

A Zonotope-based Big Data-driven Predictive Control Approach for Nonlinear Processes[★]

Shuangyu Han^{*} Yitao Yan^{*} Jie Bao^{*} Biao Huang^{**}

^{*} *School of Chemical Engineering, University of New South Wales, Sydney, NSW, 2052, Australia. (e-mail: shuangyu.han@student.unsw.edu.au; y.yan@unsw.edu.au; j.bao@unsw.edu.au).*

^{**} *Department of Chemical and Materials Engineering, University of Alberta, AB T6G2G6, Edmonton, Canada. (email: biao.huang@ualberta.ca).*

Abstract: A zonotope-based big data-driven predictive control (BDPC) approach is developed to partition the nonlinear process behaviour (represented by an input-output trajectory set) into multiple linear sub-behaviours using a two-step hierarchical clustering: Euclidean distance-based clustering and linear subspace distance-based clustering. By approximating every linear sub-behaviour as a zonotope, a data-driven interpolation is developed based on the convex combination of zonotopes. During online control, a BDPC controller is designed by determining an interpolated zonotope where its centre trajectory is closest to the online trajectory and computing the control action subject to an optimisation problem. The proposed BDPC approach is illustrated using a case study on controlling an aluminium smelting process.

Keywords: Behavioural systems theory, Big data, Clustering, Data-driven predictive control, Zonotope.

1. INTRODUCTION

The complexity of nonlinear processes leads to challenges in building an accurate dynamical process model by first principles and system identification methods (Hou and Wang, 2013). Due to the vast amount of operational data generated by industrial processes, data-driven control has attracted recent research attention, which aims to design controllers directly based on system data. The existing data-driven control approaches can be categorised based on data type: offline, online, and hybrid. For instance, recurrent neural networks are trained using offline data to capture process dynamics and are used for predictive control (Wu et al., 2019a,b). Iterative learning control uses hybrid data to learn a controller, where the tracking error approaches zero as the number of iterations increases (Chen and Wen, 1999; Moore, 2012).

However, only a few data-driven control approaches focus on developing control strategies using a vast amount of data trajectories. The big data approximating control approach uses a big database to transform the control problem into trajectory pattern matching (Stanley, 2018). However, it is not realistic to store all possible trajectories. A random forests-based approach learns a nonlinear process with a collection of local linear models in Wang et al. (2019). However, the model bias is induced by linear regression.

In this article, we propose a big data-driven predictive

control (BDPC) approach in the behavioural systems framework, which facilitates data-driven control through describing a dynamical system by a collection of trajectories (Willems, 1991; Willems and Polderman, 1997). One of the research lines relies on the fundamental lemma, which states that all possible trajectories of a controllable linear time-invariant (LTI) system can be parameterised by a single trajectory (under persistency of excitation) (Willems et al., 2005). Some data-driven predictive control approaches restricted to LTI systems have been proposed, e.g., conceptual framework (Markovskiy and Rapisarda, 2008), DeePC (Coulson et al., 2019), robust control (Huang et al., 2023) and distributed control (Yan et al., 2024). The linear tracking data-driven predictive control can deal with mild nonlinear processes in Berberich et al. (2022). A BDPC control approach has been proposed for nonlinear processes using clustering in Han et al. (2024a,b). The multi-view clustering of trajectories is formulated as a weighting-dependent optimisation problem, in which it is difficult to analyse the effect of individual views on clusters. Furthermore, the approach selects clusters during online predictive control, which cannot perform data-driven interpolation.

The basic idea of this article is to use a divide-and-conquer approach: partition the nonlinear process behaviour into linear sub-behaviours represented by trajectory clusters. Using the hierarchical clustering of input-output trajectories (Hastie et al., 2009), we approximate every trajectory cluster (linear sub-behaviour) as a zonotope. Since a zonotope is a compact representation of a convex set and is

[★] This work was supported by the Australian Research Council under Grant DP240100300. Corresponding author: Jie Bao.

closed under linear mapping and Minkowski sum (Alanwar et al., 2023; Kühn, 1998; Scott et al., 2016), we use it for data-driven interpolation based on the convex combination. Finally, an online BDPC controller is designed by determining an interpolated zonotope based on the online trajectory and computing control actions subject to an optimisation problem.

We present the preliminaries of the discrete-time behaviour framework and zonotope in Section 2. The proposed approach for hierarchical clustering of trajectories, zonotope-based approximation and interpolation, and online BDPC are presented in Section 3. An illustrative example is shown in Section 4. Section 5 gives the conclusion.

