Truncated Online Dynamic Mode Decomposition with Control for Industrial Change Detection

Marek Wadinger*, Michal Kvasnica*, and Yoshinobu Kawahara†
*Slovak University of Technology in Bratislava, Slovakia
{marek.wadinger, michal.kvasnica}@stuba.sk
†Osaka University, and RIKEN Center for Advanced Intelligence Project, Japan
kawahara@ist.osaka-u.ac.jp

***Abstract*—Modern control systems require real-time monitoring to ensure safety and reliability, yet detecting changes in system behavior remains challenging in nonlinear, high-dimensional, and time-varying environments. We introduce an interpretable change-point detection (CPD) framework based on truncated online Dynamic Mode Decomposition with control (toDMDc). The method combines user-specified rank truncation with online system identification, enabling real-time adaptation to evolving dynamics while maintaining numerical stability. By comparing reconstruction errors between reference and test windows, the framework detects changes in system behavior. We validate the approach on three case studies: synthetic step changes, a nonlinear two-tank system with input delays, and an industrial battery energy storage system. Results show that toDMDc-based detection achieves accurate change-point identification with interpretable statistics, bounded detection delays, and computational efficiency suitable for real-time deployment in safety-critical control applications.**

***Index Terms*—Change-point detection, Dynamic mode decomposition, System identification, Fault detection, Online learning, Control systems.**

I. Introduction

REAL-TIME monitoring of industrial control systems is essential for ensuring safety, reliability, and optimal performance. The ability to detect abrupt or gradual changes in system behavior—known as change-point detection (CPD)—enables timely intervention before failures occur. However, modern industrial systems present significant challenges: they are often nonlinear, high-dimensional, subject to external control inputs, and exhibit time-varying dynamics due to aging, environmental fluctuations, and varying operating conditions.

Traditional statistical process control (SPC) methods [1] assume independent and identically distributed (i.i.d.) data, which is rarely satisfied in industrial systems where measurements are temporally correlated and non-stationary. SCADA-based systems use static thresholds that cannot adapt to evolving system behavior. While supervised machine learning methods achieve high accuracy with labeled data [2], they require extensive historical annotations and fail when encountering novel patterns—limitations that are impractical in existing industrial infrastructures.

A. Subspace-Based Change-Point Detection

Subspace-based CPD methods address these limitations by modeling the system's underlying low-dimensional structure without distributional assumptions [3], [4]. The key insight is that normal system behavior evolves within a low-rank subspace, while change-points manifest as deviations that cannot be captured by this subspace. Methods based on Singular Value Decomposition (SVD) [5] and subspace identification [6] have shown promise, but most are designed for autonomous systems without explicit consideration of control inputs.

For controlled systems, Dynamic Mode Decomposition (DMD) [7] offers a powerful framework. DMD decomposes time-series data into coherent spatial-temporal modes, revealing both spatial structures (like SVD) and temporal dynamics (like Fourier analysis). Extensions to incorporate control inputs (DMDc) [8] enable system identification that disambiguates between natural dynamics and the effects of actuation—critical for industrial applications where control actions significantly influence system behavior.

B. Challenges in Online DMD for Change-Point Detection

While DMD provides interpretable models, its application to online CPD faces several challenges:

(1) Time-varying dynamics: Industrial systems evolve over time due to aging, seasonal effects, and shifting operating conditions. A static DMD model trained offline becomes invalid, necessitating continuous adaptation.

(2) Numerical stability: Online DMD updates [9] can suffer from numerical instability when eigenvalues are small, and the computational cost scales poorly with dimension without rank reduction.

(3) Rank selection: Selecting the appropriate subspace rank is critical—too low and important dynamics are lost; too high and noise corrupts the model. This selection must be performed as system characteristics evolve.

(4) Detection with adaptation: The framework must simultaneously detect change-points while updating the model, avoiding false negatives when transient dynamics appear.

This paper is supported by an internal STU grant for teams of young researchers under the project OptiVision. The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant VEGA 1/0490/23. Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I01–03–V04–00024.

C. Contributions

This paper addresses these challenges through three key contributions:

Truncated online DMD with control (toDMDc): We integrate user-specified rank truncation with online DMDc updates, maintaining numerical stability while adapting to time-varying dynamics. The truncation is performed using an efficient online SVD algorithm [10] with guaranteed orthogonality preservation.

