Eksploatacja i Niezawodnosc - Maintenance and Reliability
Volume 29 (2027), Issue 2
https://ein.org.pl
The Polish Maintenance Society

Reinforcement-learning robust tracking control of discrete-time systems with diagonal scaling

Kanyang Jiang[1], Zheng Gao[1,*], Ye Zeng[1], Jun Lu[1], Chen Ci[1,*], Shengli Xie[1]

[1] School of Automation, Guangdong University of Technology, Guangzhou 510006, China
* Corresponding Author: 1112304025@mail2.gdut.edu.cn, gdutcc@gmail.com

Received: 27 March 2026
Revised: 9 August 2026
Accepted: 27 August 2026
Online: 28 September 2026

Abstract

This paper addresses a robust tracking problem for linear discrete-time systems by proposing a reinforcement learning (RL) control method based on a diagonal-scaling strategy, offering a solution tailored to the demands of enhanced reliability. To overcome a common limitation in policy-iteration-based RL design, namely, the reliance on an initially stabilizing solution, the tracking control problem is reformulated within the robust output regulation framework as a data-driven, solvable form. A convergence-rate condition is incorporated to relax the dependence on an initial stable control policy. The core contribution lies in addressing unknown system dynamics through a diagonal-scaling strategy, as opposed to conventional scalar convergence-rate scaling. The proposed method enables more flexible and precise closed-loop pole placement. The resulting data-driven controller guarantees asymptotic convergence of the tracking error to zero while maintaining robustness against dynamic uncertainties, ensuring computational efficiency and practical ease of implementation.

Keywords: *reinforcement learning, output regulation, diagonal scaling, data-driven control, policy iteration*

Article citation:
Jiang K, Gao Z, Zeng Y, Lu J, Chen C, Xie S, Reinforcement-learning robust tracking control of discrete-time systems with diagonal scaling, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2027: 29(2) http://doi.org/10.17531/ein/235726

Highlights

- Diagonal scaling is incorporated into policy iteration for robust tracking.
- A virtual control system is used to obtain an initial stabilizing policy.
- A data-driven policy iteration algorithm is developed for discrete-time systems.
- Numerical examples illustrate the tracking performance of the proposed method.

1. Introduction

In industrial and mechanical systems, control performance directly influences equipment longevity, maintenance scheduling, and system availability. Robust tracking under model uncertainties supports predictive maintenance by reducing unplanned downtime and mitigating wear from transient oscillations. In interconnected systems, stability ensures that state trajectories remain bounded and converge despite uncertainties, a prerequisite for control objectives such as robotic manipulation, electric vehicles, and deteriorating systems [1–3].

Building upon this bedrock of stability, tracking control represents one of the most critical advanced objectives in engineering. It seeks to synthesize control inputs that drive a system's output to asymptotically follow a prescribed reference trajectory. This encompasses both set-point regulation and dynamic path following, representing a central challenge across domains ranging from industrial automation, where precise regulation of process variables is essential [4], to advanced robotics, which demands accurate trajectory execution in high-precision tasks [5]. To address this ubiquitous challenge systematically, output regulation theory provides a principled and rigorous framework for achieving exact tracking. It formally translates the controller synthesis problem into the algebraic solution of regulator equations [6,7], offering guarantees for asymptotic tracking and disturbance rejection under the assumption of a perfectly known plant model [6,8]. This framework elegantly unifies the treatment of references

and disturbances generated by an exogenous system and has been successfully applied in robotics, power systems, and networked control [9–12].

The theoretical guarantees of classical output regulation typically rely on the availability of an accurate system model, which serves as an assumption often violated in complex real-world applications due to unmodeled dynamics, parametric variations, and operational uncertainties. This mismatch between ideal models and practical conditions restricts direct applicability and drives the need for control strategies that are robust or adaptive to model errors. In this context, if system identification is performed under the prior knowledge that the underlying system is stabilizable, and if the identification process itself is constrained to preserve or enforce stabilizability, the resulting model can better support the synthesis of reliable regulators. Such an approach narrows the gap between theoretical design and practical implementation by ensuring that stability, a fundamental requirement, is embedded early in the modeling and control pipeline.

While the aforementioned strategies seek to refine or robustify the model-based design paradigm, a different approach circumvents the need for an explicit model altogether. This is exemplified by reinforcement learning (RL), which has emerged as a powerful, model-free paradigm for learning optimal control policies through direct interaction with an environment [13]. Its capacity to optimize performance without an explicit model makes it highly attractive for optimal control under uncertainty [14]. RL methodologies have been developed for optimal control of both discrete-time (DT) and continuous-time systems [15–19], with extensions addressing optimal tracking [20,21], output-feedback scenarios [22–24], and multi-agent coordination [25,26], often by synergistically integrating with the output regulation framework [27,28].

Recent advancements in adaptive and data-driven approaches have been developed to tackle the output regulation problem for DT linear systems with unknown dynamics. In [29], an adaptive DT identification scheme is used to update the internal model, enabling asymptotic regulation under unknown exosystem dynamics. Further developments in adaptive dynamic programming-based methods were introduced in [30,31], where optimal output regulation is achieved through output feedback, without requiring precise system models or explicitly solving the regulator equations. More recently, a data-driven framework was proposed in [32], which computes regulator gains directly from input-output data, ensuring closed-loop stability for constrained DT systems. Recent 2025 studies have further extended this line to continuous-time data-driven optimal output regulation via the internal model principle, output-feedback adaptive optimal output regulation without an admissible initial policy, and discrete-time nonlinear stochastic output regulation [33–35]. Additional studies, such as [36–39], have explored related techniques, including RL-based designs, adaptive cooperative regulation for multi-agent systems, and data-driven approaches to cooperative output regulation.

Despite these advances, a limitation persists in the literature on RL-based tracking control. Most prevalent policy iteration (PI) algorithms still require an initial stabilizing controller and, more importantly, generally lack a priori guarantees on the state convergence rate [40,41]. While achieving tracking with a prescribed convergence speed has been studied within the robust output regulation framework, existing solutions typically rely on tuning a scalar parameter, sacrificing design freedom that could better accommodate the dynamics of the original system, for example, when the system has only one or a few unstable poles.

Motivated by the discussion above, the central gap identified is the absence of a data-driven, RL-based method that simultaneously ensures robust asymptotic tracking without knowing system dynamics. To bridge this gap, this paper introduces an integration of robust output regulation with data-driven RL with a diagonal-scaling technique for PI. The key innovation is the advancement from conventional scalar-based rate-scaling to a diagonal-scaling regulation mechanism. This generalized formulation provides substantially greater design freedom for eigenvalue assignment compared to scalar designs, enabling more precise and flexible shaping of the transient convergence profile. Within this framework, we develop a novel data-driven controller that embeds the diagonal-scaling rate condition directly into the RL learning process. This approach guarantees asymptotic tracking at the prescribed rate while maintaining robustness, all without requiring an explicit system model, knowledge of the system output matrix, or continuous-time data recording, enhancing both performance tunability and online computational efficiency compared to prior linear output

regulation designs [41]. The proposed framework is particularly suited for applications such as networked motion control, automated production lines, and energy systems, where model-free operation and reliable tracking are essential for maintaining operational integrity.

The remainder of this paper is organized as follows. Section 2 formulates the control problem and states the RL objective without knowing the system dynamics for seeking the initial stabilizing control. Section 3 develops a model-based RL algorithm using the diagonal-scaling regulation mechanism. Section 4 proposes a model-free state-feedback-based RL algorithm, followed by an improved version that incorporates diagonal-scaling regulation. The efficacy of the proposed algorithms is demonstrated through simulation examples in Section 5. Finally, Section 6 concludes the paper.

Notations: Throughout this paper, $\mathbb{R}$ denotes the set of real numbers, and $\otimes$ denotes the Kronecker product. For a matrix $X \in \mathbb{R}^{m\times n}$, define $\mathrm{vec}(X) = [X_1^T\ X_2^T\ \cdots\ X_n^T]^T$, where X_i denotes the i-th column of X.