2. PRELIMINARIES

2.1 Notation

Let A^\dagger represent the Moore–Penrose pseudoinverse of a real matrix A . Given a vector x , the weighted 2-norm $\sqrt{x^\top P x}$ and infinity norm are represented as $\|x\|_P$ and $\|x\|_\infty$, respectively. The operators $\text{diag}(d_1, \dots, d_n)$ and $\text{col}(d_1, d_2)$ represent a diagonal matrix with entries d_1, \dots, d_n and a vector stacking d_1 over d_2 vertically, respectively. The trajectory $\tilde{w}|_{[1,L]}$ restricts the variable w to the time interval $[1, L]$, i.e., $\tilde{w}|_{[1,L]} := (w(1), w(2), \dots, w(L))$ and $w(k) \in \mathbb{R}^w, \forall k \in [1, L]$.

2.2 Discrete-time Behavioural Systems Framework

The behaviour approach describes a dynamical system by a collection of manifest variable trajectories (denoted as \tilde{w}) and defines it by a triple $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, where $\mathbb{T} \subset \mathbb{N}$ denotes the time axis, $\mathbb{W} \subset \mathbb{R}^w$ denotes the w -dimensional signal space of manifest variables and $\mathfrak{B} \subset \mathbb{W}^{\mathbb{T}}$ denotes system behaviour (Willems, 1991; Willems and Polderman, 1997). A discrete-time system behaviour restricting the trajectory length to L steps is denoted as $\mathfrak{B}|_L := \{\tilde{w}|_{[1,L]} \mid w \in \mathfrak{B}\}$. The discrete-time system behaviour is LTI if it is a shift-invariant subspace. The behaviour \mathfrak{B} is controllable if any past trajectory $\tilde{w}_p \in \mathfrak{B}$ can be driven to any future trajectory $\tilde{w}_f \in \mathfrak{B}$ by a finite length control trajectory (Markovsky and Dörfler, 2021). We use \mathfrak{L}_c^w to represent the set of controllable discrete-time LTI systems with dimension w .

Consider a trajectory $\tilde{w} \in \mathfrak{B}|_T$ and define a Hankel matrix of depth L as

$$\mathfrak{H}_L(\tilde{w}) = \begin{bmatrix} w(1) & w(2) & \dots & w(T-L+1) \\ w(2) & w(3) & \dots & w(T-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ w(L) & w(L+1) & \dots & w(T) \end{bmatrix} \in \mathbb{R}^{Lw \times g}. \quad (1)$$

Consider a trajectory set $\tilde{\mathcal{W}} = \{\tilde{w}_1, \dots, \tilde{w}_{N_d}\} \subset \mathfrak{B}|_T$, a mosaic-Hankel matrix $\mathfrak{H}_L(\tilde{\mathcal{W}}) \in \mathbb{R}^{Lw \times N_d g}$ is defined as

$$\mathfrak{H}_L(\tilde{\mathcal{W}}) = [\mathfrak{H}_L(\tilde{w}_1) \dots \mathfrak{H}_L(\tilde{w}_{N_d})], \quad (2)$$

where N_d denotes the number of trajectories. A trajectory set $\tilde{\mathcal{W}}$ is collectively persistently exciting of order L if $\mathfrak{H}_L(\tilde{\mathcal{W}})$ is of full row rank. The extended fundamental lemma states that all possible trajectories of the linear sub-behaviour $\mathfrak{B} \in \mathfrak{L}_c^w$ can be parameterised by the image of a mosaic-Hankel matrix.

Lemma 1 (Extended Fundamental Lemma (van Waarde et al., 2020)). *Let $\mathfrak{B} \in \mathfrak{L}_c^w$. Consider an input-output trajectory set $\tilde{\mathcal{W}} := (\tilde{\mathcal{U}}, \tilde{\mathcal{Y}}) \subset \mathfrak{B}|_T$. If $\tilde{\mathcal{U}}$ is collectively persistently exciting of order $L + \mathbf{n}(\mathfrak{B})$, then for all $\tilde{w}' \in \mathfrak{B}|_L$, there exists $g \in \mathbb{R}^{N_d g}$ such that*

$$\tilde{w}' = \mathfrak{H}_L(\tilde{\mathcal{W}})g, \quad (3)$$

where $\mathbf{n}(\mathfrak{B})$ denotes the order of \mathfrak{B} (i.e., the smallest dimension of state representations). The vector g in (3) can be considered as selecting a trajectory from the linear sub-behaviour. To deal with nonlinearity, the low-rank approximation is typically performed on the Hankel matrix/mosaic-Hankel matrix to approximate the true linear sub-behaviour (Markovsky, 2008). For instance, the relatively small singular values can be set to zero.