Change-point detection framework: We utilize a detection statistic based on reconstruction error comparison between base and test windows. The statistic provides interpretable measures of change severity with bounded detection delays equal to the test window size.

Experimental validation: We validate the framework on three case studies representing key industrial scenarios: (i) synthetic step changes for delay validation, (ii) a nonlinear two-tank system with input delays showcasing robustness to control actions and non-stationarity, and (iii) an industrial battery energy storage system detecting real fault transitions.

The remainder of this paper is organized as follows: Section II provides background on DMD and online extensions; Section III presents the toDMDc-based CPD framework; Section IV validates the approach experimentally; Section V discusses the method and its deployment; and Section VI concludes with future directions.

II. Preliminaries

A. Dynamic Mode Decomposition

Consider a sequence of state measurements $\{x_0, x_1, \ldots, x_n\} \in \mathbb{R}^m$ collected at times $\{t_0, t_1, \ldots, t_n\}$. DMD seeks a linear operator A that best advances the system in time:

$$A = \arg\min_A \|X' - AX\|_F, \tag{1}$$

where $X = [x_0, x_1, \ldots, x_{n-1}] \in \mathbb{R}^{m\times n}$ and $X' = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m\times n}$. The solution is $A = X'X^+$, where X^+ denotes the Moore-Penrose pseudoinverse.

For high-dimensional systems ($m \gg 1$), computing A directly is expensive. DMD leverages the singular value decomposition $X = U\Sigma V^\top$ to obtain a reduced-order representation:

$$\tilde{A} = U^\top X' V \Sigma^{-1} \in \mathbb{R}^{r\times r}, \tag{2}$$

where $r \leq m$ is the truncation rank. Eigendecomposition of $\tilde{A}$ yields DMD eigenvalues Λ and eigenvectors W, from which full-dimensional DMD modes are reconstructed as:

$$\Phi = X' V \Sigma^{-1} W \in \mathbb{R}^{m\times r}. \tag{3}$$

Each DMD mode ϕ_i oscillates with frequency and growth rate encoded in the corresponding discrete-time eigenvalue λ_i, where the magnitude $|\lambda_i|$ governs growth/decay ($|\lambda_i| > 1$ growth, $|\lambda_i| < 1$ decay) and the argument $\arg(\lambda_i)/\Delta t$ governs the oscillation frequency for a sampling interval Δt [11].

B. DMD with Control

For controlled systems, the dynamics follow:

$$x_{k+1} = Ax_k + B\theta_k, \tag{4}$$

where $\theta_k \in \mathbb{R}^l$ is the control input and $B \in \mathbb{R}^{m\times l}$ is the control matrix.

When B is known, we compensate the output snapshots:

$$\bar{X}' = X' - B\Theta, \tag{5}$$

where $\Theta = [\theta_0, \ldots, \theta_{n-1}]$, and apply standard DMD to $\{X, \bar{X}'\}$.

When B is unknown (the typical case), we augment the state with control inputs:

$$\bar{X} = \begin{bmatrix} X \\ \Theta \end{bmatrix} \in \mathbb{R}^{(m+l)\times n}, \tag{6}$$

and solve for the augmented operator $\bar{A} = [A \mid B]$ that maps $\bar{X}$ to X' [8].

C. Online DMD with Control

For streaming data, we update the DMD model incrementally without storing the entire history. Following [9], we express (1) as a recursive least-squares problem:

$$A_k = Q_k P_k, \tag{7}$$

where $Q_k = X'_k X_k^\top$ and $P_k = (X_k X_k^\top)^{-1}$ are the lag-covariance and precision matrices.

Upon receiving new snapshot pairs $\{X_{k:k+c}, X'_{k:k+c}\}$, we update:

$$P_{k+c} = P_k - P_k X_{k:k+c} \Gamma_{k+c} X_{k:k+c}^\top P_k, \tag{8}$$

$$A_{k+c} = A_k + (X'_{k:k+c} - A_k X_{k:k+c}) \Gamma_{k+c} X_{k:k+c}^\top P_k, \tag{9}$$

where $\Gamma_{k:k+c} = (C_{k:k+c}^{-1} + X_{k:k+c}^\top P_k X_{k:k+c})^{-1}$ and C is a diagonal weight matrix. To forget old snapshots (windowed online DMD), we apply (8)–(9) with negative weights $-C$.