2. Preliminaries and problem formulation

2.1. Modelling and preliminaries

We now consider a class of DT linear dynamical systems:

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = Cx(k) \tag{2}$$

where $x(k) \in \mathbb{R}^{c_x}$, $u(k) \in \mathbb{R}^{c_u}$, and $y(k) \in \mathbb{R}^{c_y}$ represent the state, input, and output vectors, respectively. The matrices $A \in \mathbb{R}^{c_x\times c_x}$, $B \in \mathbb{R}^{c_x\times c_u}$, and $C \in \mathbb{R}^{c_y\times c_x}$ are constant. This model provides a mathematical framework for predicting and controlling the evolution of a system over time. The state $x(k)$ encapsulates the system's complete internal condition (e.g., a vehicle's position and velocity), the input $u(k)$ represents an applied control action (e.g., a throttle command), and the output $y(k)$ denotes the measurable quantities (e.g., a radar distance). The matrix A governs the natural, unforced evolution of the state, B maps how inputs drive the state forward, and C projects the internal state onto the observable output space. This versatile formulation underpins an array of modern control and estimation technologies, from the car-following logic in adaptive cruise control and the sensor fusion algorithms in satellite navigation to the frequency regulation mechanisms in electric power systems.

The reference trajectory to be tracked is generated by an exogenous system:

$$x_r(k+1) = Sx_r(k) \tag{3}$$

$$y_r(k) = Rx_r(k) \tag{4}$$

where $x_r(k) \in \mathbb{R}^{c_r}$ is the reference state and $y_r(k) \in \mathbb{R}^{c_y}$ is the desired output. The matrices $S \in \mathbb{R}^{c_r\times c_r}$ and $R \in \mathbb{R}^{c_y\times c_r}$ are known and constant. Consequently, the output tracking error is defined as:

$$\mathrm{y_e(k)} = \mathrm{y(k)} - \mathrm{y_r(k)} \tag{5}$$

The problem setup rests on the following standard assumptions in output regulation theory:

Assumption 1: The pair (A, B) is controllable, and the pair (A, C) is observable.

Assumption 2: The matrix S is anti-stable, i.e., all its eigenvalues lie on or outside the unit circle.

Assumption 3: The minimal polynomial of S is known.

Assumption 4: For the solvability of the output regulation problem, the following rank condition is assumed to hold: $rank(\begin{bmatrix} A-\lambda_i I & B \\ C & 0 \end{bmatrix}) = c_x + c_y,\ \forall \lambda_i \in \lambda(S)$.

2.2. Problem formulation

To achieve asymptotic tracking, we construct an internal model-based dynamic controller:

$$s(k+1) = Fs(k) + Gy_e(k) \tag{6a}$$

$$u(k) = -Kx(k) - Hs(k) - Ts(k) \tag{6b}$$

where $s(k) \in \mathbb{R}^{c_y c_r}$ is the internal model state, and (F, G) is a c_y-copy of the matrix S. The matrices K, H, and T are controller gains to be designed.

Defining the augmented state $\hat{x}(k) = [x^T(k), s^T(k)]^T \in \mathbb{R}^{c_{\hat{x}}}$ with $c_{\hat{x}} = c_x + c_y c_r$, then the DT system formula Eqs. (1) and (2) can be further written as:

$$\mathrm{\hat{x}(k+1) = \hat{A}\hat{x} + \hat{G}Rx_r(k)} \tag{7a}$$

$$\mathrm{y_e(k) = \hat{C}\hat{x}(k) - Rx_r(k)} \tag{7b}$$

with $\mathrm{\hat{A} = \begin{bmatrix} A-BK & -BH-BT \\ -GC & F \end{bmatrix} \in \mathbb{R}^{c_{\hat{x}}\times c_{\hat{x}}}}$, $\mathrm{\hat{G} = \begin{bmatrix} 0 \\ G \end{bmatrix} \in \mathbb{R}^{c_{\hat{x}}\times c_y}}$ and $\mathrm{\hat{C} = [C\quad 0] \in \mathbb{R}^{c_y\times c_{\hat{x}}}}$. Further decompose and refine the matrix $\mathrm{\bar{A}}$:

$$\hat{A} = \begin{bmatrix} A-BK & -BH-BT \\ -GC & F \end{bmatrix} = \begin{bmatrix} A & -BT \\ -GC & F \end{bmatrix} - \begin{bmatrix} BK & BH \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} A & -BT \\ -GC & F \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} \cdot [K \quad H] \tag{8}$$

Let $\underline{A} = \begin{bmatrix} A & -BT \\ -GC & F \end{bmatrix} \in \mathbb{R}^{c_{\hat{x}}\times c_{\hat{x}}}$, $\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \in \mathbb{R}^{c_{\hat{x}}\times c_u}$ and the feedback gain matrix $\bar{K} = [K, H] \in \mathbb{R}^{c_{\bar{K}}}$, which $c_{\bar{K}} = c_u \times$

$c_y c_{x_s}$. Thus, matrix $\hat{A} = \underline{A} - \hat{B}\hat{K}$.

Therefore, the control system's dynamic properties should be as follows:

Lemma 1: When (F, G) is the c_y-copy internal model of matrix S and Assumption 4, given any matrix T:

1. If (A, B) is stabilizable, $(\underline{A}, \hat{B})$ is stabilizable.

2. If (F, T) is stabilizable when $c_y \geq c_u$, given that (A, C) is observable, $(\underline{A}, \hat{C})$ is observable.

Thus, under **Assumptions 1-4**, when $\hat{A}$ is Schur, X is the solution of the functions:

$$\begin{cases} \hat{A}X + \hat{G} = XS \\ \hat{C}X = R \end{cases} \tag{9}$$

Thus, the error $y_e(k)$ could be defined as:

$$e(k) = \hat{x}(k) - Xx_r(k) \tag{10}$$

With Eqs. (2), (8), (9) and (10), the functions can be written as:

$$e(k+1) = \underline{A}e(k) + \hat{B}u_e(k) \tag{11a}$$

$$y_e(k) = \hat{C}e(k) \tag{11b}$$

$$u_e(k) = -\hat{K}e(k) \tag{11c}$$

where the feedback matrix $\hat{K}$ should be designed based on the cost function for Eqs. (11a) and (11b) is given by:

$$J(e(k), u_e(k)) = \sum_{i=k}^{\infty} (y_e^T(i) Q y_e(i) + u_e^T(i) \hat{R} u_e(i)) \tag{12}$$

with $Q > 0$ and $\hat{R} > 0$.

To stabilize $e(k)$ in Eq. (10) to zero and minimize the cost function, the optimal feedback gain matrix $\hat{K}^*$ is designed with:

$$\hat{K}^* = (\hat{R} + \hat{B}^T P \hat{B})^{-1} \hat{B}^T P \underline{A} \tag{13}$$

and P can be solved with the condition of the DT Algebraic Riccati Equation (ARE):

$$\underline{A}^T P \hat{B} (\hat{R} + \hat{B}^T P \hat{B})^{-1} \hat{B}^T P \underline{A} = \hat{C}^T \hat{Q} \hat{C} + \underline{A}^T P \underline{A} - P \tag{14}$$

where $\hat{Q} = diag\{Q, 0\}$.

