


Change point detection of multimode processes considering both mode transitions and parameter changes

Jun Xu, Jie Zhou, Xiaofang Huang & Kaibo Wang


To cite this article: Jun Xu, Jie Zhou, Xiaofang Huang & Kaibo Wang (05 Oct 2023): Change point detection of multimode processes considering both mode transitions and parameter changes, IISE Transactions, DOI: [10.1080/24725854.2023.2266001](https://doi.org/10.1080/24725854.2023.2266001)

To link to this article: <https://doi.org/10.1080/24725854.2023.2266001>

 View supplementary material [↗](#)

 Published online: 05 Oct 2023.

 Submit your article to this journal [↗](#)



 Article views: 289

 View related articles [↗](#)

 View Crossmark data [↗](#)



Change point detection of multimode processes considering both mode transitions and parameter changes

Jun Xu^a , Jie Zhou^{a,b}, Xiaofang Huang^b, and Kaibo Wang^{a,c} 

^aDepartment of Industrial Engineering, Tsinghua University, Beijing, China; ^bControl Engineering and Protection Department, R&D Center, Wind Power Business Group, Goldwind Science & Technology Co., Ltd, Beijing, China; ^cVanke School of Public Health, Tsinghua University, Beijing, China

ABSTRACT

Multimode processes are common in modern industry and refer to processes that work in multiple operating modes. Motivated by the torque control process of a wind turbine, we determine that there exist two types of changes in multimode processes: (i) mode transitions and (ii) parameter changes. Detecting both types of changes is an important issue in practice, but existing methods mainly consider one type of change, and thus, do not work well. To address this issue, we propose a novel modeling framework for the offline change point detection problem of multimode processes, which simultaneously considers mode transitions and parameter changes. We characterize each mode with a parametric cost function and formulate the problem as an optimization model. In the model, two penalty terms penalize the number of change points, and a series of constraints specify the multimode characteristics. With certain assumptions, the asymptotic property ensures the accuracy of the model solution. To solve the model, we propose an iterative algorithm and develop a multimode-pruned exact linear time (multi-PELT) method for initialization. The simulation study and the real case study demonstrate the effectiveness of our method against the state-of-the-art methods in terms of the accuracy of change point detection, mode identification, and parameter estimation.

ARTICLE HISTORY

Received 24 February 2023
Accepted 17 September 2023

KEYWORDS

Change point detection; constrained optimization model; multimode processes; wind turbine torque control

1. Introduction

Real processes in many industrial fields (such as chemical engineering and wind energy) are designed to work in multiple operating modes (Quiñones-Grueiro *et al.*, 2019). Each mode corresponds to a relationship among system signals, which is usually specified at the design stage and can be characterized by a parametric model. We refer to such a process as a *multimode process* and refer to a process with a single mode as a *single-mode process*. Our motivating example is the *torque control process* of a horizontal-axis wind turbine (Burton *et al.*, 2021). Figure 1 shows a typical generator torque–speed curve used for torque control, which contains five modes: (i) starting mode, (ii) minimum speed mode, (iii) Maximum Power Point Tracking (MPPT) mode (Yin *et al.*, 2017), (iv) rated speed mode, and (v) rated power mode. Each mode has a control parameter to characterize its relationship. When the ambient wind speed gradually increases, the torque control process sequentially transitions from the starting mode to the rated power mode along the torque–speed curve.

In the wind energy industry, an important issue is to assess the performance of wind turbines based on historical data collected by the Supervisory Control and Data Acquisition (SCADA) system (Ding, 2019). The key aspects

of performance assessment include power generation and control consistency; however, the latter has not been well explored. Power curves are widely utilized for power generation assessment and are functional curves that map environmental inputs (such as wind speed) to power output (Prakash *et al.*, 2022). Effective methods have been developed for power curve modeling (Lee *et al.*, 2015; Prakash *et al.*, 2023) and power curve comparison (Ding *et al.*, 2021; Prakash *et al.*, 2022). As the power curve is intrinsically affected by the control process (Burton *et al.*, 2021), control consistency is another crucial aspect of performance assessment. To assess the consistency, we need to partition the data stream into pieces so that data points in the same piece have the same mode; then, for each piece, we can estimate its parameter and compare it with the designed value. This is a typical *change point detection* problem, but it becomes more challenging in the context of multimode processes.

The main challenge is that there exist two types of changes in multimode processes, as shown in the torque control process. First, wind turbines usually make frequent transitions between two different modes to accommodate the nonstationary nature of wind. For example, in Figure 2(a), the manual evaluation reveals that the wind turbine made more than 2000 transitions within just 8 days. Second,

due to reasons such as control software upgrades, the parameters of some modes might change to new values. For example, after an upgrade on January 9, 2021, the generator torque–speed scatter plot in Figure 2(b) changed from right to left, where all control parameters, with the exception of the starting mode, changed to new values. In summary, the two types of changes in multimode processes are defined as follows:

1. **Mode transitions:** Transition from one mode to another accessible mode.
2. **Parameter changes:** The parameters of some modes change to new values, which occurs less often than mode transitions.

In this article, we aim to develop an offline method for change point detection of multimode processes when both mode transitions and parameter changes exist. Although research on *multimode process modeling and monitoring* and *multiple change point detection* has recently proliferated,

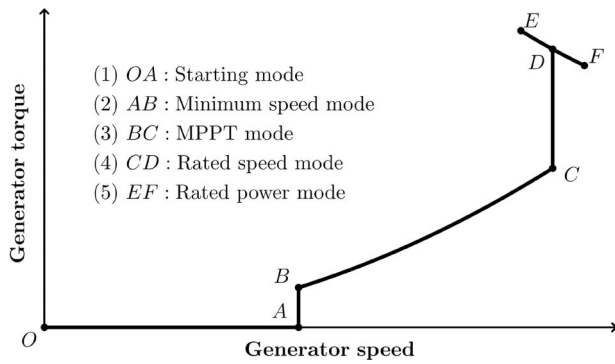


Figure 1. A typical wind turbine generator torque–speed curve with five modes.

existing methods mainly consider one type of change and thus do not work well when both types of changes exist, as we discuss below.

1.1. Literature review

1.1.1. Multimode process modeling and monitoring

The first branch of related literature is *multimode process modeling and monitoring*, which can be classified into *single-model* methods and *multiple-model* methods (Quiñones-Gruero *et al.*, 2019).

Single-model methods seek to transform the multimode data into approximately single-mode features and then construct a unified decision function for monitoring, such as the manifold learning-based approach (Yang *et al.*, 2017), neighborhood-based approach (Xie, 2020) and deep learning-based approach (Xu *et al.*, 2022). However, these methods omit the multimode characteristics and thus cannot distinguish between different modes.

Multiple-model methods characterize each mode with a local model, which can better model the multimode variation. In the offline stage, multiple-model methods are aimed at accurately modeling each mode based on historical data, which is similar to a clustering problem. Many clustering methods, such as K-means (Du *et al.*, 2017), fuzzy C-means (Xie and Shi, 2012), spectral clustering (Xu *et al.*, 2021), and density-based clustering (Yu *et al.*, 2022), have been applied. To improve interpretability, model-based methods have attracted increasing attention. For example, Jin and Liu (2013) built a piecewise linear regression tree for every quality variable in the serial-parallel multistage manufacturing process, Wang *et al.* (2015) applied a hidden Markov model to address the transitional mode, Haghani *et al.* (2015) combined a mixture model with partial least squares to model the nonlinear characteristics of wind

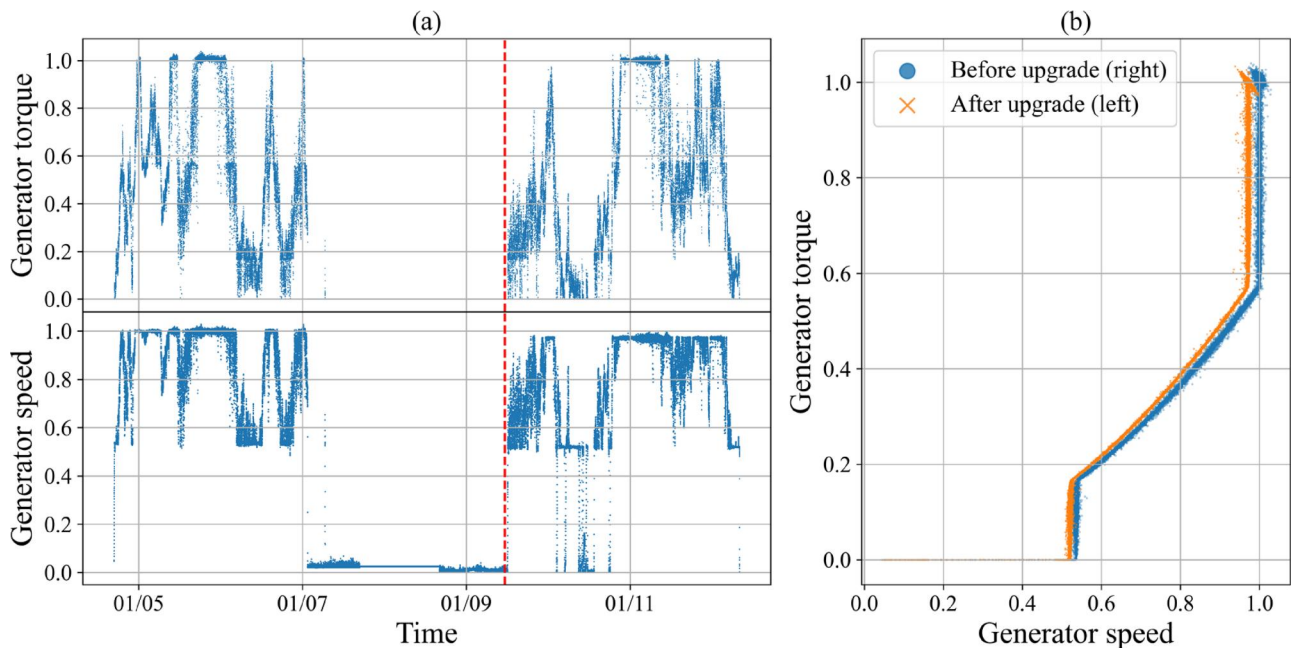


Figure 2. (a) Time series plot and (b) scatter plot of generator torque and speed for a real wind turbine during January 4–12, 2021. The two variables have been scaled by their rated values, the dashed line in (a) indicates the upgrade time, and data points during standby/downtime have been removed in (b).

turbines, and Zhao and Zhao (2022) applied the Toeplitz inverse covariance-based clustering method (Hallac *et al.*, 2017) in multimode process modeling and monitoring. However, the main limitation of the above methods is that they do not consider parameter changes in the offline stage, where data points in the same mode are assumed to share the same parameter. As a result, if two data points originate from the same mode but have different parameters, the above methods would misidentify them as different modes. In the online stage, multiple-model methods are aimed at identifying the mode of a new data point and then determining whether a fault occurs based on some monitoring statistics. For example, Jin and Liu (2013) constructed a group of Shewhart control charts to monitor the mean shifts of the residuals in each local model, and Zhao and Zhao (2022) constructed conditional thresholds of each variable for each mode to detect possible faults. However, when the parameter of some mode changes, multiple-model methods would inaccurately identify the parameter change as a new incoming mode (Chen *et al.*, 2023; Zhang *et al.*, 2023) or detect it as a fault and stop the monitoring procedure, and thus they cannot identify what the new parameter becomes and cannot well monitor the new incoming data after the parameter change.

1.1.2. Multiple change point detection

Another closely related research topic is *multiple change point detection*. Statistical process control (SPC) is widely utilized to detect process changes, and many types of control charts have been developed for accurate and timely anomaly detection (Shi, 2023; Wu *et al.*, 2023). However, these methods are typically designed to detect a single change point in an online manner, which cannot be easily extended to handle multiple change points. To detect multiple change points, various methods, both from *Bayesian* and *Non-Bayesian* perspectives, have been discussed in the literature.

Bayesian methods typically choose change points from a variety of candidate sets to maximize the posterior or marginal likelihood. For example, to detect mean shifts among multiple sequences, Ko *et al.* (2015) proposed a Dirichlet process hidden Markov model, and Jin *et al.* (2022) developed a Bayesian hierarchical model. To detect parameter changes of nonlinear univariate signals, a piecewise linear model was proposed with both particle filtering algorithm (Wu, Chen, Zhou and Li, 2016) and exact online Bayesian inference (Wu, Chen and Zhou, 2016), and Wu *et al.* (2019) further extended the univariate case to multivariate signals with a Bayesian piecewise constant mean and covariance model. However, these methods detect parameter changes with a single-mode setting and thus cannot accurately handle multimode processes.