D. Truncated Online DMD

To maintain numerical stability and computational efficiency, we integrate rank-r truncation into online DMD using online SVD [10], [12]. Let $\bar{X}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ be the truncated SVD of the augmented snapshot matrix. We project snapshots onto the first r POD modes:

$$\{\tilde{\bar{X}}_k, \tilde{X}'_k\} = \{\tilde{U}_k^\top \bar{X}_k, \tilde{U}_k^\top X'_k\}. \tag{10}$$

As new data arrives, the online SVD is updated efficiently via rank-c modifications. The change in the column space is tracked via $K_k^{U'U} = \tilde{U}_k^\top \tilde{U}_{k-1}$. To align the reduced-order DMD matrix $\tilde{\bar{A}}_k$ and precision matrix $\tilde{P}_k$ with the new basis, we decompose the rotation matrix:

$$K_k^{U'U} = \begin{bmatrix} K_{k,p\times p}^{U'U} & K_{k,p\times q}^{U'U} \\ K_{k,q\times p}^{U'U} & K_{k,q\times q}^{U'U} \end{bmatrix}, \tag{11}$$

where p and q are the ranks of the state and control subspaces. The reduced-order matrices are then transformed:

$$\tilde{\bar{A}}_k = \left[K_{k,pp}^{U'U} \tilde{A}_k (K_{k,pp}^{U'U})^\top \quad K_{k,pp}^{U'U} \tilde{B}_k K_{k,qq}^{U'U} \right], \tag{12}$$

$$\tilde{P}_k = (K_k^{U'U} P_k^{-1} (K_k^{U'U})^\top)^{-1}. \tag{13}$$

Finally, we apply updates (8)–(9) in the reduced space using the projected snapshots $\{\tilde{\bar{X}}_k, \tilde{X}_k'\}$. This maintains $\mathcal{O}(mr^2)$ complexity per update instead of $\mathcal{O}(m^3)$, making real-time deployment feasible for high-dimensional systems.

III. Change-Point Detection Framework

A. Mathematical Formulation

A change-point occurs when incoming data cannot be adequately reconstructed by the current subspace, indicating a shift in underlying system dynamics. We measure this using the orthonormal POD basis $\tilde{U}_k$ computed in Section II, whose columns span the rank-r subspace of normal behavior. Since $\tilde{U}_k$ is orthonormal, $\tilde{U}_k\tilde{U}_k^\top$ is the orthogonal projector onto this subspace, and we define the reconstruction error for an arbitrary augmented data matrix $\bar{X}_k$ as:

$$E(\bar{X}_k, \tilde{U}_k) = \|\bar{X}_k - \tilde{U}_k\tilde{U}_k^\top\bar{X}_k\|_F^2. \tag{14}$$

We use the POD basis $\tilde{U}_k$ rather than the DMD modes Φ_k as the projector because the DMD modes are not orthonormal, so $\Phi_k\Phi_k^\top$ would not define an orthogonal projection.

We maintain three sliding windows:

- **Base window** $\bar{X}_{\mathrm{B}}$: Reference data of size a representing normal behavior
- **Test window** $\bar{X}_{\mathrm{T}}$: Recent data of size c being evaluated for changes
- **Learning window** $\bar{X}_{\mathrm{L}}$: Historical data of size d for model updates

The detection statistic is:

$$S_k = \max\left(0, \frac{E(\bar{X}_{\mathrm{T}}, \tilde{U}_k)/c}{E(\bar{X}_{\mathrm{B}}, \tilde{U}_k)/a} - 1\right). \tag{15}$$

Under the null hypothesis (no change), both windows have similar reconstruction errors, yielding $S_k \approx 0$. Under the alternative hypothesis (change present), the test window exhibits significantly higher reconstruction error, resulting in $S_k > 0$. The magnitude of S_k provides an interpretable measure of change severity.