Given that $(\underline{A}, \hat{B})$ satisfies the stabilization condition and $(\underline{A}, \hat{C})$ satisfies the observability criterion, the optimal value matrix P^* can be determined first, from which the optimal feedback gain matrix K^* is subsequently derived. However, the conventional solution approach for the DT ARE requires precise knowledge of the dynamical models, which substantially increases the complexity involved in computing the optimal feedback gain matrix $\hat{K}^*$. In light of this constraint, we opt to employ the PI technique as an alternative to the direct resolution of the DT ARE. Thus, the solution of P^j and K^j in PI should be as follows:

$$P^j = (\underline{A} - \hat{B}\hat{K})^T P^j (\underline{A} - \hat{B}\hat{K}) + (\hat{K}^j)^T \hat{R} \hat{K}^j + \hat{C}^T \hat{Q} \hat{C} \tag{15a}$$

$$K^{j+1} = (\hat{R} + \hat{B}^T P^j \hat{B})^{-1} \hat{B}^T P^j \underline{A} \tag{15b}$$

As previously mentioned, the PI algorithm provides an effective solution for the DT Algebraic Riccati Equation. Next, we present Algorithm 1, which demonstrates the actual implementation.

Algorithm 1: PI for DT Algebraic Riccati Equation

Input: System matrices $\underline{A} \in \mathbb{R}^{c_{\hat{x}} \times c_{\hat{x}}}$, $\hat{B} \in \mathbb{R}^{c_{\hat{x}} \times c_u}$;
$\hat{C} \in \mathbb{R}^{c_y \times c_{\hat{x}}}$;
Weight matrices $Q, \hat{R} \in \mathbb{R}$;
Suitable initial gain matrix $\hat{K}^0$;
Tolerance ϵ;
Output: Optimal feedback gain matrix $\hat{K}^*$;
Value function matrix P^*;

```
Initialize j = 0;
repeat
    Solve (16) for P^j under the current feedback gain:
        P^j = (A - B K^j)^T P^j (A - B K^j)
        + (K^j)^T R K^j + C^T Q C;                    (16)
    Calculate the updated feedback gain from (17):
        K^{j+1} = (R + B^T P^j B)^{-1} B^T P^j A;     (17)
    if ||K^{j+1} - K^j|| ≤ ε then break;
    Set K^j ← K^{j+1} and j ← j + 1;
until the convergence condition is satisfied;
return K* = K^{j+1}, P* = P^{j+1}.
```

As shown in **Algorithm 1**, to prevent the entire algorithm from falling into an infinite loop due to overly stringent convergence criteria, a tolerance range ϵ is established. The loop terminates when the errors between $\hat{K}^{j+1}$ and $\hat{K}^j$, and between P^{j+1} and P^j fall within ϵ. Thus, it is recommended to select a minuscule positive constant value for ϵ, for instance, $1e^{-4}$, to serve as a convergence threshold or numerical tolerance. The sequence $\hat{K}^j$ converges to the optimal gain $\hat{K}^*$.

It is critical to note that the convergence and stability of the PI algorithm are guaranteed only when initiated from an admissible control policy, i.e., an initial gain $\hat{K}^0$ that stabilizes the pair $(\underline{A}, \hat{B})$. This requirement of an initial stabilizing solution is a premise for the application of the iterative procedure. The challenge addressed in subsequent sections is to develop model-free RL algorithms that achieve this optimal regulation without relying on prior knowledge of $(\underline{A}, \hat{B}, \hat{C})$, while respecting the necessity of iterative stability.

3. Main results: diagonal scaling for policy initialization in PI

In the PI-based design paradigm, it has been shown that

choosing a suitable initial feedback gain matrix K^0 is necessary. K^0 is not required to be optimal. Its primary requirement is to ensure stability at system launch initially through the constraint of the spectral radius: $\rho(\underline{A} - \hat{B}\widehat{K}) < 1$. In [40], a value-iterative method is shown for opting to self-select K^0 within the defined constraint of the spectral radius. This method cannot prove K^0 is close to the best feedback gain matrix K^* under PI. It means that the selected K^0 should satisfy $\hat{A}$ is Schur. $\widehat{K}$ can only affect $\hat{B}\widehat{K}$ in $\hat{A}$, thus $\underline{A}$ becomes an independent variable that $\widehat{K}$ is hard to affect. When $\rho(\hat{A}) > 1$, the whole system cannot be stable. However, when the system has unstable poles, especially when $\rho(\underline{A})$ is significantly greater than one, finding an initial gain K^0 that stabilizes the closed-loop system becomes a critical issue in PI. Thus, we propose using a virtual control system to solve K^0 first and then use the suitable K^0 to solve the tracking problem.

Different from the actual control system in Eqs. (6) and (7), the virtual control system is redefined as follows:

$$s_v(k+1) = Fs_v(k) + Gy_e(k) \tag{18a}$$

$$u_v(k) = -Kx_v(k) - Hs_v(k) - Ts_v(k) \tag{18b}$$

$$\hat{x}_v(k) = \hat{A}\hat{x}_v(k) + \hat{G}Rx_{r_s}(k) \tag{18c}$$

$$y_{e_v}(k) = \hat{C}\hat{x}_v(k) - Rx_{r_s}(k) \tag{18d}$$

Thus, in the virtual control system, the formula Eqs. (10) and (11) are further written as:

$$e_v(k) = \hat{x}_v(k) - Xx_{r_s}(k) \tag{19a}$$

$$e_v(k+1) = \underline{A}e_v(k) + \hat{B}u_{e_v}(k) \tag{19b}$$

$$y_{e_v}(k) = \hat{C}e_v(k) \tag{19c}$$

$$u_{e_v}(k) = -\widehat{K}_v e_v(k) \tag{19d}$$

Then, we can find a suitable initial feedback gain matrix K^0 in this virtual control system. As shown in [40], find K^0 based on this rule:

$$\rho(\hat{A}) = \rho(\underline{A} - \hat{B}\widehat{K}) \leq 1 \tag{20}$$

To solve this problem, we propose using a virtual diagonal matrix Λ to achieve the pole placement. Thus, Eqs. (18c) and (19b) are further rewritten as follows:

$$\hat{x}_v(k+1) = \hat{A}_\alpha\hat{x}_v(k) + \hat{G}Rx_{r_s}(k) \tag{21a}$$

$$e_v(k+1) = \underline{A}_\alpha e_v(k) + \hat{B}_\alpha u_{e_v}(k) \tag{21b}$$

and

$$\rho(\hat{A}_\alpha) = \rho(\underline{A}_\alpha - \hat{B}_\alpha\widehat{K}_v) = \rho(\Lambda^{-1}(\underline{A} - \hat{B}\widehat{K}_v) \tag{22}$$

where $\hat{A}_\alpha = \Lambda^{-1}\hat{A}$, $\underline{A}_\alpha = \Lambda^{-1}\underline{A}$, $\hat{B}_\alpha = \Lambda^{-1}\hat{B}$, and $\widehat{K}_v$ denotes the feedback gain in the virtual system. Here, Λ is a positive diagonal matrix. By scaling $\underline{A}$ with Λ^{-1}, the spectral radius of the closed-loop virtual system can be reduced, thereby facilitating the selection of a stabilizing $\widehat{K}_v$. Thus, without $\hat{B}\widehat{K}_v$, Eq. (22) is followed as:

$$\rho(\hat{A}_\alpha) = \rho(\underline{A}_\alpha) < 1 \tag{23}$$

Therefore, to ensure that the spectral radius $\rho(\underline{A}_\alpha) < 1$, this condition must be satisfied: $\rho(\Lambda^{-1}\underline{A}) < 1$. Thus, $\widehat{K}_v$ can be solved in the virtual control system. In this case, $\widehat{K}_v$ cannot be treated as equivalent to $\widehat{K}^0$. Thus, we have developed an iterative method to make $\Lambda \to I_n$ to ensure that $\widehat{K}_v$ can be considered equivalent to $\widehat{K}^0$.

Note that our work generalizes the scaling gain to a matrix form, offering greater design flexibility for system stabilization. Unlike the scalar schemes commonly found in the literature, our approach enables the tuning of all eigenvalues in the closed-loop system through appropriate selection of the diagonal matrix. This provides a more precise means of pole placement, which is particularly valuable in practice, as engineering systems often exhibit partially unstable poles rather than being entirely unstable.

