SCIENCE CHINA
Information Sciences

• LETTER •

July 2026, Vol. 69, Iss. 7, 179201:1–179201:2
https://doi.org/10.1007/s11432-025-4749-4

Deep Koopman modeling and predictive tracking control for metro train longitudinal dynamics

Wenbo LIAN[2], Weiqi BAI[1*] & Hairong DONG[3]

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100084, China
[2]School of Automation and Intelligence, Beijing Jiaotong University, Beijing 100044, China
[3]College of Electronic and Information Engineering, Tongji University, Shanghai 200092, China

Received 2 September 2025/Revised 13 November 2025/Accepted 23 December 2025/Published online 2 February 2026

Citation Lian W B, Bai W Q, Dong H R. Deep Koopman modeling and predictive tracking control for metro train longitudinal dynamics. Sci China Inf Sci, 2026, 69(7): 179201, https://doi.org/10.1007/s11432-025-4749-4

As a vital component of urban public transportation systems, metro trains play a crucial role in alleviating traffic congestion and supporting reliable daily commuting. With the continuous growth of passenger travel demand, automatic train operation (ATO) systems are developed to meet the requirements for enhanced operational efficiency and improved reliability [1, 2]. Accurate modeling of train dynamics is fundamental to ensuring that ATO systems maintain superior operational performance. Traditional physics-based models, while interpretable, rely on simplified assumptions that fail to capture the nonlinear and time-varying characteristics of real-world train dynamics [3]. Conversely, data-driven deep learning models can represent complex behaviors but often lack interpretability [4], and are difficult to integrate with linear control frameworks essential for safety-critical applications. Koopman operator theory offers a principled solution by lifting nonlinear dynamics into a linear latent space, enabling the application of advanced control techniques with enhanced transparency and theoretical rigor [5]. However, the application of Koopman-based modeling to train operation remains largely unexplored.

To address these challenges, this work proposes a deep Koopman operator (DKO) framework for data-driven modeling and predictive control of metro train longitudinal dynamics. The framework employs a deep neural network as a lifting function to embed nonlinear train dynamics into a linear latent space. A multi-step prediction loss function is designed to explicitly account for actuator lag and long-term temporal dependencies. The learned linear representation is subsequently incorporated into a model predictive control (MPC) framework to achieve real-time speed tracking while ensuring compliance with safety constraints under varying operating conditions.

Consider the following discrete train control model:

$$x_{k+1} = f_d(x_k, u_k), \tag{1}$$

where $x_k \in \mathbb{X} \subset \mathbb{R}$ and $u_k \in \mathbb{U} \subset \mathbb{R}$. Let $\mathbf{u}$ be a control sequence vector, defined as $\mathbf{u} \triangleq \big(u(i)\big)_{i=0}^{\infty}$, and $\mathbf{u} \in \mathcal{R}(\mathbb{U})$. The extended state vector $\mathbf{X}_k = [x_k, \mathbf{u}^\top]^\top$ lies in $\mathbb{R} \times \mathcal{R}(\mathbb{U})$. The dynamics of $\mathbf{X}_k$ is described by

$$\mathbf{X}_{k+1} = F_d(\mathbf{X}_k) = \begin{bmatrix} f_d(x_k, \mathbf{u}(0)) \\ S\mathbf{u} \end{bmatrix}, \tag{2}$$

where S is the left shift operator: $S\mathbf{u}(i) = \mathbf{u}(i+1)$. The Koopman operator κ is defined as

$$\kappa \hat{\mathbf{\Psi}}(\mathbf{X}) = \hat{\mathbf{\Psi}}\left(F_d(\mathbf{X})\right), \tag{3}$$

where $\hat{\mathbf{\Psi}} : \mathbb{X} \times \mathcal{R}(\mathbb{U}) \to \mathbb{R}$ is the nonlinear lifting function. To achieve a finite-dimensional approximation of κ, let

$$\hat{\mathbf{\Psi}}(x) = \Big[\mathbf{\Psi}(x)^\top \ \ u(0)\Big]^\top \tag{4}$$

with $\mathbf{\Psi}(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_p(x)]^\top$ being a vector of lifting functions. Define $z_k \triangleq \mathbf{\Psi}(x_k) \in \mathbb{R}^{p\times 1}$. It follows

$$\begin{aligned} z_{k+1} &= \mathbf{\Omega} z_k + \mathbf{\Gamma} u_k, \\ \hat{x}_k &= \mathbf{\Lambda} z_k, \end{aligned} \tag{5}$$

where $\hat{x}_k$ is the reconstructed state. The matrices $\mathbf{\Omega}$, $\mathbf{\Gamma}$, and $\mathbf{\Lambda}$ are obtained by solving

$$\begin{cases} \min\limits_{\mathbf{\Omega},\mathbf{\Gamma}} \sum\limits_{j=1}^{N} \left\| \Big[\mathbf{\Omega} \ \ \mathbf{\Gamma}\Big] \begin{bmatrix} \mathbf{\Psi}(x_j) \\ u_j \end{bmatrix} - \mathbf{\Psi}(x_{j+1}) \right\|_2^2, \\ \min\limits_{\mathbf{\Lambda}} \sum\limits_{j=1}^{N} \left\| \mathbf{\Lambda}\, \mathbf{\Psi}(x_j) - x_j \right\|_2^2. \end{cases} \tag{6}$$