B. Algorithm

The complete framework operates in three stages per iteration:

Stage 1: Data Preprocessing. Incoming snapshots are formed into time-delay embeddings with h delays using Hankelization. For known control matrix B, we compensate X' with control effects. Otherwise, we augment states with controls as in (6). The base, test, and learning windows are extracted from stored history.

Stage 2: Detection. We project base and test matrices onto the rank-r POD subspace using the orthonormal basis $\tilde{U}_k$:

$$\tilde{\bar{X}}_{\mathrm{B}} = \tilde{U}_k^\top\bar{X}_{\mathrm{B}}, \quad \tilde{\bar{X}}_{\mathrm{T}} = \tilde{U}_k^\top\bar{X}_{\mathrm{T}}. \tag{16}$$

Because $\tilde{U}_k$ is orthonormal, the residual energy $\|\bar{X}_k - \tilde{U}_k\tilde{U}_k^\top\bar{X}_k\|_F^2 = \|\bar{X}_k\|_F^2 - \|\tilde{U}_k^\top\bar{X}_k\|_F^2$, so the reconstruction error in (14) is computed directly from these projections. The errors are normalized by window size, and the detection

Algorithm 1 toDMDc-based Change-Point Detection

Input: Snapshot stream $\{x_k, \theta_k\}$, parameters $\{a, b, c, d, h, r\}$
Output: Detection statistic S_k

```
Initialize: Ã_0 ← I, P̃_0 ← αI, history buffer
for each incoming snapshot k do
    // Data Preprocessing
    Form time-delay embeddings: X̄_k ← Hankel(x_k, θ_k, h)
    Extract windows: X̄_B, X̄_T, X̄_L
    // Online SVD Update
    Update truncated SVD: X̄_k = Ũ_k Σ̃_k Ṽ_k^T
    Compute rotation: K_k^{U'U} = Ũ_k^T Ũ_{k-1}
    Align matrices: Ã_k, P̃_k using K_k^{U'U}
    // Change Detection
    Project: X̃_B ← Ũ_k^T X̄_B, X̃_T ← Ũ_k^T X̄_T
    Compute errors: E_B, E_T via (14)
    Compute statistic: S_k ← max(0, (E_T/c)/(E_B/a) − 1)
    if S_k > η then
        Flag change-point
    end if
    // Model Update
    if learning buffer full then
        Revert old snapshots using negative weights
    end if
    Update DMD: Ã_k, P̃_k via (8)–(9)
end for
```

statistic (15) is evaluated. A change-point is flagged if S_k exceeds a threshold η.

Stage 3: Learning. The DMD model is updated using the learning window. If the learning buffer is full, old snapshots are reverted using negative weights. New snapshots are then incorporated via (8)–(9). Importantly, detection precedes learning to avoid false negatives when transient dynamics appear.

The complete algorithm is summarized in Algorithm 1.

C. Properties

Property 1 (Deterministic detection delay): The test window must be fully populated by post-change samples before the reconstruction error of $\bar{X}_{\mathrm{T}}$ reflects the new dynamics. Consequently, for a change at time t_c, the statistic S_k attains its peak no later than time $t_c + c$, where c is the test-window size. The detection delay is therefore deterministically bounded by c, independent of the change magnitude or noise level.

Remark 1 (Convergence to batch least squares): The online updates (8)–(9) implement recursive least squares for (1). Under stationary inputs with persistently exciting data (so that $X_kX_k^\top$ is non-singular), the recursion converges to the batch least-squares DMD solution $A = X'X^+$; this is the standard convergence property of recursive least squares established in [9]. We do not claim a novel convergence rate beyond this result.

TABLE I
SUMMARY OF EXPERIMENTAL VALIDATION CASE STUDIES

Case Study	System	Change Type	Key Challenge
Synthetic	Linear, 1D $n = 10000$	Abrupt steps 9 events	Detection delay validation
Two-tank	Nonlinear, 2D $n = 12000$	Bias, drift, response change	Input delays ($\tau = 20$–30)
BESS	Industrial, 6D Real data	Hardware fault Gradual	Early warning detection

Remark 2 (Interpretability): Unlike black-box methods [13], the detection statistic S_k directly reflects the relative dissimilarity between base and test data in terms of the learned subspace, providing physical interpretability.