Building upon the diagonal scaling-based framework introduced previously, we now analyze the individual scaling factor associated with each diagonal element. To obtain a well-conditioned initial stabilizing feedback gain matrix for subsequent learning, a diagonal matrix Λ is introduced and iteratively steered toward its nominal value of $\Lambda = I_n$. A key design requirement is that this iteration must converge efficiently while avoiding undesirable oscillation around $\alpha_j = 1$ in Λ_j, a common pitfall of fixed-step update schemes. To address this, the updated law for α_j is equipped with a crossing-triggered reset mechanism and a sign-based direction encoding. Both mechanisms admit a concise mathematical interpretation that aligns with the underlying program logic. An iterative scheme for α_j is then developed based on these two concepts.

Inspired by principles of velocity and acceleration from physics, the two mechanisms are merged to dynamically modulate the iteration pace of α_j in real-time according to the system's operational state. This promotes a rapid yet stable reduction of α_i toward unity. The resulting iteration formulas are given as follows:

$$\alpha_{j+1} = \alpha_j + \Delta\alpha_{j+1} \tag{24}$$

$$\Delta\alpha_{j+1} = \mu_g\Delta\alpha_j - Acc_j \tag{25}$$

$$\mu_g = g_j \cdot \mu_v \tag{26}$$

$$g_{j+1} = \max\left(0, sgn(\alpha_j - 1) \cdot sgn(\alpha_{j+1} - 1)\right) \quad (27)$$

where α_j is the current scaling value and α_{j+1} is the next value. The quantities $\Delta\alpha_j, g_j, \varepsilon_j, Acc_j$ are available at iteration j and are used to form $\Delta\alpha_{j+1}$ and α_{j+1}. After the update, g_{j+1} records whether the transition crosses the baseline $\alpha = 1$. It implements a crossing-triggered reset mechanism based on the sign function defined as:

$$sgn(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases} \quad (28)$$

Baseline crossing detection is therefore performed between two successive accepted values, α_j and α_{j+1}. A negative sign product indicates a crossing, while a zero product indicates that the newly accepted value has reached the target baseline.

After α_{j+1} has passed the stability test, the corresponding spectral-radius trend is stored for the next update and is defined as:

$$\varepsilon_{j+1} = \frac{|1-\rho(\hat{A}_{\alpha_{j+1}})|}{|1-\rho(\hat{A}_{\alpha_j})|} \quad (29)$$

The direction and adjustment terms associated with the newly accepted state are then prepared for the next outer iteration:

$$Acc_{j+1} = \varepsilon_{j+1}\mu_s(\alpha_0 - 1) \cdot g_p \quad (30)$$

$$g_p = sgn(\alpha_{j+1} - 1) \quad (31)$$

where μ_s is a constant factor and g_p is the factor that decides the direction of Acc_j.

To enhance the controllability of the initial α_0, we set $\widehat{K}^0 = 0$. Consequently, Eq. (23) can be formulated as:

$$\alpha_0{}^{-1}\rho(\underline{A}) < 1 \quad (32)$$

Therefore, the value of α_0 must be greater than $\rho(\underline{A})$ to ensure that the division of the two numbers is less than 1. The specific setting of α_0 will be discussed in the simulation section. To provide in-depth elucidation of the operational mechanism and advantages of our proposed adaptive step-size iterative method, this section presents a series of illustrative diagrams that detail the method's dynamic behavior during actual iterations. These graphical representations help demonstrate how the adaptive step-size algorithm accelerates convergence through step-size regulation while ensuring system stability. By ensuring that the closed-loop spectral radius remains strictly within the unit circle throughout the adaptation process, the method guarantees that the system never enters an unstable or oscillatory regime. This is critical for maintaining component integrity and avoiding unscheduled maintenance interventions.

The iteration process of the scaling factor can be depicted in Fig. 1, which tracks the evolution of α_j over $j = 0, \ldots, 4$. Starting from an initial value greater than 1, α_j first decreases, then undergoes a faster decrease, and subsequently turns upward after crossing the nominal point $\alpha = 1$. This behavior reflects the adaptive step-size mechanism of the proposed iteration law, in which the update direction is adjusted according to the relative position of α_j with respect to 1, while the update magnitude is further modulated by the convergence-related quantities introduced later. The two turning stages around $\alpha = 1$ indicate that the algorithm does not enforce a rigid monotonic trajectory, but instead allows moderated overshoot and correction so that α_j can gradually approach the desired neighborhood of 1. The horizontal dashed line marks the reference level $\alpha_j = 1$, which serves as the target value of the scaling iteration.

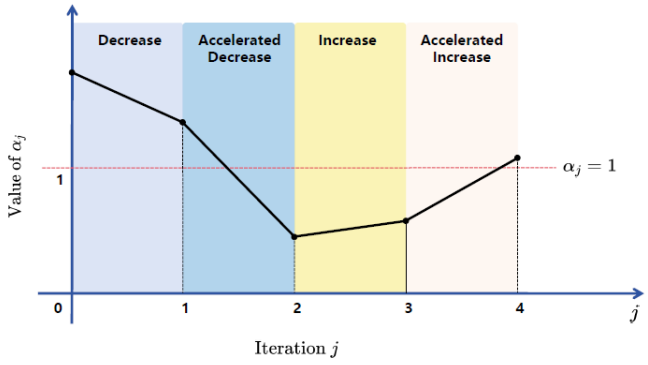

Figure 1. Iterative evolution of the scaling factor α_j.

Complementing the scaling factor dynamics, Fig. 2 shows the evolution of the spectral radius $\rho(\hat{A}_{\alpha_j})$ together with the associated distance measure $\text{Dis}_{1-\rho(\hat{A}_{\alpha_j})}$ over the same iteration horizon. The plotted trend is constructed to reflect the expected variation of the closed-loop spectral radius during the α-iteration process. Initially, the spectral radius is larger than 1, indicating that the closed-loop system is not yet Schur stable. As the iteration proceeds, $\rho(\hat{A}_{\alpha_j})$ decreases and crosses the stability boundary, then rises again in response to the change of the scaling direction. Correspondingly, the distance to the boundary $1 - \rho(\hat{A}_{\alpha_j})$ also varies from step to step and provides a direct quantitative indication of how far the current closed-loop system is from the critical stability threshold. This figure therefore serves as the basis for constructing the subsequent adaptive quantities, since the iteration of all related parameters

is designed according to the anticipated variation of this spectral-radius distance.

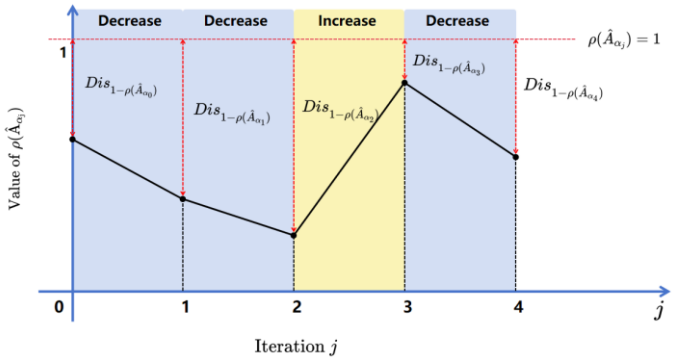

Figure 2. Evolution of the spectral radius and its distance to the stability boundary during α-iteration process.

Building on these spectral radius variations, Fig. 3 presents the evolution of ε_j for $j = 0, \ldots, 4$, where ε_j is defined as the ratio between the spectral-radius distance in the current iteration and that in the previous iteration. Hence, ε_j characterizes the relative change in the stability margin from one step to the next. When $\varepsilon_j > 1$, the current spectral radius is farther away from the critical value 1 than in the previous iteration, which implies that the closed-loop system moves towards a more stable region. In contrast, when $\varepsilon_j < 1$, the current spectral radius becomes closer to the boundary, indicating a weakened stability tendency. Therefore, ε_j provides an indicator of the instantaneous evolution trend of the spectral radius, and it can be used to adaptively accelerate or slow down the decrease or increase of α_j in the iteration.