The input-state datasets $\{(u_j, x_j, \tilde{x}_j)\}_{j=1}^{N}$ are vectorized as $\mathbf{X} = [x_1, x_2, \cdots, x_N]$, $\tilde{\mathbf{X}} = [\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_N]$, $\mathbf{U} = [u_1, u_2, \cdots, u_N]$, where $x_i, \tilde{x}_i, u_i \in \mathbb{R}$.

Define $\mathbf{\Psi}(\tilde{x}) \triangleq [\psi_1(\tilde{x}), \psi_2(\tilde{x}), \ldots, \psi_p(\tilde{x})]^\top$, $\mathbf{\Xi} = [\mathbf{\Psi}(\tilde{x}_1), \mathbf{\Psi}(\tilde{x}_2), \ldots, \mathbf{\Psi}(\tilde{x}_N)]$, and $\mathbf{\Theta} = [\mathbf{\Psi}(x_1), \mathbf{\Psi}(x_2), \ldots, \mathbf{\Psi}(x_N)]$.

By virtue of (4), the extended lifted state vectors for the dataset are given by

$$\tilde{\mathbf{\Theta}} = \left(\Big[\mathbf{\Psi}(x_1)^\top \ \ u(1)\Big]^\top, \ldots, \Big[\mathbf{\Psi}(x_N)^\top \ \ u(N)\Big]^\top \right).$$

* Corresponding author (email: wqbai@buaa.edu.cn)

Then, Eq. (6) can be equivalently rewritten as

$$\min_{\kappa} \|\mathbf{\Xi} - \kappa\tilde{\mathbf{\Theta}}\|_2^2 \tag{7}$$

and the optimal Koopman operator and output matrix can be calculated as $\kappa_{1:p} \triangleq [\mathbf{\Omega}, \mathbf{\Gamma}] = (\mathbf{\Xi}\tilde{\mathbf{\Theta}}^\top)(\tilde{\mathbf{\Theta}}\tilde{\mathbf{\Theta}}^\top)^\dagger, \mathbf{\Lambda} = (\mathbf{X}\mathbf{\Theta}^\top)(\mathbf{\Theta}\mathbf{\Theta}^\top)^\dagger$.

The framework learns nonlinear train dynamics from operational data and embeds them into a globally linear form via multi-step prediction for temporal consistency. The DKO-based modeling architecture is shown in Figure 1.

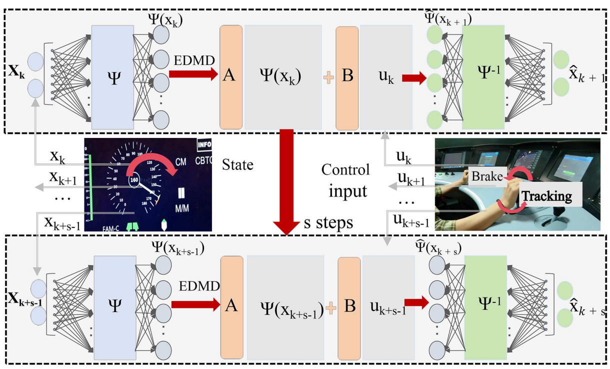

Figure 1 (Color online) The structure of the multi-step predictor based on DKO.

To learn the lifting functions, a deep encoder-decoder network $\mathbf{\Psi}(\cdot)$ and $\mathbf{\Psi}^{-1}(\cdot)$ is trained. Each layer is computed as

$$Y_*^\ell = \sigma_*^\ell(W_*^\ell Y_*^{\ell-1} + b_*^\ell), \quad * = e, d, \tag{8}$$

where $*$ denotes either the encoder (e) or the decoder (d), and W_*^ℓ and b_*^ℓ are the weights and biases. Both networks use ReLU activations for layers $\ell = 1, \ldots, H-1$, and a linear output layer without activation. The lifted and predicted states are computed as

$$\mathbf{\Psi}(x_k) = W_e^H Y_e^{H-1}, \ \hat{x}_{k+1} = W_d^H Y_d^{H-1} \tag{9}$$

and the multi-step prediction is computed as

$$\hat{\mathbf{\Psi}}(x_{k+s}) = \mathbf{\Omega}^s \mathbf{\Psi}(x_k) + \sum_{i=1}^{s} \mathbf{\Omega}^{s-i}\mathbf{\Gamma} u_{k+i-1}. \tag{10}$$