Non-Bayesian methods usually formulate the problem as an optimization model by specifying the *cost function* that characterizes the relationship among variables for a given data piece (Truong *et al.*, 2020). To achieve a specific sparsity structure and avoid overfitting, *regularization methods* are typically combined into the optimization model by adding penalty terms to the objective function, such as the

linear penalty (Killick *et al.*, 2012; Xu *et al.*, 2023) and the fused LASSO penalty (Tibshirani *et al.*, 2005; Zhang *et al.*, 2021). To address different application scenarios, different types of cost functions have been applied in the literature. For example, the nonparametric likelihood (Zou *et al.*, 2014) can be applied when the relationship is difficult to model parametrically, the negative loglikelihood (Wang and Zou, 2023) is typically used to detect changes in distributional parameters and to detect changes in regression parameters, the cost function can be set as the sum of squared errors (Truong *et al.*, 2020). Recently, to detect changes in the linear manifold structure, Xu *et al.* (2023) proposed to set the cost function as the subspace reconstruction error plus a LASSO penalty. However, the above methods use a unified cost function for multiple change point detection, whereas for multimode processes, multiple cost functions are needed to better model multiple modes. As a result, the above methods cannot distinguish between different modes and are thus unable to accurately detect the change points.

To the best of our knowledge, some multiple change point detection methods also mentioned two types of changes, but the changes in their problem settings are different from our case (i.e., mode transitions and parameter changes), as we discuss below. In Wu *et al.* (2019), except for parameter changes, they proposed to detect the steady-state from the transient-state based on the expected length of the current segment, but their method is in a single-mode setting and cannot detect mode transitions. In Juodakis and Marsland (2023), they proposed to detect both signal changes and nuisance changes, but these two types of changes are both parameter changes, which are only distinguished by the segment duration.

Above all, existing methods cannot well handle multimode processes with both mode transitions and parameter changes. To fill this gap, we propose a novel modeling framework for the offline change point detection problem of multimode processes (CPD-MMP), which simultaneously considers the aforementioned two types of changes. The main contributions are listed as follows:

1. The proposed modeling framework can simultaneously detect mode transitions and parameter changes for multimode processes. Specifically, we incorporate engineering knowledge into the model by specifying the number of modes, mode transition matrix, and mode-specific cost functions, and formulate the problem as a constrained optimization model with two penalty terms.
2. An iterative algorithm is proposed to solve the optimization model, and a multimode-Pruned Exact Linear Time (multi-PELT) method is developed for initialization.
3. The proposed method can achieve better performance than the state-of-the-art methods in terms of change point detection, mode identification, and parameter estimation.

The remainder of the article is organized as follows: We formulate the CPD-MMP problem as an optimization model

in Section 2. In Section 3, we develop an iterative algorithm to solve the optimization model and discuss how to select appropriate penalty weights. In Section 4, we present the settings, results, and analysis of both the simulation study and the real case study. We conclude this article and discuss future research directions in Section 5. A comprehensive table describing every notation introduced in this article is available in Section S1 of the [Supplementary Materials](#).

2. Problem formulation

2.1. Basic settings from engineering knowledge

Consider a multimode process of length T with p variables. At time $t = 1, \dots, T$, we collect a vector $\mathbf{Y}_t = (Y_{1t}, \dots, Y_{pt})'$ from the process, where $Y_{it}, i = 1, \dots, p$ denotes variable i at time t . We denote the data piece from t_1 to t_2 ($t_1 \leq t_2$) by $\mathbf{Y}_{t_1:t_2} = (\mathbf{Y}_{t_1}, \mathbf{Y}_{t_1+1}, \dots, \mathbf{Y}_{t_2})$. The full data stream is $\mathbf{Y}_{1:T}$.

It is assumed that the following basic settings are known from engineering knowledge:

1. **Number of modes:** $M \geq 1$ is denoted as the number of modes, and $\mathcal{M} = \{1, \dots, M\}$ is denoted as the mode set. $M = 1$ represents a single-mode process.
2. **Mode transition matrix:** $\mathbf{A} = (a_{ij}), a_{ij} = 1, i, j \in \mathcal{M}$ is denoted as the mode transition matrix, which describes the transition availability between any two modes. If mode $i \in \mathcal{M}$ can transition to mode $j \in \mathcal{M}$, $a_{ij} = 1$; otherwise, $a_{ij} = 0$.
3. **Mode-specific cost function:** $\mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)}), \theta^{(i)} \in \Theta^{(i)}$ is denoted as the *cost function* that models the relationship among the p variables when the process is operating under mode $i \in \mathcal{M}$, where $\theta^{(i)}$ is the parameter, and $\Theta^{(i)}$ is the parameter space. If $\theta^{(i)}$ is unknown, we can estimate it with $\mathbf{Y}_{t_1:t_2}$ (a data piece from mode i) by

$$\hat{\theta}_{t_1:t_2}^{(i)} = \underset{\theta^{(i)} \in \Theta^{(i)}}{\operatorname{argmin}} \mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)}). \quad (1)$$

For example, if data points from mode i are independent and identically distributed with density function $f(\cdot; \theta^{(i)})$, a natural choice of the cost function is the negative loglikelihood, that is,

$$\mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)}) = - \sum_{t=t_1}^{t_2} \log f(\mathbf{Y}_t; \theta^{(i)}),$$

where $\theta^{(i)}$ is the distribution parameter. If there exists a linear relationship between one variable (e.g., Y_{1t}) and the remaining variables, we can use the sum of squared errors, that is,

$$\mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)}) = \sum_{t=t_1}^{t_2} \left(Y_{1t} - \theta_1^{(i)} - \theta_2^{(i)} Y_{2t} - \dots - \theta_p^{(i)} Y_{pt} \right)^2,$$

where $\theta^{(i)} = (\theta_1^{(i)}, \dots, \theta_p^{(i)})'$ contains the regression coefficients.

The mode-specific cost function $\mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)})$ quantifies how well the mode i and the parameter $\theta^{(i)}$ fit the given

data piece $\mathbf{Y}_{t_1:t_2}$, which is introduced here for ease of formulating the problem as an optimization model (please see [Section 2.3](#) for details).

As a result, proper specification of the cost function is important. To ensure model performance, the cost function of mode i is expected to well capture the nature of mode i , and the cost functions of two different modes are expected to be distinguishable. Some concrete assumptions about the cost function are provided in Section S2.1 of the [Supplementary Materials](#), which are imposed to construct the asymptotic property of the model solution.

In practice, as the multimode process operates according to engineering design, we recommend selecting the cost function and the corresponding parameter space based on engineering knowledge of each mode. In this way, we can expect that the selected cost functions can well capture the multimode characteristics of the process.

2.2. Definition of a change point and the related decision variables

At time $t = 1, \dots, T$, we denote the mode of \mathbf{Y}_t by m_t and the corresponding parameter by $\theta_t^{(m_t)}$. For ease of expressing parameter changes, we store the latest parameters of all modes in a parameter list denoted by $\boldsymbol{\theta}_t = (\theta_t^{(1)}, \dots, \theta_t^{(M)})$.

In this parameter list, $\theta_t^{(m_t)}$ is the parameter of the current mode m_t , while $\theta_t^{(i)}, i \neq m_t$ are $M - 1$ virtual parameters whose values are inherited from the previous time point, that is, $\theta_t^{(i)} = \theta_{t-1}^{(i)}$ for $i \neq m_t; t = 1, \dots, T$. In [Figure 3](#), the virtual parameters are marked by dashed circles. At time 0, the parameter is initialized as $\theta_0^{(i)} = \theta_{t_{0i}}^{(i)}$, where $t_{0i} = \min\{\tau | m_\tau = i\}$ is the first appearance time of mode i . The definition of a change point is given as follows:

Definition 1. (Change point). Time t is referred to as a change point if the mode or the parameter list at time $t + 1$ differs with time t , that is, if $m_{t+1} \neq m_t$ or $\boldsymbol{\theta}_{t+1} \neq \boldsymbol{\theta}_t$, for $t = 1, \dots, T - 1$.

As discussed in [Section 1](#), there are two types of changes in multimode processes, and generally, parameter changes occur less often than mode transitions. To clearly express this point, we divide all change points into two categories based on whether the parameter is changed, as defined below.

Definition 2. (Parameter-change point & Mode-transition point). A change point t is referred to as a parameter-change point if $\boldsymbol{\theta}_{t+1} \neq \boldsymbol{\theta}_t$ (case 2 and case 3 in [Figure 3](#)) and is referred to as a mode-transition point if $m_{t+1} \neq m_t, \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$ (case 4 in [Figure 3](#)).

[Definition 2](#) is illustrated in [Figure 3](#). We can see that the parameter change is defined in terms of the parameter list $\boldsymbol{\theta}_t$ instead of the parameter of the current mode (i.e., $\theta_t^{(m_t)}$), and thus mode transitions may not lead to parameter changes. As the $M - 1$ virtual parameters in $\boldsymbol{\theta}_t$ are not free to change, $\boldsymbol{\theta}_t$ is not independent of m_t . In addition, the

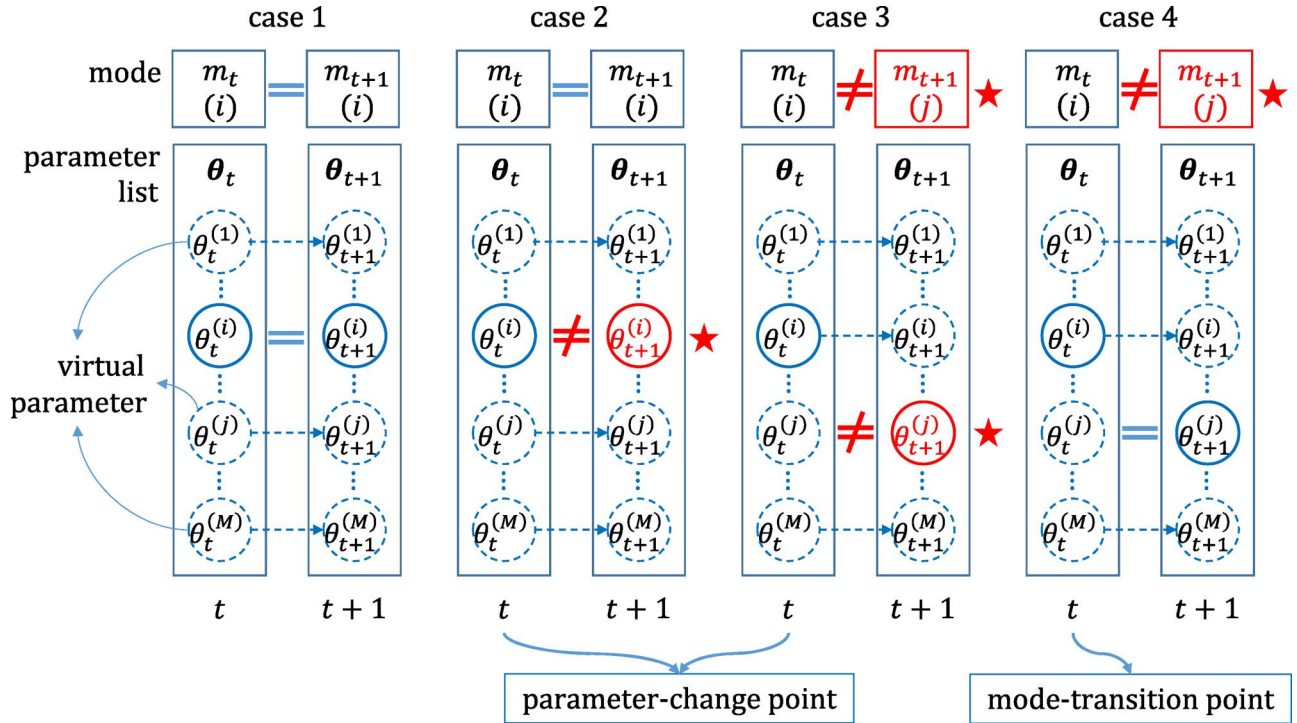


Figure 3. Illustration of parameter-change point (case 2 and case 3) and mode-transition point (case 4). The mode at time t is $m_t = i$ and the mode at time $t + 1$ is either i or $j \neq i$. The virtual parameters are marked by dashed circles, and the dashed horizontal arrow indicates that $\theta_{t+1}^{(l)} = \theta_t^{(l)}$ for $l \neq m_{t+1}$. The variables that change after time t are marked with a star.

parameter and mode can change at the same time (as shown by case 3 in Figure 3), which is referred to as a parameter-change point in Definition 2.

Definition 3. (Segment & Subsegment). The data piece between two successive parameter-change points is referred to as a segment. The data piece between two successive change points is referred to as a subsegment.

By Definition 2 and Definition 3, we assume that there are $C \geq 0$ parameter-change points in $Y_{1:T}$, which divide $Y_{1:T}$ into $C + 1$ segments; and we assume that there are $K_c \geq 0$ mode-transition points inside segment $c = 1, \dots, C + 1$, which divide segment c into $K_c + 1$ subsegments. Thus, the total number of change points in $Y_{1:T}$ is $C + \sum_{c=1}^{C+1} K_c$.