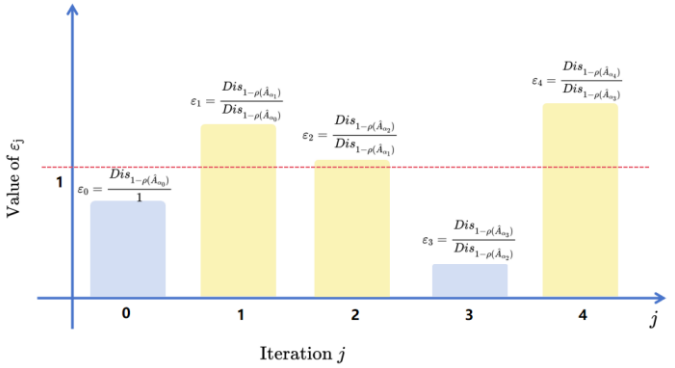

Figure 3. Variation of the stability - trend indicator ε_j over the iterations.

The directional information necessary for guiding the update is encoded in the sign-related quantity g_{p_j}, whose evolution over iterations $j = 0, \ldots, 4$ is shown in Fig. 4. This parameter is introduced to identify whether the current α_j is located above or below the nominal value 1. Specifically, g_{p_j} takes a positive value when $\alpha_j > 1$, and a negative value when $\alpha_j < 1$. In this way, g_{p_j} encodes the directional information of the scaling factor relative to the target point. The role of this parameter is to guarantee that the iterative law drives α_j downward when it lies above 1, and upward when it lies below 1, ensuring that the update direction always tends to move the scaling factor back toward the desired neighborhood of $\alpha = 1$.

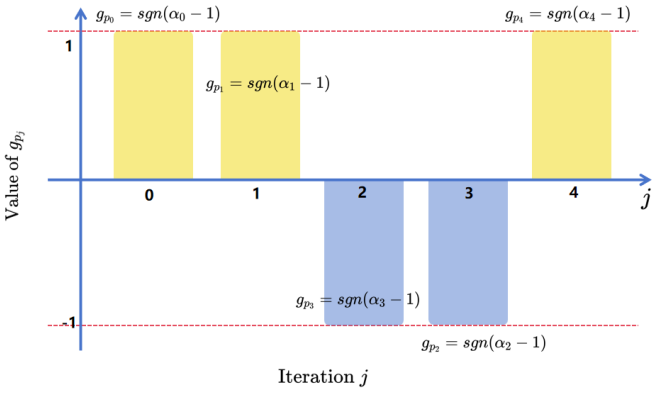

Figure 4. Directional sign parameter g_{p_j}.

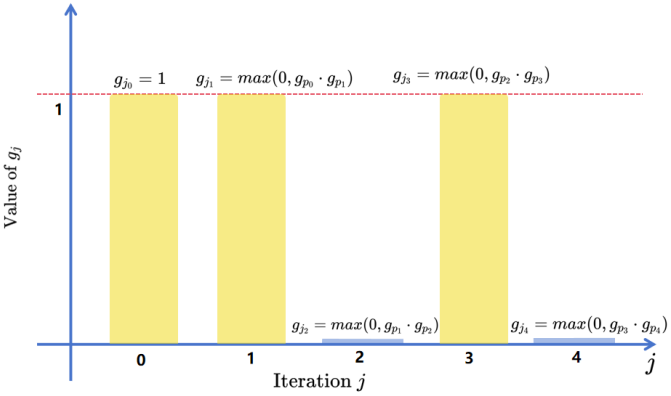

Figure 5. Evolution of the braking factor g_j for crossing detection around $\alpha = 1$.

To prevent erratic behavior when the scaling factor crosses the target value, a braking mechanism is introduced through the factor g_j, whose iterative progression is depicted in Fig. 5. This quantity is defined through a max-operation so as to detect whether the sign of $\alpha_j - 1$ changes between two consecutive iterations. When α_j remains on the same side of 1 as in the previous step, $g_j = 1$; however, once α_j changes from above 1 to below 1, or vice versa, g_j becomes 0. Therefore, g_j acts as a braking mechanism in the proposed adaptive iteration law. In particular, when the trajectory of α_j crosses the reference value 1, the update increment $\Delta\alpha_j$ is directly reset to zero, preventing the iteration from moving too aggressively after the crossing. This mechanism effectively suppresses excessive oscillation

and improves the smoothness of the adjustment process near the target point.

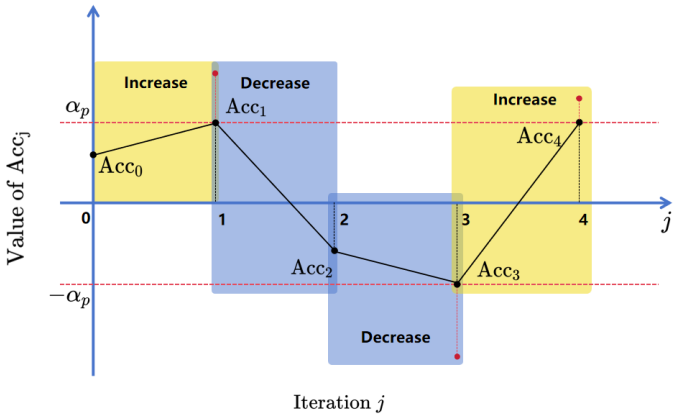

Figure 6. Evolution of the adaptive acceleration term Acc_j in the update process.

The update magnitude itself is governed by the acceleration term Acc_j, whose behavior can be illustrated in Fig. 6. The value of Acc_j is jointly influenced by the stability-trend indicator ε_j and the directional parameter gp_j. More specifically, the distance between the initial scaling factor α_0 and the target value 1, multiplied by the coefficient μ_s, is taken as the basic reference magnitude, and this magnitude is then enlarged or reduced according to the current stability tendency reflected by ε_j. Meanwhile, g_{p_j} determines the sign of the acceleration term so that the update always points toward the correcting direction. To avoid overly aggressive updates, Acc_j is constrained within the interval $[-\alpha_p, \alpha_p]$; once this range is exceeded, it is forcibly clipped to $-\alpha_p$ or α_p. This saturation mechanism prevents $\Delta\alpha_j$ from becoming excessively large in a single iteration, which could otherwise drive the spectral radius across the threshold and impair the stability of the entire control system. Here, α_p is chosen as a reduced quantity derived from α_0.

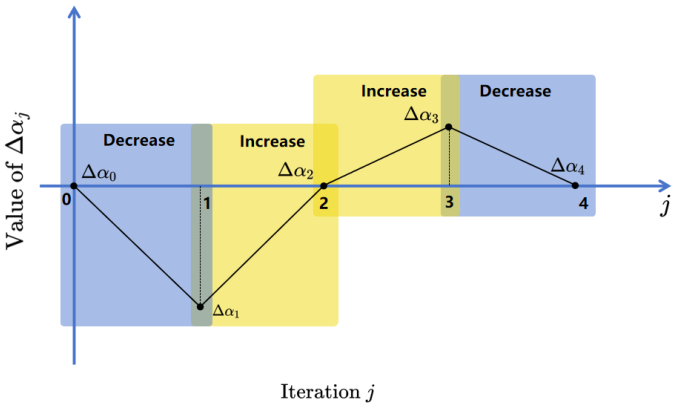

Figure 7. Iterative update increment $\Delta\alpha_j$ generated by the adaptive mechanism.