D. Parameter Selection Guidelines

The framework requires selection of four key parameters that balance detection performance and computational cost:

Learning window d: Should capture multiple operating cycles (e.g., daily or weekly patterns in industrial systems). For example, in battery systems with daily charging cycles, d should span at least 24 hours. Larger d provides more robust model identification but increases adaptation time.

Time-delay embedding h: Must exceed the system's dominant time constants to capture transient dynamics. A rule of thumb is $h \geq 2\tau_{\max}$, where $\tau_{\max}$ is the largest time constant. For systems with input delays, h should be at least twice the delay.

Base and test windows a, c: Balance detection sensitivity versus delay. Smaller windows detect changes faster but are more sensitive to noise; larger windows improve robustness but delay detection. Typically, $a = c$ is chosen equal to the expected change duration. The separation b between windows can be zero for immediate detection or positive to avoid transient effects.

Rank r: Determines the subspace dimension. Methods like Gavish-Donoho optimal threshold [14] can guide selection, though in practice r is often chosen to capture 95-99% of the signal energy.

Remark 3 (Numerical stability): The truncated SVD promotes numerical stability by discarding small singular values that would otherwise corrupt P_k. The reorthogonalization in [10] keeps the basis near-orthonormal, $\|\tilde{U}_k^\top \tilde{U}_k - I\| < \epsilon$ for a specified tolerance ϵ, preventing drift in the subspace basis.

IV. EXPERIMENTAL VALIDATION

We validate the framework on three case studies representing different change types, system complexities, and industrial scenarios. Table I summarizes the experimental configurations and key characteristics.

All experiments use a user-specified rank truncation ($r =$ 2–10) and windowed online updates with forgetting factors to adapt to time-varying conditions. The threshold η is set per experiment on a calibration window of nominal operation. Throughout this section, figures adopt a common convention: solid vertical lines mark the true occurrence of each change-point at time t_c, while dashed red vertical lines mark the deterministic detection bound $t_c + c$ from Property 1, by which S_k is guaranteed to peak. The horizontal axis in each figure shows the number of samples elapsed since the experiment start.

A. Synthetic Step Changes

Setup: We simulate a univariate signal with nine abrupt changes at intervals of 1000 samples. Each change introduces a step of magnitude $\Delta_i = 0.5i$ for $i = 1, \ldots, 9$, with additive Gaussian noise $\mathcal{N}(0, 1)$. Hyperparameters: $r = 2$, $a = 100$, $c = 100$, $d = 300$, $h = 80$.

Results: Figure 1 shows that toDMDc detects all change-points, although the first two are small relative to the noise. The detection statistic peaks shortly after $c = 100$ samples from the true change-point, consistent with the deterministic delay bound of Property 1, while being subject to noise prominence. Notably, the detection score decreases for later change-points despite larger magnitudes—this reflects that the *relative* dissimilarity (error ratio) from (15) decreases as the signal's absolute energy increases, while the *absolute* dissimilarity between errors (14) grows. This demonstrates that S_k captures the proper notion of change severity relative to the current operating state [13].

Quantitatively, the detection delay for change-points 2–9 is $\approx$ 100 samples, measured at 102 ± 3 samples (mean $\pm$ std), matching the deterministic delay bound $c = 100$. The reference SVD-based method shows 25% higher variance in the detection statistic, potentially masking subtle changes in noisy environments.

B. Nonlinear Two-Tank System with Input Delay

Setup: We simulate a two-tank system governed by:

$$\begin{aligned} \frac{\mathrm{d}h_1}{\mathrm{d}t} &= q(t-\tau) - \frac{k_1}{F_1}\sqrt{h_1}, \\ \frac{\mathrm{d}h_2}{\mathrm{d}t} &= \frac{k_1}{F_2}\sqrt{h_1} - \frac{k_2}{F_2}\sqrt{h_2}, \end{aligned} \tag{17}$$

where h_1, h_2 are tank levels, q is control flow with delay $\tau =$ 20–30 samples, and k_i, F_i are valve constants and areas. The system where random control step occurs every 200 samples experiences: (i) sensor bias from sample 4000–5000, (ii) doubled control response from sample 7600–8600, and (iii) linear drift from sample 9800–12000. Gaussian noise $\mathcal{N}(0, 0.35)$ is added. We use polynomial features (degree 2) to handle nonlinearity and set $d = 2000$, $h = 200$, $h_{\mathrm{d}} = 30$.

