. (2011), "Tensor Completion and Low-n-Rank Tensor Recovery via Convex Optimization," *Inverse Problems*, 27, 025010. [2]

Gaw, N., Yousefi, S., and Gahrooei, M. R. (2021), "Multimodal Data Fusion for Systems Improvement: A Review," *IISE Transactions*, 54, 1098–1116. DOI:10.1080/24725854.2021.1987593 [1]

Gómez, A. M. E., Li, D., and Paynabar, K. (2022), "An Adaptive Sampling Strategy for Online Monitoring and Diagnosis of High-Dimensional Streaming Data," *Technometrics*, 64, 253–269. [2]

Huang, B., Mu, C., Goldfarb, D., and Wright, J. (2015), "Provable Models for Robust Low-Rank Tensor Completion," *Pacific Journal of Optimization*, 11, 339–364. [5]

Kolda, T. G., and Bader, B. W. (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500. [2,4]

Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013), "Tensor Completion for Estimating Missing Values in Visual Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 208–220. [3,5,9]

Llosa-Vite, C., and Maitra, R. (2023), "Reduced-Rank Tensor-on-Tensor Regression and Tensor-Variate Analysis of Variance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 2282–2296. [2]

Lock, E. F. (2018), "Tensor-on-Tensor Regression," *Journal of Computational and Graphical Statistics*, 27, 638–647. [2]

Nesterov, Y. (1983), "A Method for Solving a Convex Programming Problem with Convergence Rate $\mathcal{O}\left(1/k^2\right)$," *Soviet Mathematics Doklady*, 27, 367–372. [7]

Thudumu, S., Branch, P., Jin, J., and Singh, J. (2020), "A Comprehensive Survey of Anomaly Detection Techniques for High Dimensional Big Data," *Journal of Big Data*, 7, 1–30. [1]

Wang, F., Gahrooei, M. R., Zhong, Z., Tang, T., and Shi, J. (2021), "An Augmented Regression Model for Tensors with Missing Values," *IEEE Transactions on Automation Science and Engineering*, 19, 2968–2984. [2,5,7]

Wold, S., Esbensen, K., and Geladi, P. (1987), "Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, 2, 37–52. [1]

Xiong, L., Chen, X., Huang, T. K., Schneider, J., and Carbonell, J. G. (2010), "Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization," in *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 211–222. [2]

Xue, J., Zhao, Y., Huang, S., Liao, W., Chan, J. C., and Kong, S. G. (2021), "Multilayer Sparsity-Based Tensor Decomposition for Low-Rank Tensor Completion," *IEEE Transactions on Neural Networks and Learning Systems*, 33, 6916–6930. [2]

Yan, H., Paynabar, K., and Pacella, M. (2019), "Structured Point Cloud Data Analysis via Regularized Tensor Regression for Process Modeling and Optimization," *Technometrics*, 61, 385–395. [1]

Yu, H., Rao, N., and Dhillon, I. S. (2016), "Temporal Regularized Matrix Factorization for High-Dimensional Time Series Prediction," in Advances in Neural Information Processing Systems (Vol. 29). [2]

Zhang, Z., and Yang, L. (2021), "State-Based Opportunistic Maintenance with Multifunctional Maintenance Windows," *IEEE Transactions on Reliability*, 70, 1481–1494. [2]

Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., Zhang, Z., and Cichocki, A. (2013), "Higher Order Partial Least Squares (HOPLS): A Generalized Multilinear Regression Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1660–1673. [1]

Zhong, Z., Mou, S., Hunt, J. H., and Shi, J. (2022), "Finite Element Analysis Model-Based Cautious Automatic Optimal Shape Control for Fuselage Assembly," *ASME Journal of Manufacturing Science and Engineering*, 144, 081009. [1,2]

Zhong, Z., Paynabar, K., and Shi, J. (2023), "Image-Based Feedback Control Using Tensor Analysis," *Technometrics*, 65, 305–314. DOI:10.1080/00401706.2022.2157880 [1,2,3,4,7,9,12]

Zhou, J., Sun, W. W., Zhang, J., and Li, L. (2021), "Partially Observed Dynamic Tensor Response Regression," *Journal of the American Statistical Association*, 118, 424–439. DOI:10.1080/01621459.2021.1938082 [3]