Fig. 7 depicts the iterative values of $\Delta\alpha_j$, obtained according to the assumed initial α_j and the corresponding spectral-radius evolution $\rho(\hat{A}_j)$. The figure shows how the update increment changes from step to step under the combined influence of the preceding adaptive parameters. In particular, the values at $j = 2$ and $j = 4$ are reset to zero because, at these two iterations, α_j crosses the reference value 1 once from above and once from below, respectively. Owing to the action of the braking factor g_j, the corresponding update increments are canceled, which prevents an excessively rapid reversal of the iteration direction. This behavior demonstrates that $\Delta\alpha_j$ is not assigned heuristically, but is generated by a coordinated mechanism involving the spectral-radius trend, directional encoding, acceleration adjustment, and crossing-triggered reset. We now design the diagonal matrix Λ_j as follows:

$$\Lambda_j = diag\left(\alpha_j(1), \alpha_j(2), \ldots, \alpha_j(n)\right) \tag{33}$$

where $\alpha_j(i)$ are chosen according to Eqs. (24) and (32).

Algorithm 2: Diagonal-Scaling-based Model-based Virtual Control System for Suitable initial feedback gain matrix $\hat{K}_0^*$ based on PI

Input: System matrices $\underline{A} \in \mathbb{R}^{c_{\hat{x}} \times c_{\hat{x}}}$, $\hat{B} \in \mathbb{R}^{c_{\hat{x}} \times c_u}$;
$\hat{C} \in \mathbb{R}^{c_y \times c_{\hat{x}}}$;
Weight matrices $Q, \hat{R} \in \mathbb{R}$;
Initial gain $\hat{K}_v^0 = 0$;
Constant value $\mu, \delta \in (0,1)$;
Scaling factor $\alpha_0(i) > \rho(\underline{A})$;
Tolerance ϵ, loop limit τ;
Count of $\rho(\hat{A}_{\alpha_j}) \geq 1$ break_count;

Output: Optimal feedback gain matrix K_0^*;

Initialize j = 0, $\hat{K}_v^0 = 0$, $\Lambda_0 = \mathrm{diag}(\alpha_0(1), \ldots, \alpha_0(n))$, and break_count = 0;
repeat
At the current Λ_j, launch the virtual control system and solve (34):

$$P_v^j = (\underline{A}_{\alpha_j} - \hat{B}_{\alpha_j}\hat{K}_v^j)^T P_v^j (\underline{A}_{\alpha_j} - \hat{B}_{\alpha_j}\hat{K}_v^j) + (\hat{K}_v^j)^T \hat{R} \hat{K}_v^j + \hat{C}^T \hat{Q} \hat{C}; \tag{34}$$

Update the feedback gain by

$$\hat{K}_v^{j+1} = (\hat{R} + \hat{B}_{\alpha_j}^T P_v^j \hat{B}_{\alpha_j})^{-1} \hat{B}_{\alpha_j}^T P_v^j \underline{A}_{\alpha_j}; \tag{35}$$

Repeat (34)-(35) at fixed Λ_j until $\|\hat{K}_v^{j+1} - \hat{K}_v^j\| \leq \varepsilon$;
if $\rho(\hat{A}_{\alpha_j}) \geq 1$ then
    if break_count $\geq \tau$ then terminate with failure; end if
    Replace the rejected scaling matrix by $\Lambda_j \leftarrow \Lambda_{j-1} + 0.5(\Lambda_j - \Lambda_{j-1})$;
    Set $\Delta\alpha_j = 0$, break_count $\leftarrow$ break_count + 1, and repeat at the same j;
else
    Set break_count = 0;
    if $\|\Lambda_j - I_n\| \leq \varepsilon$ then return $\hat{K}_0^* = \hat{K}_v^{j+1}$; end if
    Update Λ_{j+1} with Eqs. (33) and (24), and set j $\leftarrow$ j + 1;
end if
until $\|\Lambda_j - I_n\| \leq \varepsilon$;
return the suitable initial feedback gain matrix $\hat{K}_0^* = \hat{K}_v^{j+1}$.

The specific iterative process algorithm for the virtual system is presented in Algorithm 2. In the scaling-based framework, $\underline{A}_{\alpha_j} = \alpha_j^{-1}\underline{A}$ and $\hat{B}_{\alpha_j} = \alpha_j^{-1}\hat{B}$, where the baseline is $\alpha_j = 1$. Therefore, in Step 4 of Algorithm 2, the corresponding condition is modified to $|\alpha_j - 1|$. Since the diagonal matrix form offers more degrees of freedom, and a single parameter is considered a special case, we will prioritize discussing the diagonal matrix form in the subsequent text.

To prevent instability of $\rho(\hat{A}_{\alpha_j})$ during iteration, the scale-based framework incorporates a corresponding safe fallback mechanism, which is demonstrated in both model-driven and data-driven virtual control systems. When the virtual system becomes unstable, in other words, $\rho(\widehat{A}_{\alpha_j}) \geq 1$, $\Delta\alpha_j$ is reset to zero and $\Lambda_{j+1} \leftarrow \Lambda_j + 0.5(\Lambda_{j+1} - \Lambda_j)$. The system stability is then rechecked. If stability is restored, the process continues; otherwise, the fallback loop repeats until the system stabilizes or the loop count exceeds τ iterations. This loop limit τ is a user-defined positive integer.

4. State-feedback data-based control system and its solution via PI

In many industrial settings, obtaining a precise dynamical model is infeasible due to aging components, nonlinearities, or varying operating conditions. The proposed data-driven approach enables reliable control design using only measured input-output data, making it suitable for retrofitting existing machinery and for systems where model maintenance is impractical.

4.1. State-Feedback control system formulation

We begin by formulating the state-feedback control system, which serves as the foundation for our subsequent analysis. In the state-feedback setting, the control input relies solely on the available state information through a feedback gain matrix. Consequently, the controller can be expressed as:

$$u_s(k) = -\widehat{K}^0\hat{r}(k) + \Theta(k) \tag{36}$$

where $\hat{r}(k) = [x^{\mathrm{T}}(k) \quad \hat{s}^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{c_{\hat{x}}}$ denotes the augmented state vector, comprising the original system state $x(k)$ and an internal dynamic signal $\hat{s}(k)$. This dynamic signal $\hat{s}(k)$ is distinct from the signal $s(k)$ appearing in the internal model Eq. (6), as it is specifically constructed for the state-feedback design. The matrix $\widehat{K}^0$ represents an initial stabilizing feedback gain matrix, whose selection will be discussed in subsequent sections. The term $\Theta(k)$ denotes an exploration signal, intentionally injected to ensure sufficient excitation for data collection and learning.

The state-feedback control system is described by:

$$\hat{s}(k+1) = F\hat{s}(k) - Gy(k) + G\vartheta(k) \tag{37}$$

where $\vartheta(k)$ serves a dual purpose: it can be interpreted either as exploration noise injected for system identification purposes, or as the exogenous output $y_s(k)$ from the exosystem defined in Eq. (4). This flexibility allows our formulation to accommodate both learning and disturbance rejection scenarios.

By augmenting the original system state with this dynamic signal, we obtain the overall state-feedback system:

$$\hat{r}(k+1) = \underline{A}\hat{r}(k) + \hat{B}u_s(k) + \hat{G}\vartheta(k) \tag{38a}$$

$$y(k) = \hat{C}\hat{r}(k) \tag{38b}$$

The system matrices $\underline{A}$, $\hat{B}$, $\hat{G}$, and $\hat{C}$ are of appropriate dimensions and are assumed to be known for the data-based analysis presented in this section. Eq. (38a) describes the evolution of the augmented state under the combined influence of the control input $u_s(k)$ and the disturbance/exploration signal $\vartheta(k)$, while Eq. (38b) defines the system output $y(k)$ as a linear function of the augmented state.