Results: Figure 2 demonstrates that toDMDc accurately detects all three change types with low false alarm rate. The sensor bias (sharp change) is detected at sample 4205 (delay: 205 samples $\approx$ c = 200). The doubled control response produces a detection peak at sample 7798 (delay: 198 samples). Crucially, the framework detects the slow linear drift starting at sample 9995 (delay: 195 samples). Meanwhile, the reference SVD-based method shows better recognition of the doubled control response, while making

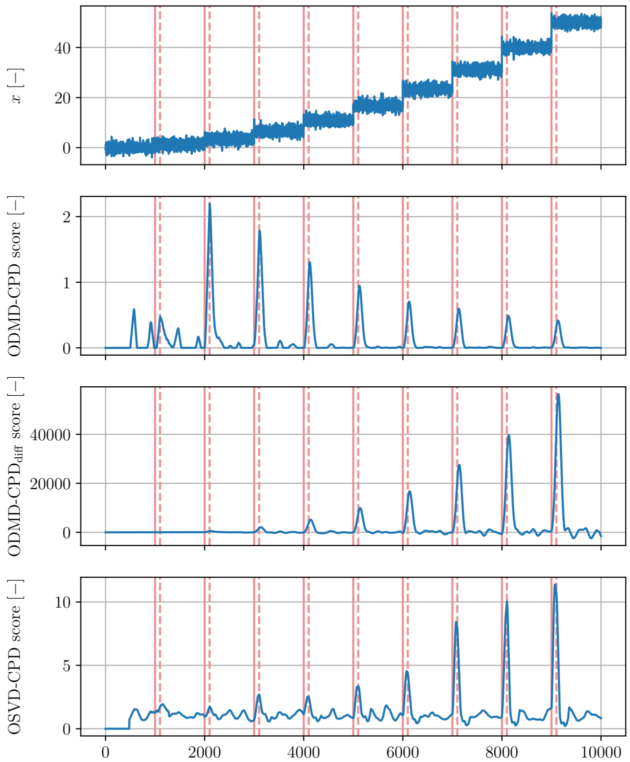

Fig. 1. Detection of synthetic step changes. The signal (top) contains nine steps of increasing magnitude. The toDMDc score (middle two plots) peaks within the deterministic delay bound $c = 100$ samples after each true change-point (measured 102 ± 3). Reference SVD-based method (bottom) shows higher noise, potentially masking the first two subtle changes.

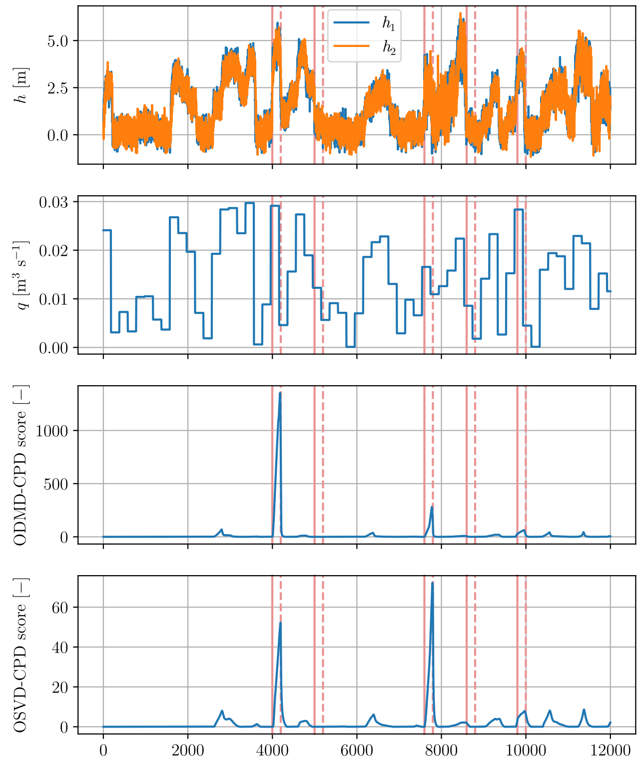

Fig. 2. CPD on nonlinear two-tank system with input delay. System states (top) and control input (second) show complex dynamics. The toDMDc score (third) detects sensor bias, control response change, and linear drift with clear peaks. Reference method (bottom) has noisier statistics.

slow linear drift detection statistics less prominent than the noise.