For the C parameter-change points, we define their positions as $0 < \tau_1^p < \dots < \tau_C^p < T$ with the convention of $\tau_0^p = 0$ and $\tau_{C+1}^p = T$, where the superscript “p” means “Parameter change”.

For the K_c mode-transition points inside segment $c = 1, \dots, C + 1$, we define their positions as $\tau_{c-1} < \tau_{c,1} < \dots < \tau_{c,K_c} < \tau_c^p$ with the convention of $\tau_{c,0} = \tau_{c-1}^p$ and $\tau_{c,K_c+1} = \tau_c^p$, where $\tau_{c,k}$ denotes the k th mode-transition point inside segment c . We refer to the k th subsegment in segment c as subsegment $c.k$.

As data points in the same subsegment share the same mode, we define $m_{c,k}$ as the mode of subsegment $c.k$ for $c = 1, \dots, C + 1; k = 1, \dots, K_c + 1$. As data points in the same segment share the same parameter, we define $\theta_c^p = (\theta_c^{(1)}, \dots, \theta_c^{(M)})$ as the parameter list of segment c for $c =$

$1, \dots, C + 1$. At time 0, θ_0^p is initialized as $\theta_0^{(i)} = \theta_{c_{0i}}^{(i)}, i \in \mathcal{M}$, where $c_{0i} = \min\{c | m_{c,k} = i \text{ for at least one } k \in \{1, \dots, K_c + 1\}\}$ is the first segment that contains a subsegment of mode i . Thus far, we have defined all decision variables as $\mathcal{DV} = \{C, (\tau_c^p), (\theta_c^p), (K_c), (\tau_{c,k}), (m_{c,k})\}$.

To construct the optimization model in Section 2.3, we also need to obtain the relationship between θ_c^p and θ_{c-1}^p . For two successive segments $c - 1$ and c , it follows from Definition 2 that the parameter changes at the first subsegment in segment c , i.e., subsegment $c.1$, which means that $\theta_c^{(m_{c,1})} \neq \theta_{c-1}^{(m_{c,1})}$, whereas the parameters of the other modes $i \neq m_{c,1}$ remain unchanged, that is,

$$\theta_c^{(i)} = \theta_{c-1}^{(i)} \text{ for } i \in \mathcal{M}, i \neq m_{c,1}; c = 1, \dots, C + 1. \quad (2)$$

Figure 4 shows an example of a multimode process with $M = 2$ modes. The mode set is $\mathcal{M} = \{1, 2\}$ and the mode transition matrix is $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. In Figure 4, there are $C = 2$ parameter-change points (i.e., τ_1^p and τ_2^p) and they divide the data stream into three segments. Taking segment 1 as an example, its parameter list is $\theta_1^p = (\theta_1^{(1)}, \theta_1^{(2)}) = (2, 5)$, and there is $K_1 = 1$ mode-transition point (i.e., $\tau_{1,1}$) inside segment 1. $\tau_{1,1}$ further divides segment 1 into two subsegments whose modes are $m_{1,1} = 1$ and $m_{1,2} = 2$. After the first parameter-change point τ_1^p , the parameter of mode 1 changes from $\theta_1^{(1)} = 2$ to $\theta_2^{(1)} = 1$, whereas the parameter of mode 2 keeps unchanged, i.e., $\theta_1^{(2)} = \theta_2^{(2)} = 5$.

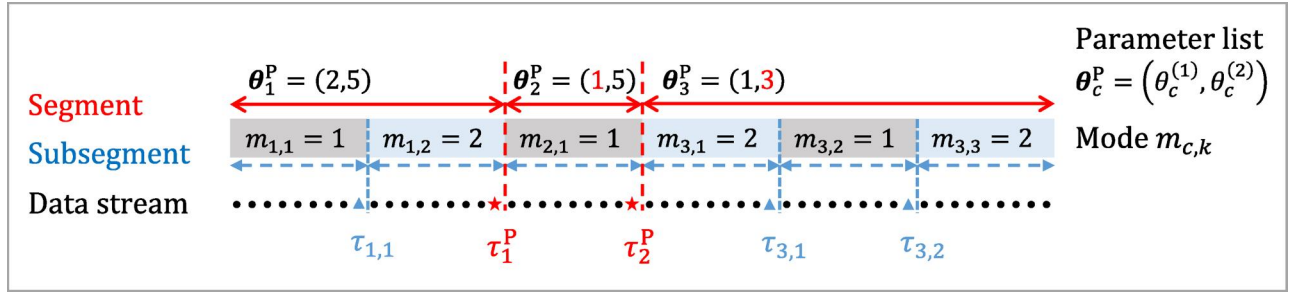


Figure 4. An example for illustrating the meanings of the decision variables.

2.3. Optimization model for CPD-MMP

In Section 2.2, we have defined all decision variables in $\mathcal{D}\mathcal{V}$. For ease of quick reference, Table 1 lists the notations used in the optimization model, and a more comprehensive notation table describing all notations introduced in this article can be found in Section S1 of the [Supplementary Materials](#).

We formulate the CPD-MMP problem as the following Model Q:

$$\begin{aligned} \min_{\mathcal{D}\mathcal{V}} & \sum_{c=1}^{C+1} \sum_{k=1}^{K_c+1} \mathcal{C}^{(m_{c,k})} \left(\mathbf{Y}_{\tau_{c,k-1}+1:\tau_{c,k}}; \theta_c^{(m_{c,k})} \right) \\ & + \beta \left(C + \sum_{c=1}^{C+1} K_c \right) + \lambda C, \end{aligned} \quad (\text{Q})$$

subject to :

$$0 = \tau_0^P < \tau_1^P < \dots < \tau_C^P < \tau_{C+1}^P = T; \quad (\text{C1})$$

$$\begin{aligned} \tau_{c-1}^P = \tau_{c,0} < \tau_{c,1} < \dots < \tau_{c,K_c} < \tau_{c,K_c+1} = \tau_c^P \text{ for } c \\ = 1, \dots, C+1; \end{aligned} \quad (\text{C2})$$

$$a_{m_{c,K_c+1}m_{c+1,1}} = 1 \text{ for } c = 1, \dots, C \text{ if } C > 0; \quad (\text{C3})$$

$$a_{m_{c,k}m_{c,k+1}} = 1 \text{ for } c = 1, \dots, C+1; k = 1, \dots, K_c \text{ if } K_c > 0; \quad (\text{C4})$$

$$\begin{aligned} \theta_0^{(i)} = \theta_{\min\{c|m_{c,k}=i\}}^{(i)} \text{ for } i \in \mathcal{M} \text{ if } \exists m_{c,k} = i; \theta_0^{(i)} = \text{null for } i \\ \in \mathcal{M} \text{ if } \nexists m_{c,k} = i; \end{aligned} \quad (\text{C5})$$

$$\theta_c^{(i)} = \theta_{c-1}^{(i)} \text{ for } i \in \mathcal{M}, i \neq m_{c,1}; c = 1, \dots, C+1; \quad (\text{C6})$$

$$C, K_1, \dots, K_{C+1} \in \mathbb{N}. \quad (\text{C7})$$

In the objective function of Model Q, the first term evaluates the total cost of all subsegments, the second term with weight β penalizes the total number of change points, and the third term with weight λ ensures the sparsity of parameter-change points. Constraints C1 and C2 specify all parameter-change points and all mode-transition points. Constraints C3 and C4 ensure the mode transition availability between two successive subsegments over two adjacent segments and inside the same segment. Constraints C5 initialize the parameter list, and when there does not exist a

subsegment of mode i , $\theta_0^{(i)}$ is set to null. Constraints C6 utilize Equation (2) to bridge the relationship between two successive parameter lists, and constraint C7 specifies the number of change points.

Notably, the proposed model is flexible, and thus, it can address a variety of multimode processes. First, there is no rigid restriction on the form of cost functions, which can be specified by engineering knowledge. Second, special cases have been covered by the proposed model: when $M = 1$, it degrades to the change point detection problem of a single-mode process (e.g., Xu *et al.*, 2023); when $C = 0$, it addresses a multimode process without a parameter change (e.g., Hallac *et al.*, 2017). Under certain assumptions, the asymptotic property given in Section S2 of the [Supplementary Materials](#) indicates that when the sampling resolution is sufficiently high, we can get a reasonably accurate solution by solving Model Q.

3. Optimization method

In this section, we derive an equivalent formulation of Model Q in Section 3.1. Based on the new formulation, we propose an iterative algorithm for solving CPD-MMP and develop a multi-PELT method for initialization, which is detailed in Section 3.2. We discuss how to select the two penalty weights (i.e., β and λ) in Section 3.3.

3.1. An equivalent formulation of the optimization model

To facilitate the derivation of the proposed algorithm, we derive a new formulation of CPD-MMP, where each parameter-change point is not explicitly defined, but rather it is implicitly specified by the parameter change of two adjacent subsegments.

Specifically, for the $K = C + \sum_{c=1}^{C+1} K_c$ change points defined in Section 2.2, that is, $\tau_{1,1}, \dots, \tau_{1,K_1}, \tau_1^P, \dots, \tau_C^P, \tau_{C+1,1}, \dots, \tau_{C+1,K_{C+1}}$, we rewrite them with a single index $k = 1, \dots, K$ as $\tau_1, \tau_2, \dots, \tau_K$, and we set $\tau_0 = 0$ and $\tau_{K+1} = T$. By Definition 3, these K change points divide the data stream into $K+1$ subsegments. For each subsegment $k = 1, \dots, K+1$, we denote its mode by m_k and the corresponding parameter list by $\theta_k = \left(\theta_k^{(1)}, \dots, \theta_k^{(M)} \right)$, where $\theta_k^{(i)}, i \neq m_k$ are $M-1$ virtual parameters that satisfy $\theta_k^{(i)} = \theta_{k-1}^{(i)}$ with $\theta_0^{(i)} = \theta_{k_{0i}}^{(i)}, k_{0i} = \min\{k|m_k = i\}$. For notational simplicity,

Table 1. Notations used in the optimization model.

| Notation | Description |
|--|--|
| T | The length of the data |
| $\mathbf{Y}_t = (Y_{1t}, \dots, Y_{pt})'$ | The vector of the p variables collected at time t |
| $\mathbf{Y}_{t_1:t_2} = (\mathbf{Y}_{t_1}, \mathbf{Y}_{t_1+1}, \dots, \mathbf{Y}_{t_2})$ | The data piece that contains data points from time t_1 to t_2 |
| $M, \mathcal{M} = \{1, \dots, M\}$ | The number of modes and the mode set |
| $\mathbf{A} = (a_{ij}), a_{ij} = 1, i, j \in \mathcal{M}$ | The mode transition matrix |
| $\mathcal{C}^{(i)}(\mathbf{Y}_{t_1:t_2}; \theta^{(i)})$ | The mode-specific cost function for mode i with parameter $\theta^{(i)}$ |
| C | The number of parameter-change points |
| K_c | The number of mode-transition points inside segment c |
| $\tau_1^p, \dots, \tau_C^p$ | The positions of all C parameter-change points with $\tau_0^p = 0, \tau_{C+1}^p = T$ |
| $\tau_{c,1}, \dots, \tau_{c,K_c}$ | The positions of all K_c mode-transition points inside segment c with $\tau_{c,0} = \tau_{c-1}^p, \tau_{c,K_c+1} = \tau_c^p$ |
| $m_{c,k}$ | The mode of subsegment $c.k$ |
| $\theta_0^{(i)}$ | The initial parameter of mode i |
| $\theta_c^p = (\theta_c^{(1)}, \dots, \theta_c^{(M)})$ | The parameter list of segment c |
| \mathcal{DV} | The set of decision variables in the optimization model |
| β, λ | The two penalty weights of the optimization model |

we define $\boldsymbol{\tau} = (\tau_1, \dots, \tau_K)$, $\mathbf{m} = (m_1, \dots, m_{K+1})$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{K+1})$.