2.3 Zonotope

In this article, we use the zonotope as a convex set representation to approximate trajectory clusters.

Definition 1 (Zonotope (Kühn, 1998; Scott et al., 2016)). *Given a centre $c \in \mathbb{R}^{n_x}$ and a generator matrix $G \in \mathbb{R}^{n_x \times n_g}$, \mathcal{Z} is a zonotope defined as*

$$\mathcal{Z} = \left\{ x \in \mathbb{R}^{n_x} \mid x = c + Gg, \|g\|_\infty \leq 1 \right\}. \quad (4)$$

We use $\mathcal{Z} = \{c, G\}$ to represent a zonotope. Given a scalar α , then $\alpha\mathcal{Z} = \{\alpha c, \alpha G\}$. The Minkowski sum of two zonotopes $\mathcal{Z}_1 = \{c_1, G_1\}$ and $\mathcal{Z}_2 = \{c_2, G_2\}$ is $\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{x_1 + x_2 \mid x_1 \in \mathcal{Z}_1, x_2 \in \mathcal{Z}_2\}$, which can be computed as (Alanwar et al., 2023)

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{c_1 + c_2, [G_1, G_2]\}. \quad (5)$$

3. PROPOSED APPROACH

The section starts with clustering input-output trajectories to obtain trajectory clusters (representing linear sub-behaviours). By approximating trajectory clusters as zonotopes, a data-driven interpolation method is developed. Finally, the online BDPC is formulated.

3.1 Hierarchical Clustering of Trajectories

This section aims to use hierarchical clustering to cluster input-output trajectories based on two steps: Euclidean distance of trajectories and linear subspace distance. Given two T -length trajectories $\tilde{w}_{[1,T]}^1$ and $\tilde{w}_{[1,T]}^2$, the Euclidean distance of two trajectories is defined as

$$d_E(\tilde{w}^1, \tilde{w}^2) = \left\| \tilde{w}_{[1,T]}^1 - \tilde{w}_{[1,T]}^2 \right\|. \quad (6)$$

The linear subspace distance is defined in terms of principal angles. Given two linear subspaces $\mathfrak{H}_L(\tilde{w}^1)$ and $\mathfrak{H}_L(\tilde{w}^2)$, the linear subspace distance is defined as (Ye and Lim, 2016)

$$d_S(\mathfrak{H}_L(\tilde{w}^1), \mathfrak{H}_L(\tilde{w}^2)) = \left(\frac{|n_1 - n_2| \pi^2}{4} + \sum_{i=1}^{\min(n_1, n_2)} \theta_i^2 \right)^{\frac{1}{2}}, \quad (7)$$

where $\mathfrak{H}_L(\tilde{w}^1) \in Gr(n_1, Lw)$ and $\mathfrak{H}_L(\tilde{w}^2) \in Gr(n_2, Lw)$. We use $Gr(n_1, Lw)$ and $Gr(n_2, Lw)$ to denote the Grassmannian of n_1 -dimensional subspaces and n_2 -dimensional

subspaces in \mathbb{R}^{Lw} , respectively. We denote θ_i as the i -th principal angle between $\mathfrak{H}_L(\tilde{w}^1)$ and $\mathfrak{H}_L(\tilde{w}^2)$, which can be efficiently determined through singular value decomposition (Björck and Golub, 1973).

The hierarchical clustering algorithm is presented in Algorithm 1, which clusters trajectories based on Euclidean distance in (6) firstly and linear subspace distance in (7) subsequently. We obtain N_c trajectory clusters (representing N_c linear sub-behaviours) after performing hierarchical clustering on N_d input-output trajectories of length T . The centre trajectory of a cluster can be obtained by taking an average of all trajectories within the cluster.