C. Industrial Battery Energy Storage System

Setup: We analyze a 151 kWh Li-ion battery module during a hardware fault in the cooling system on August 23, 2023. The dataset contains six temperature sensors sampled at 30-second intervals. The fault causes persistent temperature elevation. Since the system follows daily solar generation patterns, we set $d = 2880$ samples (24 hours) to capture diurnal cycles. Windows are set to $a = c = 240$ samples (2 hours, matching the battery's maximum charge/discharge rate of 1C). Time-delay embedding uses $h = 240$ samples.

Results: Figure 3 shows that toDMDc detects the transition to faulty operation with high confidence ($S_{\max} \approx 2.0$ at confirmed fault time). Several smaller peaks ($S \approx 1.0$) appear 2000 samples *before* the confirmed fault time, suggesting early onset of the change — an unmodeled slow dynamics. These early warnings align with anomalies detected in [15], providing cross-validation.

V. Discussion

A. Comparison with State-of-the-Art

Unlike statistical methods (CUSUM, GLR) [16] that assume a known distribution and fail under temporal correlation, toDMDc makes no such assumptions and explicitly models dynamics. Deep-learning approaches like TIRE [13] achieve high accuracy but lack interpretability and require labeled data; toDMDc is unsupervised, with interpretable reconstruction-error statistics and DMD modes that enable root cause analysis. Autonomous subspace methods [3], [5] ignore control inputs, whereas our DMDc formulation disambiguates actuation from system change. Kernel methods [17], [18] scale as $\mathcal{O}(n^2)$ versus our $\mathcal{O}(mr^2)$ and are less interpretable.

B. Applicability of toDMDc

The framework is particularly well-suited for:

(1) Controlled systems: Industrial processes where control actions significantly influence behavior. The DMDc formulation is essential here—standard DMD would misinterpret control as system changes.

(2) Multi-scale dynamics: Systems with multiple time constants (fast actuation, slow drift). Time-delay embedding h captures transients while windowed updates track long-term evolution.

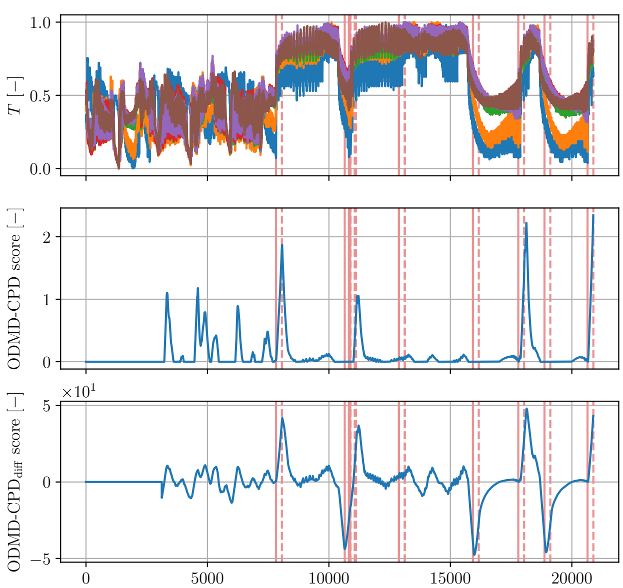

Fig. 3. Faulty HVAC operation in BESS. Temperature profiles (top) show gradual elevation after the fault. The toDMDc score (middle) detects the transition with early precursor signals. Error difference formulation (bottom) shows bidirectional detection (positive peaks for transitions to fault, negative for returns to normal).

(3) Interpretability requirements: Safety-critical applications where operators need to understand *why* an alarm triggered. DMD modes reveal which frequencies changed, while S_k quantifies relative severity.