In this way, a change point τ_k is a parameter-change point if $\boldsymbol{\theta}_{k+1} \neq \boldsymbol{\theta}_k$ for $k = 1, \dots, K$. Thus, the number of parameter-change points can be computed by

$$C = \sum_{k=1}^K \mathbb{I}(\boldsymbol{\theta}_{k+1} \neq \boldsymbol{\theta}_k) = \sum_{k=1}^K \sum_{i=1}^M \mathbb{I}(\theta_{k+1}^{(i)} \neq \theta_k^{(i)}).$$

Substituting the above notations and expressions into Model Q, we obtain the following Model P:

$$\begin{aligned} \min_{K, \boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\theta}} & \sum_{k=1}^{K+1} \mathcal{C}^{(m_k)}(\mathbf{Y}_{\tau_{k-1}+1:\tau_k}; \theta_k^{(m_k)}) + \beta K \\ & + \lambda \sum_{k=1}^K \sum_{i=1}^M \mathbb{I}(\theta_{k+1}^{(i)} \neq \theta_k^{(i)}), \end{aligned} \quad (\text{P})$$

subject to :

$$0 = \tau_0 < \tau_1 < \dots < \tau_K < \tau_{K+1} = T; \quad (\text{C8})$$

$$a_{m_k m_{k+1}} = 1 \text{ for } k = 1, \dots, K \text{ if } K > 0; \quad (\text{C9})$$

$$\begin{aligned} \theta_0^{(i)} &= \theta_{\min\{k|m_k=i\}}^{(i)} \text{ for } i \in \mathcal{M} \text{ if } \exists m_k = i; \theta_0^{(i)} = \text{null for } i \\ &\in \mathcal{M} \text{ if } \nexists m_k = i; \end{aligned} \quad (\text{C10})$$

$$\theta_k^{(i)} = \theta_{k-1}^{(i)} \text{ for } i \in \mathcal{M}, i \neq m_k; k = 1, \dots, K+1; \quad (\text{C11})$$

$$K \in \mathbb{N}. \quad (\text{C12})$$

3.2. Solution algorithm for CPD-MMP

We first introduce the solution algorithm for change point detection of single-mode processes. On this basis, we detail the proposed solution algorithm for CPD-MMP in Section 3.2.1–Section 3.2.3.

Change point detection of single-mode processes is typically formulated as

$$\min_{\substack{K \geq 0, 0 < \tau_1 < \dots < \tau_K < T, \\ \theta_k \in \Theta, k = 1, \dots, K+1}} \sum_{k=1}^{K+1} \mathcal{C}(\mathbf{Y}_{\tau_{k-1}+1:\tau_k}; \theta_k) + \beta K, \quad (3)$$

where K is the number of change points, τ_1, \dots, τ_K are K change points with $\tau_0 = 0$ and $\tau_{K+1} = T$, and $\theta_k \in \Theta$ is the parameter of subsegment k for $k = 1, \dots, K+1$. Equation (3) is simpler than Model P because it only involves one mode and one type of change point.

To exactly solve Equation (3), Jackson *et al.* (2005) proposed the Optimal Partitioning (OP) method based on dynamic programming. At time $s = 1, \dots, T$, the OP method utilizes a recursive formula by conditioning on the *last change point* before s . Let $F(s)$ denote the optimal value of Equation (3) for $\mathbf{Y}_{1:s}$, and let $t \in R_s$ denote the last change point before s , where $R_s = \{t | 0 \leq t < s\}$ is the candidate set of the last change point. As there is no change point between t and s , $F(s)$ is related to the optimal value for $\mathbf{Y}_{1:t}$, i.e., $F(t)$, plus the cost for subsegment $\mathbf{Y}_{t+1:s}$ and the penalty weight β , that is,

$$F(s) = \min_{t \in R_s} \left\{ F(t) + \min_{\theta \in \Theta} \mathcal{C}(\mathbf{Y}_{t+1:s}; \theta) + \beta \right\}. \quad (4)$$

Setting $F(0) = -\beta$ and utilizing the above recursion for $s = 1, \dots, T$, we can obtain the optimal partition of $\mathbf{Y}_{1:T}$ with a computation complexity of $\mathcal{O}(T^2)$.

To further improve the computational efficiency, Killick *et al.* (2012) proposed the PELT method that recursively prunes the candidate set based on the following theorem.

Theorem 1. (Theorem 3.1 in Killick *et al.*, 2012). *If there exists a constant K_p , such that for all $t < s < T$, we have*

$$\min_{\theta_l \in \Theta} \mathcal{C}(\mathbf{Y}_{t+1:s}; \theta_l) + \min_{\theta_r \in \Theta} \mathcal{C}(\mathbf{Y}_{s+1:T}; \theta_r) + K_p \leq \min_{\theta \in \Theta} \mathcal{C}(\mathbf{Y}_{t+1:T}; \theta). \quad (5)$$

Then, if

$$F(t) + \min_{\theta \in \Theta} \mathcal{C}(\mathbf{Y}_{t+1:s}; \theta) + K_p \geq F(s) \quad (6)$$

holds, t can never be the optimal last change point before T .

Based on [Theorem 1](#), the last change point that satisfies [Equation \(6\)](#) can be removed from the candidate set R_s , which can help reduce the size of R_s . The pruning process is

$$R_{s+1} = \{s\} \cup \left\{ t \in R_s \mid F(t) + \min_{\theta \in \Theta} \mathcal{C}(\mathbf{Y}_{t+1:s}; \theta) + K_P < F(s) \right\}. \quad (7)$$

The PELT method is detailed in [Algorithm 1](#). Killick *et al.* (2012) pointed out that nearly all cost functions used in practice satisfy [Equation \(5\)](#). For example, $K_P = 0$ if the cost function is the negative loglikelihood. They further proved that under certain conditions, the computation complexity of [Algorithm 1](#) is $\mathcal{O}(T)$, much faster than $\mathcal{O}(T^2)$ of the OP method.

Algorithm 1. The PELT method.

Set $F(0) = -\beta$, $R_1 = \{0\}$.

For $s = 1, \dots, T$:

 Calculate $F(s)$ by [Equation \(4\)](#).

 Set R_{s+1} by [Equation \(7\)](#).

The optimal value is $F(T)$.

The optimal solution is obtained by following the recursive path of $F(T)$.

However, [Algorithm 1](#) cannot be directly extended to solve Model P, because a recursion like [Equation \(4\)](#) is not satisfied in our case. In Model P, the second penalty term and the constraints C10 and C11 introduce a complex parameter dependency structure. As a result, the estimation of the parameter $\theta_k^{(m_k)}$ not only depends on subsegment k itself, but also depends on other subsegments $l \in \{l \neq k \mid m_l = m_k, \theta_l^{(m_l)} = \theta_k^{(m_k)}\}$ with the same mode and the same parameter.

To tackle this difficulty, we learn from Wang *et al.* (2022) and propose an iterative algorithm outlined in [Algorithm 2](#) to solve Model P, where the decision variables are divided into two groups:

1. K, τ, \mathbf{m} : The number of change points, the list of change points, and the modes of all subsegments.
2. C, τ^P, θ^P : The number of parameter-change points, the list of parameter-change points, and the parameter lists of all segments.

Note that for ease of algorithm derivation, we represent $\theta = (\theta_1, \dots, \theta_{K+1})$ (i.e., the $K+1$ parameter lists of all subsegments in Model P) by $\tau^P = (\tau_1^P, \dots, \tau_C^P)$ (i.e., the C parameter-change points with $\tau_0^P = 0, \tau_{C+1}^P = T$) and $\theta^P = (\theta_1^P, \dots, \theta_{C+1}^P)$ (i.e., the $C+1$ parameter lists of all segments). Since subsegments in the same segment share the same parameter list, if we obtain C, τ^P, θ^P , we can get θ by

$$\begin{aligned} \theta_k &= \theta_c^P \text{ if } \tau_{c-1}^P < \tau_k \leq \tau_c^P, \text{ for } k = 1, \dots, K+1; c \\ &= 1, \dots, C+1. \end{aligned}$$

In [Algorithm 2](#), we first apply a multi-PELT method to initialize K, τ, \mathbf{m} , which is detailed in [Section 3.2.1](#); then, we alternately update C, τ^P, θ^P and K, τ, \mathbf{m} until converge, as detailed in [Section 3.2.2](#) and [Section 3.2.3](#). [Algorithm 2](#) converges when all decision variables keep unchanged.

Algorithm 2. An iterative algorithm for solving Model P.

Initialize $K = \tilde{K}, \tau = \tilde{\tau}, \mathbf{m} = \tilde{\mathbf{m}}$; please refer to [Section 3.2.1](#).

Iteration until converge:

 Update C, τ^P, θ^P given K, τ, \mathbf{m} ; please refer to [Section 3.2.2](#).

 Update K, τ, \mathbf{m} given C, τ^P, θ^P ; please refer to [Section 3.2.3](#).

3.2.1. Initializing K, τ, \mathbf{m}

Good initialization of K, τ, \mathbf{m} is important for [Algorithm 2](#). One intuitive idea is to slightly modify Model P so that the modified model is relatively easy to solve. Based on the above consideration, we propose to initialize K, τ, \mathbf{m} by solving the following Model P0, which is obtained by dropping the second penalty term of Model P:

$$\min_{K, \tau, \mathbf{m}, \theta} \sum_{k=1}^{K+1} \mathcal{C}^{(m_k)}(\mathbf{Y}_{\tau_{k-1}+1:\tau_k}; \theta_k^{(m_k)}) + \beta K, \quad (\text{P0})$$

subject to constraints C8–C12 of Model P.

Since there is no penalty for the number of parameter-change points, Model P0 will generate a solution where all change points become parameter-change points, so the parameter of each subsegment can be independently estimated, no longer relying on other subsegments. Therefore, Model P0 can be solved by dynamic programming.

Let $F(s, j)$ denote the optimal value of Model P0 for $\mathbf{Y}_{1:s}$ when the mode of \mathbf{Y}_s is j . Then, $F(s, j)$ satisfies the following recursive relationship:

$$F(s, j) = \min_{t \in R(s, j)} \left\{ \min_{i: a_{ij}=1} F(t, i) + \min_{\theta^{(j)} \in \Theta^{(j)}} \mathcal{C}^{(j)}(\mathbf{Y}_{t+1:s}; \theta^{(j)}) + \beta \right\}, \quad (8)$$

where $R(s, j) = \{t \mid 0 \leq t < s\}$ is the candidate set of the last change point at time s when the mode of \mathbf{Y}_s is j . Recall that for single-mode processes, the PELT method can help prune the candidate set. Fortunately, for multimode processes, [Theorem 2](#) below enables us to do similar pruning for $R(s, j)$.

Theorem 2. *If there exists a constant K_P , such that for all $t < s < T$ and for all $j \in \mathcal{M}$, we have*

$$\begin{aligned} & \min_{\theta^{(j)} \in \Theta^{(j)}} \mathcal{C}^{(j)}(\mathbf{Y}_{t+1:s}; \theta^{(j)}) + \min_{\theta^{(j)} \in \Theta^{(j)}} \mathcal{C}^{(j)}(\mathbf{Y}_{s+1:T}; \theta^{(j)}) + K_P \\ & \leq \min_{\theta^{(j)} \in \Theta^{(j)}} \mathcal{C}^{(j)}(\mathbf{Y}_{t+1:T}; \theta^{(j)}). \end{aligned} \quad (9)$$

Then, if

$$\min_{i: a_{ij}=1} F(t, i) + \min_{\theta^{(j)} \in \Theta^{(j)}} \mathcal{C}^{(j)}(\mathbf{Y}_{t+1:s}; \theta^{(j)}) + K_P \geq \min_{i: a_{ij}=1} F(s, i) \quad (10)$$

holds, t can never be the optimal last change point before T when the mode of \mathbf{Y}_T is j .

The proof of [Theorem 2](#) is provided in Section S3 of the [Supplementary Materials](#). Based on [Theorem 2](#), for $j \in \mathcal{M}$ and at time s , we only need to consider candidate last change points that violate [Equation \(10\)](#) for future recursions. Therefore, the candidate set $R(s, j)$ can be pruned recursively by

$$\begin{aligned} R(s+1, j) &= \{s\} \\ &\cup \left\{ t \in R(s, j) \mid \min_{i: a_{ij}=1} F(t, i) + \min_{\theta^{(j)} \in \Theta^{(j)}} C^{(j)}(\mathbf{Y}_{t+1:s}; \theta^{(j)}) \right. \\ &\quad \left. + K_P < \min_{i: a_{ij}=1} F(s, i) \right\}. \end{aligned} \quad (11)$$

The multi-PELT method for solving Model P0 is detailed in [Algorithm 3](#).

Algorithm 3. The multi-PELT method for solving Model P0.

Set $F(0, j) = -\beta, R(1, j) = \{0\}$ for all $j \in \mathcal{M}$.

For $s = 1, \dots, T$:

 Calculate $F(s, j)$ by [Equation \(8\)](#) for all $j \in \mathcal{M}$.

 Set $R(s+1, j)$ by [Equation \(11\)](#) for all $j \in \mathcal{M}$.

The optimal value is $\min_{j \in \mathcal{M}} F(T, j)$.

The optimal solution is obtained by following the recursive path of $\min_{j \in \mathcal{M}} F(T, j)$.