Algorithm 1 Hierarchical clustering of trajectories

- 1: **Input:** N_d input-output trajectories, thresholds ϵ_E and ϵ_S .
 - 2: Initialise a cluster containing N_d input-output trajectories and the number of clusters $N_c = 1$.
 - 3: **Clustering based on Euclidean distance:**
 - 4: Find two trajectories \tilde{w}_i^1 and \tilde{w}_i^2 that have the maximum distance d_i^{max} in (6) in the i -th cluster, $\forall i \in [1, N_c]$.
 - 5: **if** there exists a cluster C_\star such that $d_\star^{max} \geq \epsilon_E$ **then**
 - 6: Initialise two new clusters: $C_\star^1 = \{\tilde{w}_\star^1\}, C_\star^2 = \{\tilde{w}_\star^2\}$.
 - 7: **if** $d_E(\tilde{w}, \tilde{w}_\star^1) \leq d_E(\tilde{w}, \tilde{w}_\star^2), \forall \tilde{w} \in C_\star$ **then**
 - 8: $\tilde{w} \in C_\star^1$.
 - 9: **else**
 - 10: $\tilde{w} \in C_\star^2$.
 - 11: **end if**
 - 12: $N_c \leftarrow N_c + 1$. Go to Step 4.
 - 13: **else**
 - 14: go to Step 16.
 - 15: **end if**
 - 16: **Clustering based on linear subspace distance:**
 - 17: Find two trajectories \tilde{w}_i^1 and \tilde{w}_i^2 that have the maximum distance d_i^{max} in (7) in the i -th cluster, $\forall i \in [1, N_c]$.
 - 18: **if** there exists a cluster C_\star such that $d_\star^{max} \geq \epsilon_S$ **then**
 - 19: Initialise two new clusters: $C_\star^1 = \{\tilde{w}_\star^1\}, C_\star^2 = \{\tilde{w}_\star^2\}$.
 - 20: **if** $d_S(\tilde{w}, \tilde{w}_\star^1) \leq d_S(\tilde{w}, \tilde{w}_\star^2), \forall \tilde{w} \in C_\star$ **then**
 - 21: $\tilde{w} \in C_\star^1$.
 - 22: **else**
 - 23: $\tilde{w} \in C_\star^2$.
 - 24: **end if**
 - 25: $N_c \leftarrow N_c + 1$. Go to Step 17.
 - 26: **else**
 - 27: go to Step 29.
 - 28: **end if**
 - 29: **Output:** N_c trajectory clusters.
-

3.2 Zonotope-based approximation and interpolation

Every trajectory cluster can be approximated as a zonotope since (i) each cluster represents a linear subspace; (ii) within every cluster, all trajectories are close to each other in the Euclidean space. By deviating all trajectories of the i -th cluster from the i -th cluster centre trajectory (denoted as $\Delta\tilde{W}_i$) and performing appropriate low-rank approximation, we obtain a mosaic-Hankel matrix $\mathfrak{H}_L(\Delta\tilde{W}_i)$ for the i -th cluster. Then, the i -th trajectory cluster can be approximated as a zonotope, that is,

$$\mathcal{Z}_i = \{\tilde{w} | \tilde{w} = \tilde{c}_i + \mathfrak{H}_L(\Delta\tilde{W}_i)g, \|g\|_\infty \leq 1\}, \quad (8)$$

where \tilde{c}_i denotes the L -length centre trajectory of the i -th cluster. One possible solution is to take an average of all possible L -length segments of the T -length trajectory. We use $\mathcal{Z}_i = \{\tilde{c}_i, \mathfrak{H}_L(\Delta\tilde{W}_i)\}$ to represent the i -th cluster. Since finite trajectory clusters are obtained by hierarchical clustering, we use the convex combination of zonotopes for interpolation. Supposing that there are N_c zonotopes and considering them as the vertices of a convex polytope, the interpolated zonotope can be computed as

$$\tilde{\mathcal{Z}} = (\alpha_1 \mathcal{Z}_1) \oplus (\alpha_2 \mathcal{Z}_2) \oplus \cdots \oplus (\alpha_{N_c} \mathcal{Z}_{N_c}), \quad (9a)$$

$$\sum_{i=1}^{N_c} \alpha_i = 1, \alpha_i \geq 0, \quad (9b)$$

where $\tilde{\mathcal{Z}} = \{\tilde{c}, \tilde{\mathfrak{H}}_L(\Delta\tilde{W})\}$. Eq. (9) provides a data-driven interpolation that determines the interpolated zonotope by a linear combination of N_c zonotopes obtained from clustering.