(4) Limited labeled data: Existing installations where historical change-point annotations are unavailable. The unsupervised nature and online training enable immediate deployment.

C. Practical Deployment Considerations

Real-world deployment revealed several insights:

Threshold selection: While we used a fixed η per experiment, adaptive thresholding based on the historical S_k distribution can reduce false alarms. A percentile-based approach (e.g., $\eta = 95$th percentile of S_k during normal operation) adapts to system-specific noise levels, although it may introduce a systematic bias when the calibration window itself is contaminated by undetected changes.

Multi-resolution detection: Using multiple test windows $c_1 < c_2 < \cdots$ enables detection at different time scales—fast c_1 for abrupt changes, slow c_n for gradual drift. This addresses the "resolution-delay tradeoff" inherent in CPD.

Feature attribution: The reconstruction error can be decomposed by DMD mode: $E = \sum_i E_i$ where E_i is the contribution of mode i. This identifies *which* dynamics changed.

VI. Conclusion

We presented an interpretable change-point detection (CPD) framework based on truncated online Dynamic Mode Decomposition with control (toDMDc). By combining user-specified rank truncation with online system identification, the framework maintains numerical stability while adapting to time-varying dynamics in real-time. The detection statistic, derived from reconstruction error ratios, provides interpretable measures of change severity with deterministically bounded detection delays, while toDMDc maintains an up-to-date reduced-order linear model of the system dynamics that underpins the reconstruction-error statistic.

Validation on three case studies—synthetic benchmarks, nonlinear controlled systems with delays, and industrial battery storage—demonstrates accurate detection across diverse change types (abrupt, gradual, drift) with low noise sensitivity. The method's computational efficiency ($\mathcal{O}(mr^2)$ per update) and interpretability make it suitable for deployment in safety-critical industrial applications.

References

[1] S. Joe Qin, "Statistical process monitoring: basics and beyond," *Journal of Chemometrics*, vol. 17, no. 8-9, pp. 480–502, 2003. [Online]. Available: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.800

[2] L. Korycki and B. Krawczyk, "Concept drift detection from multi-class imbalanced data streams," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, April 2021, pp. 1068–1079.

[3] Y. Kawahara, T. Yairi, and K. Machida, "Change-point detection in time-series data based on subspace identification," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, Oct 2007, pp. 559–564.

[4] M. Döhler and L. Mevel, "Subspace-based fault detection robust to changes in the noise covariances," *Automatica*, vol. 49, no. 9, pp. 2734–2743, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109813003336

[5] V. Moskvina and A. Zhigljavsky, "An algorithm based on singular spectrum analysis for change-point detection," *Communications in Statistics - Simulation and Computation*, vol. 32, no. 2, pp. 319–352, 2003. [Online]. Available: https://doi.org/10.1081/SAC-120017494

[6] S. Hosur and D. Duan, "Subspace-driven output-only based change-point detection in power systems," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1068–1076, March 2019.

[7] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[8] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016. [Online]. Available: https://doi.org/10.1137/15M1013857

[9] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "Online dynamic mode decomposition for time-varying systems," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 3, pp. 1586–1609, 2019. [Online]. Available: https://doi.org/10.1137/18M1192329

[10] Y. Zhang, "An answer to an open question in the incremental svd," 2022.

[11] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed. Cambridge University Press, 2022.

[12] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20–30, 2006, special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0024379505003812

[13] T. De Ryck, M. De Vos, and A. Bertrand, "Change point detection in time series data using autoencoders with a time-invariant representation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3513–3524, 2021.

[14] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, Aug 2014.

[15] M. Wadinger and M. Kvasnica, “Adaptable and interpretable framework for anomaly detection in scada-based industrial systems,” *Expert Systems with Applications*, vol. 246, p. 123200, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424000654

[16] J. Ye, Y. Xu, and Q. Wang, “Subspace change point detection under spiked wigner model,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 1995–2010, 2023.

[17] A. Ferrari, C. Richard, A. Bourrier, and I. Bouchikhi, “Online change-point detection with kernels,” *Pattern Recognition*, vol. 133, p. 109022, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322005027

[18] L. Chu and H. Chen, “Sequential change-point detection for high-dimensional and non-euclidean data,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 4498–4511, 2022.