We assume that there are \tilde{K} change points in the optimal solution of Model P0, denoted by $\tilde{\tau} = (\tilde{\tau}_1, \dots, \tilde{\tau}_{\tilde{K}})$ with the convention of $\tilde{\tau}_0 = 0$ and $\tilde{\tau}_{\tilde{K}+1} = T$, and we denote the optimal modes of all subsegments by $\tilde{\mathbf{m}} = (\tilde{m}_1, \dots, \tilde{m}_{\tilde{K}+1})$. We initialize $K, \boldsymbol{\tau}, \mathbf{m}$ with

$$K = \tilde{K}, \boldsymbol{\tau} = \tilde{\boldsymbol{\tau}}, \mathbf{m} = \tilde{\mathbf{m}}.$$

3.2.2. Updating $C, \boldsymbol{\tau}^P, \boldsymbol{\theta}^P$

To update $C, \boldsymbol{\tau}^P, \boldsymbol{\theta}^P$, we need to solve Model P given $K = \tilde{K}, \boldsymbol{\tau} = \tilde{\boldsymbol{\tau}}, \mathbf{m} = \tilde{\mathbf{m}}$, which is formulated as the following Model P1:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \sum_{k=1}^{\tilde{K}+1} C^{(\tilde{m}_k)}(\mathbf{Y}_{\tilde{\tau}_{k-1}+1:\tilde{\tau}_k}; \boldsymbol{\theta}_k^{(\tilde{m}_k)}) \\ + \lambda \sum_{k=1}^{\tilde{K}} \sum_{i=1}^M \mathbb{I}(\boldsymbol{\theta}_{k+1}^{(i)} \neq \boldsymbol{\theta}_k^{(i)}), \end{aligned} \quad (P1)$$

subject to :

$$\begin{aligned} \boldsymbol{\theta}_0^{(i)} &= \boldsymbol{\theta}_{\min\{k|\tilde{m}_k=i\}}^{(i)} \text{ for } i \in \mathcal{M} \text{ if } \exists \tilde{m}_k = i; \boldsymbol{\theta}_0^{(i)} = \text{null for } i \\ &\in \mathcal{M} \text{ if } \nexists \tilde{m}_k = i; \end{aligned} \quad (C13)$$

$$\boldsymbol{\theta}_k^{(i)} = \boldsymbol{\theta}_{k-1}^{(i)} \text{ for } i \in \mathcal{M}, i \neq \tilde{m}_k; k = 1, \dots, \tilde{K} + 1. \quad (C14)$$

Note that $\boldsymbol{\theta}$ will be expressed as $C, \boldsymbol{\tau}^P, \boldsymbol{\theta}^P$ in the following solution procedures.

To solve Model P1, we rewrite its objective function by putting terms from the same mode together:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \sum_{k=1}^{\tilde{K}+1} C^{(\tilde{m}_k)}(\mathbf{Y}_{\tilde{\tau}_{k-1}+1:\tilde{\tau}_k}; \boldsymbol{\theta}_k^{(\tilde{m}_k)}) + \lambda \sum_{k=1}^{\tilde{K}} \sum_{i=1}^M \mathbb{I}(\boldsymbol{\theta}_{k+1}^{(i)} \neq \boldsymbol{\theta}_k^{(i)}) \\ = \min_{\boldsymbol{\theta}} \sum_{i=1}^M \left[\sum_{k \in \{k|\tilde{m}_k=i\}} C^{(i)}(\mathbf{Y}_{\tilde{\tau}_{k-1}+1:\tilde{\tau}_k}; \boldsymbol{\theta}_k^{(i)}) + \lambda \sum_{k=2}^{\tilde{K}+1} \mathbb{I}(\boldsymbol{\theta}_k^{(i)} \neq \boldsymbol{\theta}_{k-1}^{(i)}) \right] \\ = \sum_{i=1}^M \min_{\boldsymbol{\theta}_k^{(i)}, k=1, \dots, \tilde{K}+1} \left[\underbrace{\sum_{k \in \{k|\tilde{m}_k=i\}} C^{(i)}(\mathbf{Y}_{\tilde{\tau}_{k-1}+1:\tilde{\tau}_k}; \boldsymbol{\theta}_k^{(i)}) + \lambda \sum_{k=2}^{\tilde{K}+1} \mathbb{I}(\boldsymbol{\theta}_k^{(i)} \neq \boldsymbol{\theta}_{k-1}^{(i)})}_{\text{The objective function of the } i\text{th sub-model}} \right]. \end{aligned}$$

Therefore, Model P1 is equivalent to M independent sub-models, each of which deals with subsegments of the same mode, as formulated in the following Model P1.i:

$$\begin{aligned} \min_{\boldsymbol{\theta}_k^{(i)}, k=1, \dots, \tilde{K}+1} \left[\sum_{k \in \{k|\tilde{m}_k=i\}} C^{(i)}(\mathbf{Y}_{\tilde{\tau}_{k-1}+1:\tilde{\tau}_k}; \boldsymbol{\theta}_k^{(i)}) \right. \\ \left. + \lambda \sum_{k=2}^{\tilde{K}+1} \mathbb{I}(\boldsymbol{\theta}_k^{(i)} \neq \boldsymbol{\theta}_{k-1}^{(i)}) \right], \end{aligned} \quad (P1.i)$$

subject to :

$$\boldsymbol{\theta}_0^{(i)} = \boldsymbol{\theta}_{\min\{k|\tilde{m}_k=i\}}^{(i)}; \quad (C15)$$

$$\boldsymbol{\theta}_k^{(i)} = \boldsymbol{\theta}_{k-1}^{(i)} \text{ if } i \neq \tilde{m}_k, \text{ for } k = 1, \dots, \tilde{K} + 1. \quad (C16)$$

To simplify the notation, we assume that there are K^i subsegments of mode i , with their indexes denoted by $k_{i,1}, \dots, k_{i,K^i}$. Let $\mathbf{Y}_1^i, \dots, \mathbf{Y}_{K^i}^i$ denote the data pieces in these K^i subsegments, in which

$$\mathbf{Y}_l^i = \mathbf{Y}_{\tilde{\tau}_{k_{i,l}-1}+1:\tilde{\tau}_{k_{i,l}}}, l = 1, \dots, K^i$$

is the data piece in the l th subsegment of mode i , and let $\mathbf{Y}_{l_1:l_2}^i = (\mathbf{Y}_{l_1}^i, \dots, \mathbf{Y}_{l_2}^i), 1 \leq l_1 \leq l_2 \leq K^i$ denote the data pieces from l_1 to l_2 . Substituting constraints C15, C16 and the above notations into the objective function of Model P1.i, we obtain the following Model P1.i':

$$\min_{\boldsymbol{\theta}_{k_{i,l}}^{(i)}, l=1, \dots, K^i} \sum_{l=1}^{K^i} C^{(i)}(\mathbf{Y}_l^i; \boldsymbol{\theta}_{k_{i,l}}^{(i)}) + \lambda \sum_{l=2}^{K^i} \mathbb{I}(\boldsymbol{\theta}_{k_{i,l}}^{(i)} \neq \boldsymbol{\theta}_{k_{i,l-1}}^{(i)}). \quad (P1.i')$$

Model P1.i' is similar to the change point detection problem of single-mode processes and can be solved by the PELT method. Let $G^{(i)}(l)$ denote the optimal value of Model P1.i' for $\mathbf{Y}_{1:l}^i$. Then, $G^{(i)}(l)$ satisfies the following recursive relationship:

$$G^{(i)}(l) = \min_{n \in R^{(i)}(l)} \left\{ G^{(i)}(n) + \min_{\boldsymbol{\theta}^{(i)} \in \Theta^{(i)}} \sum_{o=n+1}^l C^{(i)}(\mathbf{Y}_o^i; \boldsymbol{\theta}^{(i)}) + \lambda \right\}, \quad (12)$$

where $R^{(i)}(l)$ is the candidate set of the last change point for the l th subsegment of mode i . To apply the PELT method, we assume that there exists a constant K_{P1} such that for all

Table 2. Relationships, parameters, and cost functions of the five modes in the torque control process.

| Mode | Relationship | Parameter | Cost function |
|-----------------------|--|--|--|
| 1. Starting mode | $Y_{2t} = \theta^{(1)} + \epsilon_1$ | $\theta^{(1)} \in \Theta^{(1)} = \{0\}$ | $\sum_{t=t_1}^{t_2} (Y_{2t} - \theta^{(1)})^2$ |
| 2. Minimum speed mode | $Y_{1t} = \theta^{(2)} + \epsilon_2$ | $\theta^{(2)} \in \Theta^{(2)} = [0.45, 0.60]$ | $\sum_{t=t_1}^{t_2} (Y_{1t} - \theta^{(2)})^2$ |
| 3. MPPT mode | $Y_{2t} = \theta^{(3)} Y_{1t}^2 + \epsilon_3$ | $\theta^{(3)} \in \Theta^{(3)} = [0.45, 0.65]$ | $\sum_{t=t_1}^{t_2} (Y_{2t} - \theta^{(3)} Y_{1t}^2)^2$ |
| 4. Rated speed mode | $Y_{1t} = \theta^{(4)} + \epsilon_4$ | $\theta^{(4)} \in \Theta^{(4)} = [0.90, 1.05]$ | $\sum_{t=t_1}^{t_2} (Y_{1t} - \theta^{(4)})^2$ |
| 5. Rated power mode | $Y_{2t} = \theta^{(5)} Y_{1t}^{-1} + \epsilon_5$ | $\theta^{(5)} \in \Theta^{(5)} = [0.80, 1.10]$ | $\sum_{t=t_1}^{t_2} (Y_{2t} - \theta^{(5)} Y_{1t}^{-1})^2$ |

$n < l < L$ and $i \in \mathcal{M}$,

$$\min_{\theta^{(i)} \in \Theta^{(i)}} \sum_{o=n+1}^l C^{(i)}(\mathbf{Y}_o^i; \theta^{(i)}) + \min_{\theta^{(i)} \in \Theta^{(i)}} \sum_{o=l+1}^L C^{(i)}(\mathbf{Y}_o^i; \theta^{(i)}) + K_{P1} \leq \min_{\theta^{(i)} \in \Theta^{(i)}} \sum_{o=n+1}^L C^{(i)}(\mathbf{Y}_o^i; \theta^{(i)}).$$

Then, we can recursively prune the candidate set $R^{(i)}(l)$ by

$$R^{(i)}(l+1) = \{l\} \cup \left\{ n \in R^{(i)}(l) \mid G^{(i)}(n) + \min_{\theta^{(i)} \in \Theta^{(i)}} \sum_{o=n+1}^l C^{(i)}(\mathbf{Y}_o^i; \theta^{(i)}) + K_{P1} < G^{(i)}(l) \right\}. \quad (13)$$

The PELT method for solving Model P1. i' is summarized in [Algorithm 4](#).

Algorithm 4. The PELT method for solving Model P1. i' .

Set $G^{(i)}(0) = -\lambda, R^{(i)}(1) = \{0\}$.

For $l = 1, \dots, K^i$:

 Calculate $G^{(i)}(l)$ by [Equation \(12\)](#).

 Set $R^{(i)}(l+1)$ by [Equation \(13\)](#).

The optimal value is $G^{(i)}(K^i)$.

The optimal solution is obtained by following the recursive path of $G^{(i)}(K^i)$.

We assume that there are $\tilde{C}^{(i)}$ change points in the optimal solution of Model P1. i' , denoted by $\tilde{\tau}_c^{P,i}, c = 1, \dots, \tilde{C}^{(i)}$, which are all parameter-change points of Model P1. The optimal parameter for each subsegment of mode i is denoted by $\tilde{\theta}_{k,l}^i, l = 1, \dots, K^i$. Then, the optimal parameters of Model P1. i , denoted by $\tilde{\theta}_k^{(i)}, k = 1, \dots, \tilde{K} + 1$, can be obtained by extending $\tilde{\theta}_{k,l}^i, l = 1, \dots, K^i$ to all $\tilde{K} + 1$ subsegments according to constraints C15 and C16.