3.3 Online Big Data-driven Predictive Control

The online BDPC has two steps within each receding horizon. Firstly, it determines the interpolated zonotope $\tilde{\mathcal{Z}} = \{\tilde{c}, \tilde{\mathfrak{H}}_L(\Delta\tilde{W})\}$ based on the online trajectory \tilde{w} . It aims to find an interpolated zonotope, where the centre trajectory is closest to the online trajectory \tilde{w} . It can be formulated as

$$\min_{\alpha_1, \alpha_2, \dots, \alpha_{N_c}} \|\tilde{w} - \tilde{c}\| \quad (10a)$$

$$\text{s.t. (9)}. \quad (10b)$$

The second step is to predict the future step of manifest variables by a mosaic-Hankel matrix and implement the control actions into the process. We predict one future step in the current horizon and redetermine the interpolated zonotope in (10) in the next horizon. In this article, we use the following partition and permutation of the centre trajectory and Hankel matrix of an interpolated zonotope:

$$\tilde{c} \sim \begin{bmatrix} \tilde{c}_p \\ \tilde{c}_f \end{bmatrix}, \tilde{\mathfrak{H}}_L(\Delta\tilde{W}) \sim \begin{bmatrix} \Delta\tilde{U}_p \\ \Delta\tilde{U}_f \\ \Delta\tilde{Y}_p \\ \Delta\tilde{Y}_f \end{bmatrix}, \quad (11)$$

where p and f denote ‘‘past’’ and ‘‘future’’, respectively, and \sim denotes the equivalence under a coordinate permutation. We denote the deviation trajectory as $\Delta\tilde{w} = \tilde{w} - \tilde{c}$. To drive the online trajectory \tilde{w} to a reference trajectory $\tilde{w}_r = (\tilde{u}_r, \tilde{y}_r)$, an optimisation problem at the k -th step is formulated as

$$\min_{\Delta\tilde{u}(k)} \|\tilde{y}(k) - \tilde{y}_r(k)\|_{\Lambda_y} + \|\tilde{u}(k) - \tilde{u}_r(k)\|_{\Lambda_u} \quad (12a)$$

$$\text{s.t. } \Delta\tilde{y}(k) = \Delta\tilde{Y}_f \begin{bmatrix} \Delta\tilde{Y}_p \\ \Delta\tilde{U}_p \\ \Delta\tilde{U}_f \end{bmatrix}^\dagger \begin{bmatrix} \Delta\tilde{y} \\ \Delta\tilde{u} \end{bmatrix}_{[k-L+1, k-1]} \quad (12b)$$

$$\begin{bmatrix} \tilde{u}(k) \\ \tilde{y}(k) \end{bmatrix} = \begin{bmatrix} \Delta\tilde{u}(k) \\ \Delta\tilde{y}(k) \end{bmatrix} + \tilde{c}_f, \quad (12c)$$

where the weights Λ_u and Λ_y are positive definite. Eq. (12b) predicts the future deviation output using the least squares method. The optimisation can be readily solved by MATLAB toolboxes, e.g., YALMIP (Löfberg, 2004) and SeDuMi (Sturm, 1999). The online BDPC is summarised in Algorithm 2.

This article focuses on the zonotope-based trajectory cluster interpolation and online predictive control implementation. Stability conditions will be developed in future work. One possible solution is to develop incremental dissipativity and stability conditions similar to Han et al. (2024a).

Algorithm 2 Online BDPC

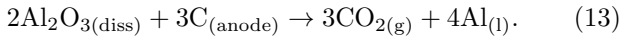
- 1: **Input:** N_c zonotopes, weights Λ_u and Λ_y .
 - 2: Find $\bar{\mathcal{Z}} = \{\bar{c}, \bar{\mathfrak{H}}_L(\Delta\bar{\mathcal{W}})\}$ in (10) at the k -th step.
 - 3: Compute the control action $\tilde{u}(k)$ in (12) and implement it into the process.
 - 4: $k \leftarrow k + 1$, go to Step 2.
-

4. ILLUSTRATIVE EXAMPLE

We illustrate the proposed approach by controlling an aluminium smelting process. We only used the model to collect input-output data trajectories in this study. The process model has not been used for controller design. The section starts with introducing an aluminium smelting process. Then the simulation results are presented.