The total loss combines the reconstruction loss, the embedding consistency loss, and the multi-step prediction loss,

$$L_f = \sum_{i=1}^{3} \alpha_i L_i + \alpha_4 \left(\|\theta_E\|_1 + \|\theta_D\|_1 + \|\theta_E\|_2^2 + \|\theta_D\|_2^2\right), \tag{11}$$

$$L_1 = \frac{1}{N}\sum_{i=1}^{N}(x_k - \hat{x}_k)^2, \tag{12}$$

$$L_2 = \frac{1}{s}\sum_{i=1}^{s} L\left(\hat{\mathbf{\Psi}}(x_{k+i}), \mathbf{\Psi}(x_{k+i})\right), \tag{13}$$

$$L_3 = \frac{1}{s}\sum_{i=1}^{s} L(x_{k+i}, \hat{x}_{k+i}), \tag{14}$$

where $L(\cdot)$ is the mean squared error loss.

Then, Algorithm 1 is given to guide the training process of the deep Koopman operator training model (5), where N_{data} denotes the size of the training data, N_{batch} denotes the number of the training batch and $(\theta_E, \theta_D, \mathbf{\Omega}, \mathbf{\Gamma}, \mathbf{\Lambda})$ are optimized via stochastic gradient descent method.

Algorithm 1 Deep Koopman modeling of the train dynamics.

Input: Parameters $\theta_D, \theta_E, \xi, \eta, s, \mathbf{\Omega}, \mathbf{\Gamma}, epoch_{\max}$; hyperparameters α_i, N_{batch}, $batchsize$, p;
Output: Optimized parameters: $\theta_D, \theta_E, \mathbf{\Omega}, \mathbf{\Gamma}, \mathbf{\Lambda}$;

```
epoch ⇐ 0;
while epoch < epoch_max do
    for i = 0 to N_batch − 1 do
        Compute Θ̃ ⇐ Ψ̂(X), Ξ ⇐ Ψ(X̃);
        Calculate κ_{1:p} ≜ [Ω, Γ] = (ΞΘ̃^⊤)(Θ̃Θ̃^⊤)^†;
        for j = 0 to batchsize − 1 do
            k ⇐ i × N_batch + j;
            if k ⩾ N_data then
                break;
            end if
            Compute x̂_k and L_1 with (5) and (12);
            for l = 1 to s do
                Compute Ψ̂(x_{k+l}), L_2, L_3 with (10), (13), and (14);
            end for
        end for
        Compute L_f with (11), and update θ_E ⇐ θ_E − ξ∇_{θ_E} L_f,
        θ_D ⇐ θ_D − ξ∇_{θ_D} L_f;
    end for
    epoch ⇐ epoch + 1;
end while
Compute Λ with Λ = (XΘ^⊤)(ΘΘ^⊤)^†.
```

Based on the DKO model (5), a Koopman-based MPC (KMPC) can be formulated. The controller minimizes the tracking error and control effort, where $\Delta u_j = u_{j+1} - u_j$ and Q, R and F are the weighting matrices. Set $N_p = N_c$. The KMPC cost function is

$$\min_{\mathbf{U_k}} \sum_{i=0}^{N_p-1}\left(\|\hat{y}_{k+i|k} - y^r_{k+i|k}\|_Q^2 + \|u_{k+i|k}\|_R^2\right) + \sum_{j=0}^{N_c-2}\|\Delta u_{k+j|k}\|_F^2.$$

This can be reformulated as a quadratic program (QP): $\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^\top\mathbf{H}\mathbf{U} + f^\top\mathbf{U}$ subject to linear constraints. At each time step, the first element of the optimal sequence is applied, and the horizon shifts forward.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U24B20118, 62327806, 62303034) and Fundamental Research Funds for the Central Universities (Grant No. KG16365501).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhou M, Dong H, Song H, et al. Embodied intelligence-based perception, decision-making, and control for autonomous operations of rail transportation. IEEE Trans Intell Veh, 2025, 10: 5061–5065
2 Lin P, Huang Y, Zhang Q, et al. Distributed velocity and input constrained tracking control of high-speed train systems. IEEE Trans Syst Man Cybern Syst, 2021, 51: 7882–7888
3 Bai W, Zheng Y, Dong H. Cooperative event-triggered self-stretched feedback control for multiple trains subject to safety constraints. IEEE Trans Veh Technol, 2025, 74: 18610–18619
4 Liu Y, Yang L, Zhang S, et al. Parameter identification and platoon control for virtually coupled train set with physics-informed neural network dynamics model. Vehicle Syst Dyn, 2025, 63: 71–92
5 Lusch B, Kutz J N, Brunton S L. Deep learning for universal linear embeddings of nonlinear dynamics. Nat Commun, 2018, 9: 4950