Moreover, if there exists a mode $i \in \mathcal{M}$ not in $\tilde{\mathbf{m}}$, that is, if $\tilde{m}_k \neq i$ for all $k = 1, \dots, \tilde{K} + 1$, we cannot get the corresponding parameter $\tilde{\theta}_k^{(i)}$ by [Algorithm 4](#), as there is no data from mode i . In this case, we propose to estimate $\tilde{\theta}_k^{(i)}$ in the full data stream $\mathbf{Y}_{1:T}$ with a moving window of width w , as shown in [Equation \(14\)](#). The width w can be specified according to engineering knowledge. The number of parameter-change points for mode i is set to $\tilde{C}^{(i)} = 0$.

$$\tilde{\theta}_k^{(i)} = \operatorname{argmin}_{\theta^{(i)} \in \Theta^{(i)}} \left\{ \min_{t=1, \dots, T-w+1} C^{(i)}(\mathbf{Y}_{t:t+w-1}; \theta^{(i)}) \right\}. \quad (14)$$

After solving all Model P1. i' for $i \in \mathcal{M}$, we merge their optimal solutions $(\tilde{C}^{(i)}, (\tilde{\tau}_c^{P,i}), (\tilde{\theta}_k^{(i)}))$ to obtain the optimal solution of Model P1. Specifically, let $\tilde{C} = \sum_{i=1}^M \tilde{C}^{(i)}$ denote the total number of parameter-change points, let $\tilde{\tau}_1^P, \dots, \tilde{\tau}_{\tilde{C}}^P$ denote the \tilde{C} parameter-change points with $\tilde{\tau}_0^P = 0, \tilde{\tau}_{\tilde{C}+1}^P = T$, and let $\tilde{\theta}_c^P = (\tilde{\theta}_c^{(1)}, \dots, \tilde{\theta}_c^{(M)})$ denote the parameter list of segment c for $c = 1, \dots, \tilde{C} + 1$. We define $\tilde{\tau}^P = (\tilde{\tau}_1^P, \dots, \tilde{\tau}_{\tilde{C}}^P)$ and $\tilde{\theta}^P = (\tilde{\theta}_1^P, \dots, \tilde{\theta}_{\tilde{C}+1}^P)$. Then, we update C, τ^P, θ^P with

$$C = \tilde{C}, \tau^P = \tilde{\tau}^P, \theta^P = \tilde{\theta}^P. \quad (15)$$

3.2.3. Updating K, τ, \mathbf{m}

To update K, τ, \mathbf{m} , we need to solve Model P given $C = \tilde{C}, \tau^P = \tilde{\tau}^P, \theta^P = \tilde{\theta}^P$, which is formulated as the following Model P2:

$$\min_{K, \tau, \mathbf{m}, \theta, \tau^P} \sum_{k=1}^{K+1} C^{(m_k)}(\mathbf{Y}_{\tau_{k-1}+1:\tau_k}; \theta_k^{(m_k)}) + \beta K, \quad (P2)$$

subject to C8 – C12 of Model P, and

$$\tau_1^P < \dots < \tau_{\tilde{C}}^P \text{ and } \{\tau_1^P, \dots, \tau_{\tilde{C}}^P\} \subseteq \{\tau_1, \dots, \tau_K\}; \quad (C17)$$

$$\theta_k^{(i)} = \tilde{\theta}_c^{(i)} \text{ if } \tau_{c-1}^P < \tau_k \leq \tau_c^P, \text{ for } k = 1, \dots, K + 1; i \in \mathcal{M}; c = 1, \dots, \tilde{C} + 1. \quad (C18)$$

Here, $\tau_1^P, \dots, \tau_{\tilde{C}}^P$ are auxiliary decision variables denoting the \tilde{C} updated parameter-change points in τ , which may differ from the given $\tilde{\tau}^P = (\tilde{\tau}_1^P, \dots, \tilde{\tau}_{\tilde{C}}^P)$, as shown in constraint C17. The given parameter $\tilde{\theta}_c^{(i)}, i \in \mathcal{M}$ is assigned to all subsegments in segment c for $c = 1, \dots, \tilde{C} + 1$ by constraints C18.

Due to the combinatorial optimization structure of Model P2, we propose to solve Model P2 by an iterative algorithm outlined in [Algorithm 5](#), where the constraints related to θ and τ^P have been incorporated into the objective

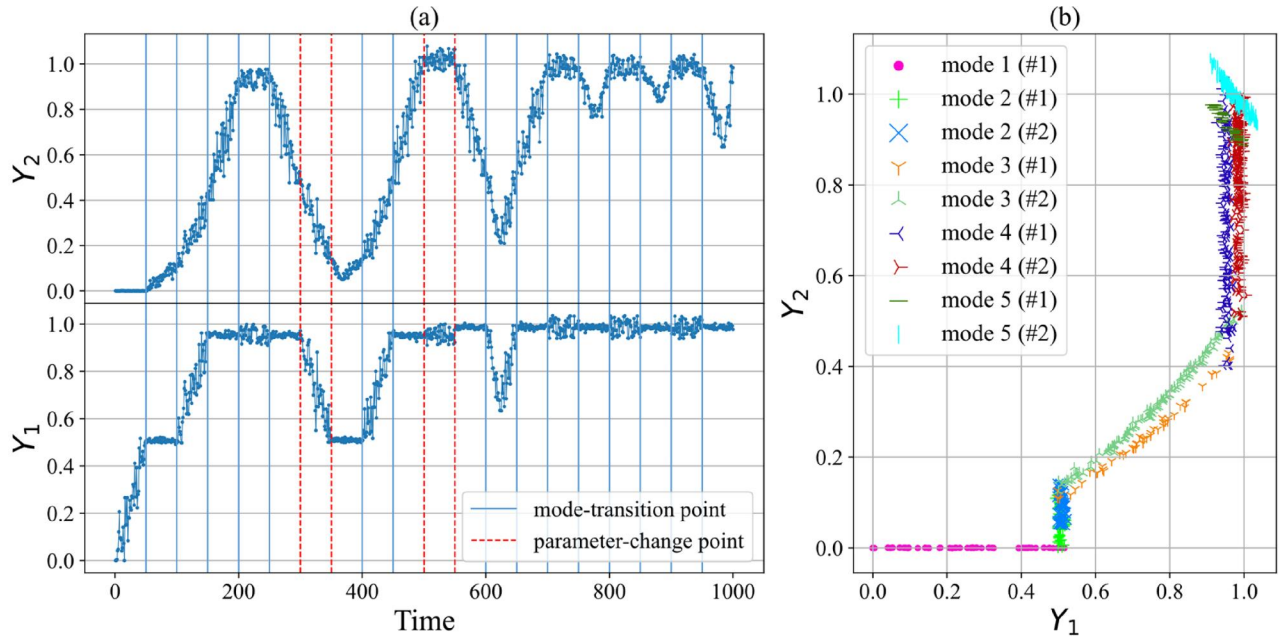


Figure 5. (a) Time series plot and (b) scatter plot of a simulated sample with $C = 4$ and $\sigma = 0.006$. The legend “mode i ($\#j$)” in (b) represents data points from mode i with the j th parameter level.

function, $\tau^p = (\tau_1^p, \dots, \tau_{\tilde{C}}^p)$ has been merged into τ, \mathbf{m} as $\tau = (\tau_1, \tau_1^p, \dots, \tau_{\tilde{C}}^p, \tau_{\tilde{C}+1})$ and $\mathbf{m} = (\mathbf{m}_1, \dots, \mathbf{m}_{\tilde{C}+1})$, and τ, \mathbf{m} are further divided into two groups:

1. $\tau_c, \mathbf{m}_c, c = 1, \dots, \tilde{C} + 1$: The mode-transition points and the modes of subsegments in segment c .
2. $\tau^p = (\tau_1^p, \dots, \tau_{\tilde{C}}^p)$: The updated parameter-change points in τ .

Algorithm 5. An iterative algorithm for solving Model P2.

Initialize $\tau^p = \hat{\tau}^p$.

Iteration until converge:

Update $\tau_c, \mathbf{m}_c, c = 1, \dots, \tilde{C} + 1$ given τ^p .

Update τ^p given $\tau_c, \mathbf{m}_c, c = 1, \dots, \tilde{C} + 1$.

The details for updating $\tau_c, \mathbf{m}_c, c = 1, \dots, \tilde{C} + 1$ and updating τ^p are provided in Section S4 of the [Supplementary Materials](#). The main ideas are summarized as follows:

1. Updating $\tau_c, \mathbf{m}_c, c = 1, \dots, \tilde{C} + 1$: We first independently update τ_c, \mathbf{m}_c for each subsegment by the multi-PELT method and then merge them by dynamic programming.
2. Updating τ^p : For the simple case, we update a single τ_c^p by a line search. For the complex case, we simultaneously update $n \geq 2$ successive parameter-change points by dynamic programming.

3.3. Selection of the penalty weights

There are two penalty weights in the proposed model: $\beta > 0$ is for the number of change points, and $\lambda > 0$ is for the

number of parameter-change points. We provide a simple approach for selecting β and λ with labeled training data. In the following context, we denote the list of true (estimated) change points by τ^0 ($\hat{\tau}$), the list of true (estimated) parameter-change points by τ^{p0} ($\hat{\tau}^p$), and the candidate sets of β and λ by B and Λ . The two penalty weights can be selected by

$$\beta, \lambda = \underset{\beta \in B, \lambda \in \Lambda}{\operatorname{argmin}} \left\{ \operatorname{Dist}(\tau^0, \hat{\tau}) + \operatorname{Dist}(\hat{\tau}, \tau^0) + \operatorname{Dist}(\tau^{p0}, \hat{\tau}^p) + \operatorname{Dist}(\hat{\tau}^p, \tau^{p0}) \right\}, \quad (16)$$

where $\operatorname{Dist}(\mathbf{a}, \mathbf{b}) = \sum_i \min_j |a_i - b_j|$ measures the total distance of each a_i in \mathbf{a} to the nearest b_j in \mathbf{b} . In Equation (16), smaller $\operatorname{Dist}(\tau^0, \hat{\tau})$ and $\operatorname{Dist}(\tau^{p0}, \hat{\tau}^p)$ correspond to a lower missing detection rate, whereas smaller $\operatorname{Dist}(\hat{\tau}, \tau^0)$ and $\operatorname{Dist}(\hat{\tau}^p, \tau^{p0})$ correspond to a lower false detection rate.

Note that a weighted combination of the four distances can also be used for penalty weights selection, and as discussed in Section S5 of the [Supplementary Materials](#), using equal weights as in Equation (16) can achieve relatively balanced and robust performance.

4. Simulation study and real case study

To verify the effectiveness of the proposed method, a simulation study and a real case study are conducted based on the torque control process of a wind turbine. We introduce the basic settings in Section 4.1 and then present the results and analysis of the simulation study in Section 4.2 and the real case study in Section 4.3.

Table 3. Results of the simulation study (boldface indicates the best, and “/” indicates not applicable).

| Performance measure | σ | C = 3 | | | C = 4 | | | C = 5 | | |
|-------------------------------------|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | 0.003 | 0.006 | 0.012 | 0.003 | 0.006 | 0.012 | 0.003 | 0.006 | 0.012 |
| Precision _{cp} (%) | Proposed | 99.83 | 99.19 | 93.04 | 99.83 | 98.81 | 93.40 | 99.89 | 97.57 | 93.02 |
| | TICC | 26.46 | 25.93 | 27.68 | 28.83 | 21.57 | 23.96 | 28.67 | 24.93 | 26.54 |
| | SBP | 98.32 | 93.91 | 85.95 | 97.66 | 93.07 | 74.33 | 97.38 | 94.95 | 79.96 |
| | DSSL | 24.13 | 21.37 | 21.95 | 21.81 | 22.11 | 21.81 | 21.03 | 22.61 | 23.41 |
| | MCP | 96.45 | 94.91 | 83.86 | 98.40 | 93.75 | 85.04 | 98.50 | 95.44 | 84.20 |
| | RBM | 40.06 | 35.42 | 29.00 | 38.20 | 32.20 | 28.08 | 37.29 | 32.49 | 25.96 |
| Recall _{cp} (%) | Proposed | 99.67 | 98.74 | 92.82 | 99.46 | 98.80 | 93.24 | 99.77 | 98.52 | 92.66 |
| | TICC | 25.99 | 27.14 | 25.66 | 27.68 | 22.07 | 24.00 | 26.62 | 27.41 | 27.93 |
| | SBP | 98.90 | 89.92 | 84.26 | 98.42 | 92.70 | 86.94 | 97.52 | 94.62 | 84.50 |
| | DSSL | 30.57 | 46.04 | 60.94 | 76.72 | 66.04 | 48.02 | 46.11 | 69.74 | 66.16 |
| | MCP | 99.06 | 94.70 | 85.53 | 98.77 | 93.83 | 86.87 | 98.18 | 95.22 | 85.34 |
| | RBM | 93.53 | 90.18 | 80.43 | 94.49 | 84.84 | 84.17 | 93.50 | 89.00 | 70.93 |
| Precision _{p-cp} (%) | Proposed | 99.17 | 97.17 | 83.51 | 98.85 | 97.75 | 91.45 | 97.11 | 94.75 | 95.65 |
| | Other | / | / | / | / | / | / | / | / | / |
| Recall _{p-cp} (%) | Proposed | 95.00 | 92.67 | 87.00 | 92.00 | 91.00 | 81.00 | 96.80 | 93.00 | 82.40 |
| | Other | / | / | / | / | / | / | / | / | / |
| Mode accuracy (%) | Proposed | 98.02 | 97.48 | 95.87 | 98.12 | 97.49 | 96.09 | 98.10 | 97.56 | 96.02 |
| | TICC | 74.66 | 73.38 | 75.42 | 74.07 | 72.84 | 74.13 | 73.96 | 73.82 | 73.35 |
| | RBM | 88.34 | 85.52 | 81.29 | 88.48 | 84.83 | 82.75 | 87.40 | 86.01 | 78.98 |
| | Other | / | / | / | / | / | / | / | / | / |
| Parameter RMSE ($\times 10^{-2}$) | Proposed | 0.039 | 0.082 | 0.164 | 0.056 | 0.084 | 0.185 | 0.044 | 0.088 | 0.204 |
| | RBM | 0.239 | 0.381 | 0.701 | 0.350 | 0.635 | 0.724 | 0.314 | 0.465 | 0.928 |
| | Other | / | / | / | / | / | / | / | / | / |

4.1. Basic settings of the torque control process

As introduced in Section 1, there are $M = 5$ modes in the torque control process, with the mode set defined as $\mathcal{M} = \{1, 2, 3, 4, 5\}$. The mode transition matrix is

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The scaled generator speed and torque at time t are denoted by Y_{1t} and Y_{2t} . Based on engineering knowledge of the wind turbine under study, the relationships, parameters, and cost functions of the five modes are summarized in Table 2, where $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ represents the random noise, $\theta^{(i)}$ denotes the parameter, and $\Theta^{(i)}$ denotes the parameter space for $i \in \mathcal{M}$. The cost functions are set to the sum of squared errors. As shown in Table 2, the parameter of the starting mode is always $\theta^{(1)} = 0$, while the remaining four modes could change their parameters within an interval.