4.2. Solution of the state-feedback control system via PI

With the augmented system formulation established, we now develop a PI approach to solve the optimal control problem associated with this system. Recall that the closed-loop system matrix under a feedback gain $\widehat{K}$ is given by $\hat{A} = \underline{A} - \hat{B}\widehat{K}$ as in Eq. (11). For the PI algorithm, we denote the gain at iteration j as $\widehat{K}^j$, and the corresponding closed-loop matrix as $\hat{A}^j = \underline{A} - \hat{B}\widehat{K}^j$. Using this notation, the augmented state dynamics Eq. (38a) can be rewritten in a form that facilitates the PI update:

$$\hat{r}(k+1) = \hat{A}^j\hat{r}(k) + \hat{B}(u_s(k) + \widehat{K}^j\hat{r}(k)) + \hat{G}\vartheta(k) \tag{39}$$

This decomposition separates the dynamics into three components: the closed-loop behavior under the current policy $\hat{A}^j\hat{r}(k)$, a term capturing the deviation of the actual control from the current policy $\hat{B}(u_s(k) + \widehat{K}^j\hat{r}(k))$, and the disturbance term $\hat{G}\vartheta(k)$. To derive the PI update equations, we employ a quadratic value function of the form:

$$V^j(\hat{r}(k)) = \hat{r}^{\mathrm{T}}(k)P^j\hat{r}(k) \tag{40}$$

where P^j is a symmetric positive definite matrix satisfying the Bellman equation for the current policy. Leveraging the dynamics Eq. (39) and the PI update rules Eqs. (15a) and (15b),

we obtain the following key relationship:

$$\begin{aligned}&\hat{r}^{\mathrm{T}}(k+1)P^{j+1}\hat{r}(k+1)-\hat{r}^{\mathrm{T}}(k)P^{j+1}\hat{r}(k)\\&=-\hat{r}^{\mathrm{T}}(k)(\hat{Q}+(\widehat{K}^{j})^{\mathrm{T}}\hat{R}\widehat{K}^{j})\hat{r}(k)+\vartheta^{\mathrm{T}}(k)\hat{G}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)\\&+(u_s(k)+\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}}\hat{B}^{\mathrm{T}}P^{j+1}\hat{B}(u_s(k)+\widehat{K}^{j}\hat{r}(k))\\&+2u_s^{\mathrm{T}}(k)\hat{B}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)+2\hat{r}^{\mathrm{T}}(k)\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)\\&+2\hat{r}^{\mathrm{T}}(k)\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{B}(\widehat{K}^{j}\hat{r}(k)+u_s(k))\end{aligned}\tag{41}$$

To isolate the cost term and obtain a form suitable for parameter estimation, we rearrange the equation by moving all terms except the cost to the left-hand side:

$$\begin{aligned}&\hat{r}^{\mathrm{T}}(k+1)P^{j+1}\hat{r}(k+1)-\hat{r}^{\mathrm{T}}(k)P^{j+1}\hat{r}(k)\\&-\vartheta^{\mathrm{T}}(k)\hat{G}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)\\&-2u_s^{\mathrm{T}}(k)\hat{B}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)-2\hat{r}^{\mathrm{T}}(k)\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{G}\vartheta(k)\\&-2\hat{r}^{\mathrm{T}}(k)\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{B}(\widehat{K}^{j}\hat{r}(k)+u_s(k))\\&-(u_s(k)+\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}}\hat{B}^{\mathrm{T}}P^{j+1}\hat{B}(u_s(k)+\widehat{K}^{j}\hat{r}(k))=\\&-\hat{r}^{\mathrm{T}}(k)(\hat{Q}+(\widehat{K}^{j})^{\mathrm{T}}\hat{R}\widehat{K}^{j})\hat{r}(k)\end{aligned}\tag{42}$$

For notational convenience, we define:

$$\hat{Q}^{j}\triangleq\hat{Q}+(\widehat{K}^{j})^{\mathrm{T}}\hat{R}\widehat{K}^{j}\tag{43}$$

which represents the effective state-weighting matrix when the current policy $\widehat{K}^{j}$ is employed. This definition corrects a potential ambiguity in the original formulation and clarifies that $\hat{Q}^{j}$ combines the original state cost $\hat{Q}$ with the control cost contribution from the current policy. Eq. (42) forms the cornerstone of our PI-based solution, which enables the recursive update of both the value function matrix P^{j+1} and the policy gain $\widehat{K}^{j+1}$ using data collected from the system, ultimately converging to the optimal solution that minimizes the cumulative cost. Eq. (42) can be cast into a Kronecker product representation:

$$\begin{aligned}&(\hat{r}^{\mathrm{T}}(k+1)\otimes\hat{r}^{\mathrm{T}}(k+1)-\hat{r}^{\mathrm{T}}(k)\otimes\hat{r}^{\mathrm{T}}(k))vec(P^{j+1})\\&-(\vartheta^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k))vec(\hat{G}^{\mathrm{T}}P^{j+1}\hat{G})\\&-2(u_s^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k))vec(\hat{B}^{\mathrm{T}}P^{j+1}\hat{G})\\&-2(\hat{r}^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k))vec(\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{G})\\&-2(\hat{r}^{\mathrm{T}}(k)\otimes(\widehat{K}^{j}\hat{r}(k)+u_s(k))^{\mathrm{T}})vec(\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{B})\\&-((u_s(k)+\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}}\otimes(u_s(k)+\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}})vec(\hat{B}^{\mathrm{T}}P^{j+1}\hat{B})\\&=-\hat{r}^{\mathrm{T}}(k)\hat{Q}^{j}\hat{r}(k)\end{aligned}\tag{44}$$

Define the following auxiliary matrices:

$$\begin{aligned}&L_1^{j+1}=\hat{G}^{\mathrm{T}}P^{j+1}\hat{G},L_2^{j+1}=\hat{B}^{\mathrm{T}}P^{j+1}\hat{G},L_3^{j+1}=\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{G},\\&L_4^{j+1}=\underline{\hat{A}}^{\mathrm{T}}P^{j+1}\hat{B},L_5^{j+1}=\hat{B}^{\mathrm{T}}P^{j+1}\hat{B}\end{aligned}\tag{45}$$

and the data matrices

$$\begin{aligned}M_1^{j}(k)&=\hat{r}^{\mathrm{T}}(k+1)\otimes\hat{r}^{\mathrm{T}}(k+1)-\hat{r}^{\mathrm{T}}(k)\otimes\hat{r}^{\mathrm{T}}(k),\\M_2^{j}(k)&=-\vartheta^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k),\\M_3^{j}(k)&=-2(u_s^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k)),\\M_4^{j}(k)&=-2(\hat{r}^{\mathrm{T}}(k)\otimes\vartheta^{\mathrm{T}}(k)),\\M_5^{j}(k)&=-2(\hat{r}^{\mathrm{T}}(k)\otimes(\widehat{K}^{j}\hat{r}(k)+u_s(k))^{\mathrm{T}}),\\M_6^{j}(k)&=-((u_s(k)-\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}}\otimes(u_s(k)+\widehat{K}^{j}\hat{r}(k))^{\mathrm{T}}).\end{aligned}\tag{46}$$

Using $n+1$ data samples $k_0,k_1,\dots,k_n$, we construct:

$$\Gamma_n^{j}=\begin{bmatrix}-\hat{r}^{\mathrm{T}}(k_0)\hat{Q}^{j}\hat{r}(k_0)\\-\hat{r}^{\mathrm{T}}(k_1)\hat{Q}^{j}\hat{r}(k_1)\\\vdots\\-\hat{r}^{\mathrm{T}}(k_n)\hat{Q}^{j}\hat{r}(k_n)\end{bmatrix}\tag{47}$$

$$\begin{aligned}\mathcal{H}^{j+1}=[&vec^{T}(P^{j+1}),vec^{T}(L_1^{j+1}),vec^{T}(L_2^{j+1})\\&vec^{T}(L_3^{j+1}),vec^{T}(L_4^{j+1}),vec^{T}(L_5^{j+1})]\end{aligned}\tag{48}$$

$$\Upsilon_n^{j}=\begin{bmatrix}M_1^{j}(k_0)&M_2^{j}(k_0)&\cdots&M_6^{j}(k_0)\\M_1^{j}(k_1)&M_2^{j}(k_1)&\cdots&M_6^{j}(k_1)\\\vdots&\vdots&\vdots&\vdots\\M_1^{j}(k_n)&M_2^{j}(k_n)&\cdots&M_6^{j}(k_n)\end{bmatrix}\tag{49}$$