4.1 Aluminium Smelting Process

Aluminium is produced by reacting dissolved alumina powders with carbon anodes in a reduction cell (Figure 1), where the dominant chemical reaction:



In a reduction cell, the total line current is kept constant to control the reaction rate. A parallel electrical circuit is established in the cell due to the parallel connection of the anodes. Figure 1 shows that the cell consists of 36 feeding regions (every region has an anode) and 3 alumina feeders. The cell can be partitioned into 3 zones, where each zone has an alumina feeder. In this study, we keep the constant total line current as 425 kA and anode-cathode distance ACD as 2.8 cm. Besides, we adopt the sampling period as 1 minute to omit the fast-dissolving alumina dynamics. Then the process model of alumina dissolution in the i -th feeding region ($i \in [1, 36]$):

$$C_i^u(k+1) = C_i^u(k) - k_d C_i^u(k) + \frac{g_i(k)r}{m_i}, \quad (14a)$$

$$C_i^d(k+1) = C_i^d(k) + k_d C_i^u(k) + \frac{g_i(k)(1-r) + M_i^e(k) - Fa_{\text{Al}_2\text{O}_3}(I_i^a)(k)}{m_i}, \quad (14b)$$

where C_i^u , C_i^d , I_i^a , M_i^e , m_i and g_i denote the undissolved alumina concentration, dissolved alumina concentration, anode current, dissolved alumina exchange induced by the mass flow, mass of bath and amount of fed alumina in the i -th region, respectively. We use r and K_d to denote two constants: weight ratio between the fast and slow-dissolving alumina, and dissolution rate, respectively. The alumina consumption rate function $Fa_{\text{Al}_2\text{O}_3}(\cdot)$ is defined as

$$Fa_{\text{Al}_2\text{O}_3}(I) = \frac{I \times M_{\text{Al}_2\text{O}_3} \times \eta}{F \times z}, \quad (14c)$$

where F , z , η , $M_{\text{Al}_2\text{O}_3}$ denote the Faraday constant, number of electrons transferred, current efficiency and molar

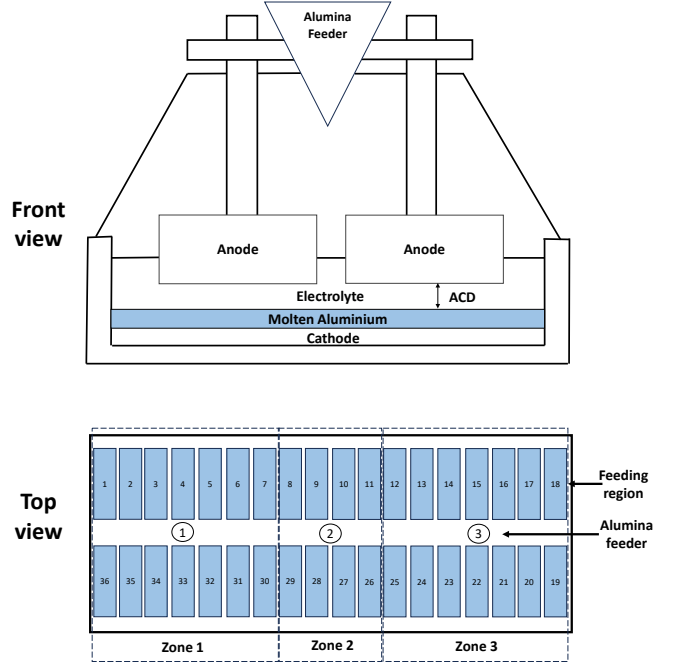


Fig. 1. The reduction cell configuration.

mass of alumina, respectively.

The feed rate control of the aluminium smelting process aims to regulate the average alumina concentration around a setpoint. The manipulated variables are the feed rate of three zones, which are typically controlled by a logic control strategy of four feeding windows in Figure 2 in the industry. The base-feed can be computed theoretically by the reaction rate. The average alumina concentration can be roughly controlled by regulating the cell voltage around a setpoint by manipulating the feed rate of three zones. The output variables of the process are the anode current of three zones and a common cell voltage, which can be computed as

$$C_{avg}^d(k) = \frac{1}{36} \sum_{i=1}^{36} C_i^d(k), \quad (15a)$$

$$I^l = \sum_{i=1}^{36} I_i^a(k), \quad (15b)$$

$$V^c(k) = h(I^l, ACD, C_{avg}^d(k), \theta), \quad (15c)$$

where C_{avg}^d , I^l and V^c denote the averaged alumina concentration, total line current and cell voltage, respectively. $h(\cdot)$ and θ denote a nonlinear function and cell design and operation parameters, respectively. See Grjotheim and Welch (1988) and Yao et al. (2017) for details.