4.2. Simulation study

To systematically test our method, we consider nine simulated cases and make comparisons with five state-of-the-art methods. The simulated data are generated with the following specifications:

- The sample length is set to $T = 1000$ with $p = 2$ variables (i.e., generator speed and torque).
- The maximum number of change points is set to $K_{\max} = 19$, and the possible positions of a change point are $kT/(K_{\max} + 1) = 50k$ for $k = 1, \dots, 19$.

- To represent different sparsity of parameter-change points, we consider three levels of the number of parameter-change points as $C \in \{3, 4, 5\}$.
- For the random noise $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, $i \in \mathcal{M}$, we compute from real wind turbine data that $\sigma_1 : \sigma_2 : \sigma_3 : \sigma_4 : \sigma_5 \approx 0.07 : 1 : 0.98 : 1.27 : 1.47$, so we use $\sigma_1 = 0.07\sigma$, $\sigma_2 = \sigma$, $\sigma_3 = 0.98\sigma$, $\sigma_4 = 1.27\sigma$, and $\sigma_5 = 1.47\sigma$. To represent different noise intensities, we consider three levels as $\sigma \in \{0.003, 0.006, 0.012\}$, where $\sigma = 0.003$ represents the noise level in real data.

Thus, there are a total of nine different simulated cases (three levels of $C \times 3$ levels of σ). For each case, we randomly generate 10 samples for tuning and 100 samples for testing. The data generation process is detailed in Section S6 of the [Supplementary Materials](#).

Figure 5 shows a simulated sample with $C = 4$ and $\sigma = 0.006$. For the time series plot in Figure 5(a), four parameter-change points are indicated by vertical dashed lines, and 15 mode-transition points are marked by vertical solid lines. As shown in the scatter plot in Figure 5(b), mode 1 involves no parameter change, while the remaining four modes change their parameters exactly once in this simulated sample.

We make comparisons with five alternative methods, as introduced below:

1. [Toeplitz inverse covariance-based clustering method](#) (TICC; [Hallac et al., 2017](#); [Zhao and Zhao 2022](#)): This is a multimode process modeling and monitoring method that models each mode with a Markov random field.
2. [Sequential Bayesian partition method](#) (SBP; [Wu et al., 2019](#)): This is a Bayesian multiple change point detection method detecting distributional changes with varying parameters.

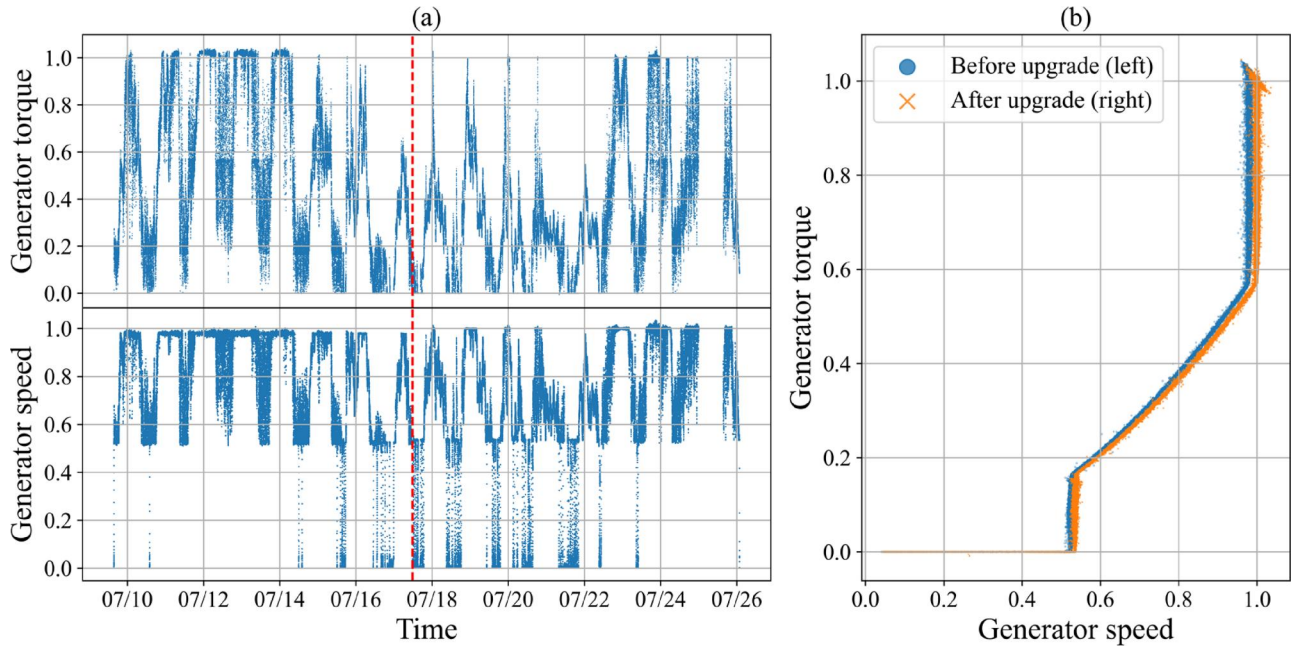


Figure 6. (a) Time series plot and (b) scatter plot of generator torque and speed for a real wind turbine during July 9–26, 2021. The two variables have been scaled by their rated values, the dashed line in (a) indicates the upgrade time, and data points during standby/downtime have been removed in (b).

3. Dynamic sparse subspace learning method (DSSL; Xu *et al.*, 2023): This is a non-Bayesian multiple change point detection method that detects changes in linear manifold structure, which achieves better performance than Zhang *et al.* (2021) that applies the fused LASSO penalty.
4. Mean and covariance change model with the PELT method (MCP; Killick *et al.*, 2012; Wang and Zou, 2023): This is a non-Bayesian multiple change point detection method that detects distributional changes with varying parameters. The implementation details are introduced in Section S7 of the [Supplementary Materials](#).
5. Rule-based method (RBM): This method is commonly employed in the wind turbine industry. The “if-then” rules are designed based on a similar idea to the binning method (Ding, 2019). The details are provided in Section S8 of the [Supplementary Materials](#).

The performance of each method is evaluated in three aspects. First, for the accuracy of change point detection, we consider the precision and recall of all change points:

$$\text{Precision}_{\text{cp}} = \frac{1}{\hat{K}} \sum_{i=1}^{\hat{K}} \mathbb{I} \left(\inf_j |\hat{\tau}_i - \tau_j| \leq s \right),$$

$$\text{Recall}_{\text{cp}} = \frac{1}{K} \sum_{j=1}^K \mathbb{I} \left(\inf_i |\hat{\tau}_i - \tau_j| \leq s \right),$$

where $\hat{\tau}_1, \dots, \hat{\tau}_{\hat{K}}$ are the estimated change points, τ_1, \dots, τ_K are the true change points, and s is the detection error bound, which is set to $s = 5$. In practice, accurate detection of parameter-change points is especially important; thus, we also compute the precision and recall of all parameter-change points:

$$\text{Precision}_{\text{p-cp}} = \frac{1}{\hat{C}} \sum_{i=1}^{\hat{C}} \mathbb{I} \left(\inf_j |\hat{\tau}_i^{\text{p}} - \tau_j^{\text{p}}| \leq s \right),$$

$$\text{Recall}_{\text{p-cp}} = \frac{1}{C} \sum_{j=1}^C \mathbb{I} \left(\inf_i |\hat{\tau}_i^{\text{p}} - \tau_j^{\text{p}}| \leq s \right),$$

where $\hat{\tau}_1^{\text{p}}, \dots, \hat{\tau}_{\hat{C}}^{\text{p}}$ are the estimated parameter-change points, and $\tau_1^{\text{p}}, \dots, \tau_C^{\text{p}}$ represents the true values.

Second, for the accuracy of mode identification, we compute the mode accuracy as

$$\text{Mode accuracy} = \frac{1}{T} \sum_{t=1}^T \mathbb{I}(\hat{m}_t = m_t),$$

where \hat{m}_t is the estimated mode at time t , and m_t is the true mode.

Third, for the accuracy of parameter estimation, we compute the Root Mean Square Error (RMSE) of the estimated parameters for data points with truly identified modes:

$$\text{Parameter RMSE} = \sqrt{\frac{\sum_{t=1}^T \mathbb{I}(\hat{m}_t = m_t) \left| \hat{\theta}_t^{(\hat{m}_t)} - \theta_t^{(m_t)} \right|^2}{\sum_{t=1}^T \mathbb{I}(\hat{m}_t = m_t)}},$$

where $\hat{\theta}_t^{(\hat{m}_t)}$ is the estimated parameter at time t for mode \hat{m}_t , and $\theta_t^{(m_t)}$ is the true value.

In the tuning process, the same 10 samples are used for each method. For our method, we select the two penalty weights by Equation (16) with $\beta \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$ and $\lambda \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. For DSSL, the two penalty weights λ_1, λ_2 are selected using the approach provided by Xu *et al.* (2023). For the remaining four methods, as they do not solve for the parameter-change point, their tuning processes only consider the first two terms of Equation (16). Specifically,

Table 4. Results of the real case study (boldface indicates the best, and "/" indicates not applicable).

| | $Precision_{cp}$ (%) | $Recall_{cp}$ (%) | $Precision_{p-cp}$ (%) | $Recall_{p-cp}$ (%) | Mode accuracy (%) | Parameter RMSE ($\times 10^{-2}$) |
|----------|----------------------|-------------------|------------------------|---------------------|-------------------|-------------------------------------|
| Proposed | 99.64 | 74.12 | 100.00 | 100.00 | 96.27 | 0.0327 |
| TICC | 52.16 | 25.62 | / | / | 58.45 | / |
| SBP | 66.08 | 51.36 | / | / | / | / |
| DSSL | 36.34 | 90.26 | / | / | / | / |
| MCP | 61.85 | 41.43 | / | / | / | / |
| RBM | 57.23 | 54.64 | / | / | 86.50 | 0.2425 |

for TICC, we set the number of clusters as the number of modes in the torque control process, i.e., $K = 5$, and select the window size w and two penalty weights λ, β with $w \in \{1, 3, 5\}$, $\lambda \in \{0.0001, 0.001, 0.01, 0.1\}$ and $\beta \in \{10, 20, 30, 40, 50\}$. For SBP, according to the recommendation in Wu *et al.* (2019), we set $p_0 = 0.1$, $\mu_0 = (0, 0)'$, $\Psi_0 = 0.003^2 \nu_0 \mathbf{I}_2$ (\mathbf{I}_2 is a two-dimensional identity matrix), and select ν_0, γ_0 with $\nu_0 \in \{2, 5, 10, 15, 20, 25, 30, 35, 40, 45\}$ and $\gamma_0 \in \{10^{-16}, 10^{-13}, 10^{-10}, 10^{-7}, 10^{-4}, 10^{-1}\}$. For MCP, we select the penalty weight with $\beta \in \{25, 50, 75, 100, 125, 150, 175, 200, 350, 500\}$. For RBM, we select the three thresholds with $thre_1 \in \{0.04, 0.05, 0.06\}$, $thre_2 \in \{5, 6, 7\}$, and $thre_3 \in \{3, 4, 5\}$. After the tuning process, we use another 100 samples for testing, and the average performance measures are reported in Table 3.

As shown in Table 3, we horizontally observe that as the noise level σ increases from 0.003 to 0.012, the performance of the proposed method worsens. One possible explanation is that with a higher noise level, the boundary between two adjacent modes becomes more obscure, which increases the difficulty of change detection. Unlike the noise level, the number of parameter-change points C seems to have minimal influence on performance.

On the other hand, by vertically comparing the results, we observe that our method outperforms the state-of-the-art methods in all cases. The poor performance of TICC and DSSL may be due to the mismatch of the data with their model assumptions. For TICC, the parameter of each mode is assumed to be fixed, whereas in our setting, the parameter can change to a new value. For DSSL, the assumed linear relationship may not well model the five modes of the torque control process. With an increase in σ , the performance of our method declines more slowly than SBP, MCP, and RBM, indicating that our method is more robust to higher noise levels. In terms of change point detection, only our method can detect parameter-change points, which is important for real applications. Because of this advantage, our method could more accurately estimate parameters. In addition, our method converges rapidly among all cases, with 3.06 iterations on average and seven iterations at most. Therefore, it is speculated that our method is more suitable for modeling multimode processes with mode transitions and parameter changes than the state-of-the-art methods.

4.3. Real case study

The SCADA data are derived from a 2MW direct drive wind turbine, and the sampling interval is 8 seconds. During 2021, this wind turbine experienced two control software

upgrades. The first upgrade occurred on January 9, as shown in Figure 2. The second upgrade occurred on July 17 and is shown in Figure 6. We use data covering the first upgrade (2021/1/4 16:44:05 ~ 2021/1/12 8:04:58) for tuning and use data covering the second upgrade (2021/7/9 15:30:18 ~ 2021/7/26 1:41:57) for testing.

For data preprocessing, we scale the generator speed and torque by their rated values and then remove data points collected during standby/downtime. After preprocessing and manual evaluation, there are 51,294 data points in the tuning data, with 2043 change points and four parameter-change points, and there are 148,872 data points in the testing data, with 6213 change points and three parameter-change points.

The tuning process is the same as in the simulation study. For our method, the two penalty weights are selected as $\beta = 0.0001$ and $\lambda = 0.1$, and it takes six iterations to converge. The testing results of the real case study are reported in Table 4.

From Table 4, we can see that our method achieves the best performance among the six methods for performance measures except for $Recall_{cp}$. Although DSSL achieves the highest $Recall_{cp}$ of 90.26%, its false detection rate is very high, with a precision of only 36.34%, which is not acceptable in practice. Moreover, our method accurately detects all parameter-change points, with 100% precision and 100% recall. Therefore, it is speculated that our method is more adaptive to real data than the state-of-the-art methods and has better comprehensive performance. In addition, we observe that $Recall_{cp}$ is relatively lower than $Precision_{cp}$ for methods other than DSSL, which is different from the simulation study. For possible causes, we discover that approximately half of all subsegments in the testing data are of length five or less, so the information contained in these subsegments is lower. As a result, it is relatively difficult to accurately distinguish these subsegments from adjacent subsegments, leading to a lower recall of all change points.

5. Conclusion

In this article, we propose a novel modeling framework for the offline change point detection problem of multimode processes, which simultaneously considers mode transitions and parameter changes. To improve interpretability, we incorporate engineering knowledge into the model by specifying the number of modes, mode transition matrix, and cost functions. Then, we formulate the problem as a constrained optimization model with two penalty terms. The asymptotic property shows that under certain assumptions, the optimal solution of the model will converge to the

ground truth as the sampling resolution goes to infinity. To solve the optimization model, we propose an iterative algorithm and develop a multi-PELT method for initialization. Based on the torque control process of a wind turbine, both the simulation study and the real case study demonstrate the effectiveness of the proposed method for change point detection, mode identification, and parameter estimation.

This work can be extended in several directions. First, the proposed method does not consider the detailed transition process between two different modes, because it is nearly transient in our example of the torque control process. However, for multimode processes that transition gradually from one mode to another mode, the transition process should be characterized in detail. Second, except for mode transition availability, there might exist other constraints between two different modes, which can be considered to help refine the model. Last, online monitoring and diagnosis methods are worthy of further investigation for multimode processes with mode transitions and parameter changes.

Acknowledgments

The authors would like to thank the Department Editor, the Associate Editor and anonymous referees for their valuable and helpful comments and suggestions.

Funding

This work was supported by the Key Program of the National Natural Science Foundation of China under Grant No. 71932006.

Notes on contributors

Jun Xu is currently a Ph.D. student in the Department of Industrial Engineering, Tsinghua University. He received his B.Eng. degree in Industrial Engineering from Tsinghua University in 2019. His research interests include modeling, monitoring and diagnosis of complex systems.

Jie Zhou is a senior engineer in Goldwind Science & Technology Co., Ltd, Beijing, China. His work focuses on wind turbine diagnosis and safety control. He is also currently working towards the D.Eng. degree in Industrial Engineering with Tsinghua University, Beijing, China. He received his B.S. and M.S. degrees in Electrical Engineering from Dalian University of Technology, Dalian, China.

Xiaofang Huang is a senior engineer. She received her master's degree from Xidian University, Xi'an, China in 2006. She is currently a department lead of the R&D Center of Goldwind Science & Technology Co.,Ltd, mainly engaged in the development and localization of wind turbine main control system software, as well as the development of intelligent control and protection technology of wind turbines and wind farms.

Kaibo Wang is professor in the Department of Industrial Engineering, Tsinghua University, jointly appointed by Vanke School of Public Health, Tsinghua University. He received his Ph.D. in Industrial Engineering and Engineering Management from Hong Kong University of Science and Technology, Hong Kong, in 2006. His research focuses on statistical quality control and data-driven system modeling, monitoring, diagnosis and control, with a special emphasis on the integration of engineering knowledge and statistical theories.

ORCID

Jun Xu  <http://orcid.org/0000-0001-9918-6148>

Kaibo Wang  <http://orcid.org/0000-0001-9888-4323>

Data availability statement

The participants of this study did not give written consent for their data to be shared publicly, so due to the nature of the research supporting data is not available.

References

- Burton, T., Jenkins, N., Bossanyi, E., Sharpe, D. and Graham, M. (2021) *Wind Energy Handbook*, third edition, John Wiley & Sons, Hoboken, NJ.
- Chen, Z., Zhou, D., Zio, E., Xia, T. and Pan, E. (2023) Adaptive transfer learning for multimode process monitoring and unsupervised anomaly detection in steam turbines. *Reliability Engineering and System Safety*, **234**, 109162.
- Ding, Y. (2019) *Data Science for Wind Energy*, CRC Press, Boca Raton, FL.
- Ding, Y., Kumar, N., Prakash, A., Kio, A.E., Liu, X., Liu, L. and Li, Q. (2021) A case study of space-time performance comparison of wind turbines on a wind farm. *Renewable Energy*, **171**, 735–746.
- Du, W., Fan, Y. and Zhang, Y. (2017) Multimode process monitoring based on data-driven method. *Journal of the Franklin Institute*, **354**(6), 2613–2627.
- Haghani, A., Krueger, M., Jeansch, T., Ding, S.X. and Engel, P. (2015) Data-driven multimode fault detection for wind energy conversion systems. *IFAC-PapersOnLine*, **48**(21), 633–638.
- Hallac, D., Vare, S., Boyd, S. and Leskovec, J. (2017) Toeplitz inverse covariance-based clustering of multivariate time series data, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*, ACM, New York, NY, pp. 215–223.
- Jackson, B., Scargle, J.D., Barnes, D., Arabhi, S., Alt, A., Gioumoussis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L. and Tsai, T.T. (2005) An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, **12**(2), 105–108.
- Jin, H., Yin, G., Yuan, B. and Jiang, F. (2022) Bayesian hierarchical model for change point detection in multivariate sequences. *Technometrics*, **64**(2), 177–186.
- Jin, R. and Liu, K. (2013) Multimode variation modeling and process monitoring for serial-parallel multistage manufacturing processes. *IIE Transactions*, **45**(6), 617–629.
- Juodakis, J. and Marsland, S. (2023) Epidemic changepoint detection in the presence of nuisance changes. *Statistical Papers*, **64**, 17–39.
- Killick, R., Fearnhead, P. and Eckley, I.A. (2012) Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, **107**(500), 1590–1598.
- Ko, S.I.M., Chong, T.T.L. and Ghosh, P. (2015) Dirichlet process hidden Markov multiple change-point model. *Bayesian Analysis*, **10**(2), 275–296.
- Lee, G., Ding, Y., Genton, M.G. and Xie, L. (2015) Power curve estimation with multivariate environmental factors for inland and offshore wind farms. *Journal of the American Statistical Association*, **110**(509), 56–67.
- Prakash, A., Tuo, R. and Ding, Y. (2022) Gaussian process-aided function comparison using noisy scattered data. *Technometrics*, **64**(1), 92–102.
- Prakash, A., Tuo, R. and Ding, Y. (2023) The temporal overfitting problem with applications in wind power curve modeling. *Technometrics*, **65**(1), 70–82.
- Quiñones-Grueiro, M., Prieto-Moreno, A., Verde, C. and Llanes-Santiago, O. (2019) Data-driven monitoring of multimode

- continuous processes: A review. *Chemometrics and Intelligent Laboratory Systems*, **189**, 56–71.
- Shi, J. (2023) In-process quality improvement: Concepts, methodologies, and applications. *IIE Transactions*, **55**(1), 2–21.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society, Series B*, **67**, 91–108.
- Truong, C., Oudre, L. and Vayatis, N. (2020) Selective review of offline change point detection methods. *Signal Processing*, **167**, 107299.
- Wang, A., Chang, T.-S. and Shi, J. (2022) Multiple event identification and characterization by retrospective analysis of structured data streams. *IIE Transactions*, **54**(9), 908–921.
- Wang, F., Tan, S. and Shi, H. (2015) Hidden Markov model-based approach for multimode process monitoring. *Chemometrics and Intelligent Laboratory Systems*, **148**, 51–59.
- Wang, G. and Zou, C. (2023) cps: An R package for change-point detection by sample-splitting methods. *Journal of Quality Technology*, **55**(1), 61–74.
- Wu, J., Chen, Y. and Zhou, S. (2016) Online detection of steady-state operation using a multiple-change-point model and exact Bayesian inference. *IIE Transactions*, **48**(7), 599–613.
- Wu, J., Chen, Y., Zhou, S. and Li, X. (2016) Online steady-state detection for process control using multiple change-point models and particle filters. *IEEE Transactions on Automation Science and Engineering*, **13**(2), 688–700.
- Wu, J., Xu, H., Zhang, C. and Yuan, Y. (2019) A sequential Bayesian partitioning approach for online steady-state detection of multivariate systems. *IEEE Transactions on Automation Science and Engineering*, **16**(4), 1882–1895.
- Wu, Z., Li, Y., Tsung, F. and Pan, E. (2023) Real-time monitoring and diagnosis scheme for IoT-enabled devices using multivariate SPC techniques. *IIE Transactions*, **55**(4), 348–362.
- Xie, X. and Shi, H. (2012) Multimode process monitoring based on fuzzy C-means in locality preserving projection subspace. *Chinese Journal of Chemical Engineering*, **20**(6), 1174–1179.
- Xie, Y. (2020) Double-weighted neighborhood standardization method with applications to multimode-process fault detection. *Journal of Intelligent & Fuzzy Systems*, **39**(1), 1243–1256.
- Xu, R., Wu, J., Yue, X. and Li, Y. (2023) Online structural change-point detection of high-dimensional streaming data via dynamic sparse subspace learning. *Technometrics*, **65**(1), 19–32.
- Xu, X., Ding, J., Liu, Q. and Chai, T. (2021) A novel multi-manifold joint projections model for multimode process monitoring. *IEEE Transactions on Industrial Informatics*, **17**(9), 5961–5970.
- Xu, X., Qin, F., Zhao, W., Xu, D., Wang, X. and Yang, X. (2022) Anomaly detection with GRU based Bi-autoencoder for industrial multimode process. *International Journal of Control, Automation and Systems*, **20**(6), 1827–1840.
- Yang, J., Zhang, M., Shi, H. and Tan, S. (2017) Dynamic learning on the manifold with constrained time information and its application for dynamic process monitoring. *Chemometrics and Intelligent Laboratory Systems*, **167**, 179–189.
- Yin, M., Li, W., Chung, C.Y., Zhou, L., Chen, Z. and Zou, Y. (2017) Optimal torque control based on effective tracking range for maximum power point tracking of wind turbines under varying wind conditions. *IET Renewable Power Generation*, **11**(4), 501–510.
- Yu, F., Liu, J. and Liu, D. (2022) Multimode process monitoring based on modified density peak clustering and parallel variational autoencoder. *Mathematics*, **10**(14), 2526.
- Zhang, C., Yan, H., Lee, S. and Shi, J. (2021) Dynamic multivariate functional data modeling via sparse subspace learning. *Technometrics*, **63**, 370–383.
- Zhang, J., Zhou, D. and Chen, M. (2023) Self-learning sparse PCA for multimode process monitoring. *IEEE Transactions on Industrial Informatics*, **19**(1), 29–39.
- Zhao, Y. and Zhao, C. (2022) Dynamic multivariate threshold optimization and alarming for nonstationary processes subject to varying conditions. *Control Engineering Practice*, **124**, 105180.
- Zou, C., Yin, G., Feng, L. and Wang, Z. (2014) Nonparametric maximum likelihood approach to multiple change-point problems. *Annals of Statistics*, **42**(3), 970–1002.