4.2 Simulation results

In this study, we used 100 input-output data trajectories. The proposed BDPC approach was used to control the anode currents of three zones and cell voltage by manipulating the feed rate of three alumina feeders. We assumed that the alumina fed by each feeder is uniformly spread across all the regions within the zone. The collected data trajectories were 100 minutes and had different feeding window

5. CONCLUSION

We developed a BDPC approach for nonlinear processes using hierarchical clustering of input-output trajectories and zonotope-based approximation of trajectory clusters. A data-driven interpolation method has been developed based on the linear combinations of zonotopes. The online BDPC controller determines an interpolated zonotope, where the centre trajectory is closest to the online trajectory. The online control action is computed using the interpolated Hankel matrix and centre trajectory subject to an optimisation problem. The example of controlling an aluminium smelting process has demonstrated that the anode currents of three zones and cell voltage have converged to the setpoint using the proposed BDPC approach.

REFERENCES

- Alanwar, A., Koch, A., Allgöwer, F., and Johansson, K.H. (2023). Data-driven reachability analysis from noisy data. *IEEE Transactions on Automatic Control*, 68(5), 3054–3069.
- Berberich, J., Köhler, J., Müller, M.A., and Allgöwer, F. (2022). Linear tracking MPC for nonlinear systems—Part II: The data-driven case. *IEEE Transactions on Automatic Control*, 67(9), 4406–4421.
- Björck, Å. and Golub, G.H. (1973). Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123), 579–594.
- Chen, Y. and Wen, C. (1999). *Iterative learning control: convergence, robustness and applications*. Springer.
- Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, 307–312.
- Grjothheim, K. and Welch, B.J. (1988). *Aluminium Smelter Technology—a Pure and Applied Approach*. Aluminium-Verlag, P. O. Box 1207, Königsallee 30, D 4000 Dusseldorf 1, FRG, 1988.
- Han, S., Yan, Y., Bao, J., and Huang, B. (2024a). A big data-driven predictive control approach for nonlinear processes using behaviour clusters. *Journal of Process Control*, 140, 103252.
- Han, S., Yan, Y., Bao, J., and Huang, B. (2024b). Big data-driven predictive control using multi-view clustering. In *2024 American Control Conference (ACC)*, 5100–5105.
- Hastie, T., Tibshirani, R., Friedman, J.H., and Friedman, J.H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*, volume 2. Springer.
- Hou, Z.S. and Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235, 3–35.
- Huang, L., Zhen, J., Lygeros, J., and Dörfler, F. (2023). Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Transactions on Automatic Control*, 68(5), 3163–3170.
- Kühn, W. (1998). Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61, 47–67.
- Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No. 04CH37508)*, 284–289.

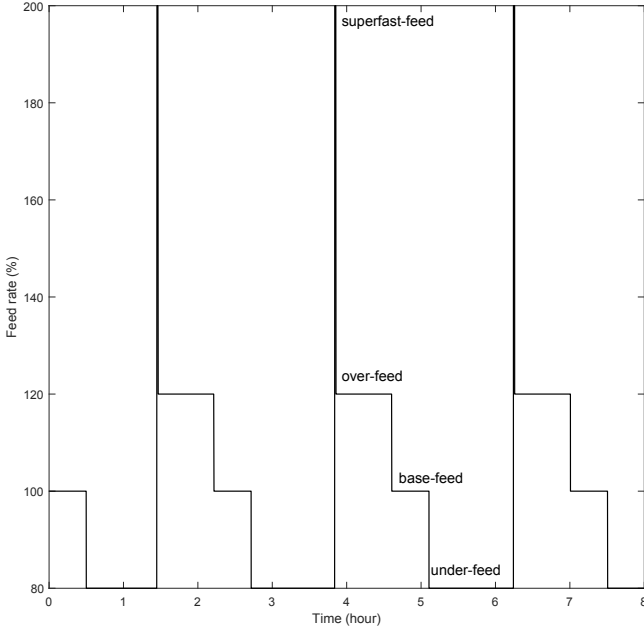


Fig. 2. Four alumina feed rate windows.

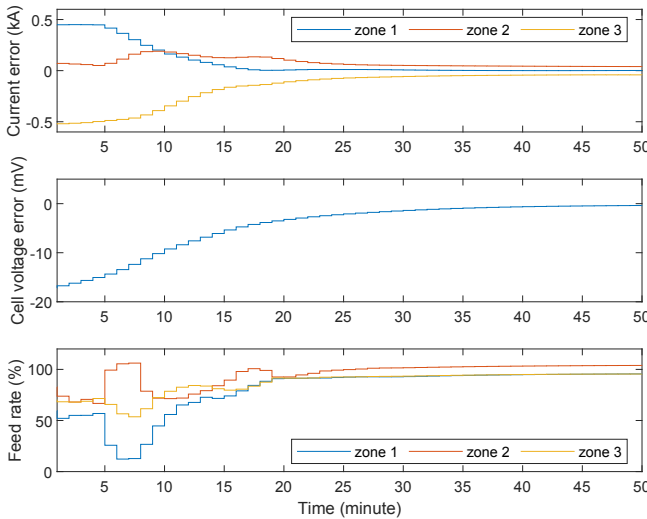


Fig. 3. Control performance of the proposed BDPC approach.

lengths and initial conditions of averaged alumina concentration. All manifest variables were normalised globally in the trajectory set. By setting $L = 7$, $\epsilon_E = 10$ and $\epsilon_S = 2$, we obtained 7 clusters from hierarchical clustering in Algorithm 1. We set the reference points as

$$y_r = \text{col}(I_{r_1}^z, I_{r_2}^z, I_{r_3}^z, V_r^c) = \text{col}(167.2, 94.3, 163.5, 4.02),$$

$$u_r = \text{col}(95.88, 108.15, 93.76),$$

where $I_{r_i}^z$ denotes the i -th zone reference anode current. The units of $I_{r_i}^z$, V_r^c and u_r are kiloampere (kA), volt (V) and %, respectively. By setting the weights $\Lambda_u = \text{diag}(1,1,1)$ and $\Lambda_y = \text{diag}(45,45,45,220)$, the tracking error of anode current of three zones and cell voltage, and the feed rate of three zones are shown in Figure 3. It has been demonstrated that the anode currents of three zones and cell voltage have converged to the setpoint using the proposed BDPC approach.

- Markovsky, I. (2008). Structured low-rank approximation and its applications. *Automatica*, 44(4), 891–909.
- Markovsky, I. and Dörfler, F. (2021). Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 52, 42–64.
- Markovsky, I. and Rapisarda, P. (2008). Data-driven simulation and control. *International Journal of Control*, 81(12), 1946–1959.
- Moore, K.L. (2012). *Iterative learning control for deterministic systems*. Springer Science & Business Media.
- Scott, J.K., Raimondo, D.M., Marseglia, G.R., and Braatz, R.D. (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69, 126–136.
- Stanley, G. (2018). Big Data Approximating Control (BDAC)—A new model-free estimation and control paradigm based on pattern matching and approximation. *Journal of Process Control*, 67, 141–159.
- Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4), 625–653.
- van Waarde, H.J., De Persis, C., Camlibel, M.K., and Tesi, P. (2020). Willems’ fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3), 602–607.
- Wang, R., Bao, J., and Yao, Y. (2019). A data-centric predictive control approach for nonlinear chemical processes. *Chemical Engineering Research and Design*, 142, 154–164.
- Willems, J.C. (1991). Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36(3), 259–294.
- Willems, J.C. and Polderman, J.W. (1997). *Introduction to mathematical systems theory: A behavioral approach*, volume 26. Springer Science & Business Media.
- Willems, J.C., Rapisarda, P., Markovsky, I., and De Moor, B.L. (2005). A note on persistency of excitation. *Systems & Control Letters*, 54(4), 325–329.
- Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019a). Machine learning-based predictive control of nonlinear processes. Part I: Theory. *AIChE Journal*, 65(11), e16729.
- Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019b). Machine-learning-based predictive control of nonlinear processes. Part II: Computational implementation. *AIChE Journal*, 65(11), e16734.
- Yan, Y., Bao, J., and Huang, B. (2024). Distributed data-driven predictive control via dissipative behavior synthesis. *IEEE Transactions on Automatic Control*, 69(5), 2899–2914.
- Yao, Y., Cheung, C.Y., Bao, J., Skyllas-Kazacos, M., Welch, B.J., and Akhmetov, S. (2017). Estimation of spatial alumina concentration in an aluminum reduction cell using a multilevel state observer. *AIChE Journal*, 63(7), 2806–2818.
- Ye, K. and Lim, L.H. (2016). Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3), 1176–1197.