Eq. (42) can then be expressed compactly as

$$\Upsilon_n^{j}\,\mathcal{H}^{j+1}=\Gamma_n^{j}\tag{50}$$

To guarantee a unique solution of Eq. (46), the regressor matrix Υ_n^{j} must have full column rank, i.e.

$$rank(\Upsilon_n^{j})=\dim(\mathcal{H}^{j+1})\tag{51}$$

When the rank condition is satisfied, the least squares solution of Eq. (49) is given by:

$$\mathcal{H}^{j+1}=((\Upsilon_n^{j})^{T}\Upsilon_n^{j})^{-1}(\Upsilon_n^{j})^{T}\Gamma_n^{j}\tag{52}$$

Finally, the updated feedback gain matrix is obtained as

$$\widehat{K}^{j+1}=L_4^{j+1}(\hat{R}+L_5^{j+1})^{-1}\tag{53}$$

4.3. Data-based virtual control for initial gain K^0 via PI

Unlike model-based control, a data-driven control algorithm must first collect data until the rank condition is satisfied. Only then can the entire control system be solved, yielding the optimal feedback gain matrix K^*. To provide a more intuitive illustration of the data-driven state-feedback DT systems with PI, the operational flow is summarized in Algorithm 3.

The virtual control system is constructed as:

$$\hat{r}_v(k+1)=\underline{\hat{A}}_{\alpha_j}\hat{r}(k)+\hat{B}_{\alpha_j}u_s(k)+\hat{G}_{\alpha_j}\vartheta(k)\tag{54a}$$

$$y_v(k)=\hat{C}\hat{r}_v(k)\tag{54b}$$

where Λ_j is a diagonal matrix that does not alter the dimensions of the matrices; consequently, the rank condition Eq. (48) remains unchanged. Within this virtual system, the data-based PI equation Eq. (47) takes the form:

$$\Upsilon_{n_v}^{j}\,\mathcal{H}_{\alpha_{j+1}}^{j+1}=\Gamma_{n_v}^{j}\tag{55}$$

Its least squares solution and feedback gain matrixes are given by:

$$\mathcal{H}_{\alpha_{j+1}}^{j+1} = ((\Upsilon_{n_v}^j)^T \Upsilon_{n_v}^j)^{-1} (\Upsilon_{n_v}^j)^T \Gamma_{n_v}^j \quad (56a)$$

$$\widehat{K}_v^{j+1} = L_{4,\alpha_{j+1}}^{j+1} (\hat{R} + L_{5,\alpha_{j+1}}^{j+1})^{-1} \quad (56b)$$

Algorithm 3: State-feedback Control System with off-line PI

Input: Suitable initial feedback gain matrix from virtual control system $\widehat{K}_0^*$;
Weight matrices $Q, \hat{R} \in \mathbb{R}$;
Constant value $\mu, \delta \in (0,1)$;
Large enough scaling factor α_0;
Internal model (F, G) of matrix S;
Tolerance ϵ;
Output: Optimal feedback gain matrix $\widehat{K}^*$;
Optimal controller $\hat{u}^*(k)$;

Initialize $\varepsilon_0 = 1$, $j = 0$, and $K^0 = \widehat{K}_0^*$;
repeat
Under the current policy $u_s(k) = -\widehat{K}^j \hat{r}(k) + \Theta(k)$, collect a new data batch for (50);
Continue collection until the rank condition (51) is satisfied;
Solve (52) to find L_4^{j+1} and L_5^{j+1} in $\mathcal{H}^{j+1}$;
Refresh the feedback gain matrix $\widehat{K}^{j+1}$ with (53);
if $\|\widehat{K}^{j+1} - \widehat{K}^j\| \le \varepsilon$ then break; end if
Set $\widehat{K}^j \leftarrow \widehat{K}^{j+1}$ and $j \leftarrow j + 1$;
until the convergence condition is satisfied;
return $\widehat{K}^* = \widehat{K}^{j+1}$ and $u^*(k) = -\widehat{K}^* \hat{r}(k)$.

Algorithm 4: Diagonal-Scaling-based State-feedback Control System with off-line PI

Input: Suitable initial feedback gain matrix $\widehat{K}_v^j$;
Weight matrices $Q, \hat{R} \in \mathbb{R}$;
Constant value $\mu, \delta \in (0,1)$;
Large enough scaling factor α_0;
Tolerance ϵ;
Count of $\rho(\hat{A}_{\alpha_j}) \ge 1$ break_count;
Internal model (F, G) of matrix S;

Initialize $\widehat{K}_v^0 = 0$, $\varepsilon_0 = 1$, $j = 0$, $\Lambda_0 = \mathrm{diag}(\alpha_0(1), \ldots, \alpha_0(n))$, and break_count = 0;
repeat
repeat
At the current Λ_j, collect a new data batch for (55) using (36) until (51) is satisfied;
Solve (56a) to find $L_{4\alpha_{j+1}}^{j+1}$ and $L_{5\alpha_{j+1}}^{j+1}$ in $\mathcal{H}_{\alpha_{j+1}}^{j+1}$;
Refresh the feedback gain matrix $\widehat{K}_v^{j+1}$ with (56b);
until $||\widehat{K}_v^{j+1} - \widehat{K}_v^j|| \le \varepsilon$;
Evaluate $\rho(\hat{A}_{\alpha_j})$ from (57);
if $\rho(\hat{A}_{\alpha_j}) \ge 1$ then
if break_count $\ge \tau$ then terminate with failure; end if
Replace the rejected scaling matrix by $\Lambda_j \leftarrow \Lambda_{j-1} + 0.5(\Lambda_j - \Lambda_{j-1})$;
Set $\Delta\alpha_j = 0$, break_count $\leftarrow$ break_count + 1, and repeat at the same j.
else
Set break_count = 0;
if $\|\Lambda_j - I_n\| \le \varepsilon$ then return $\widehat{K}_0^* = \widehat{K}_v^{j+1}$; end if
Update Λ_{j+1} with Eqs. (33) and (24), set $\Lambda_j \leftarrow \Lambda_{j+1}$, and set $j \leftarrow j + 1$;
end if
until $\|\Lambda_j - I_n\| \le \varepsilon$;
return $\widehat{K}_0^* = \widehat{K}_v^{j+1}$.

From Eqs. (55) and (56), the feedback gain matrix $\widehat{K}_v^j$ in the virtual system can be iteratively updated using collected data and the least-squares method. To clearly describe this iterative procedure and specify the updating mechanism of the diagonal scaling factor, the above derivations are organized into a complete offline PI algorithm, as shown in Algorithm 4.

In model-based DT systems, the spectral radius of $\hat{A} = \underline{A} - \hat{B}\widehat{K}$ can be directly evaluated. However, for data-driven case, the system matrices $\underline{A}$ and $\hat{B}$ are unknown, rendering direct computation infeasible. To circumvent this difficulty, we express the spectral radius of $\hat{A}_{\alpha_j}$ in an alternative form. Starting from an appropriate Lyapunov-like relation [42], one has:

$$\rho(\hat{A}_{\alpha_j}) = 1 - \frac{\lambda_{\min}(\hat{Q} + (\widehat{K}^j)^T \hat{R} \widehat{K}^j)}{\lambda_{\max}(P^j)} \quad (57)$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalues of the corresponding matrices, respectively. This characterization enables indirect assessment of the closed-loop stability without explicit knowledge of $\underline{A}$ and $\hat{B}$. The complete procedure for the data-driven state-feedback virtual control system is presented in Algorithm 4.

5. Numerical Study

In this section, a linear mass-spring-damper system is employed as a testbed, representing a canonical dynamic subsystem relevant to both engineering and network science. This system can be interpreted as a single node within a network of interconnected agents, with the mass, spring, and damper parameters representing inertia, inter-node coupling, and damping effects, respectively. Such an abstraction allows the numerical example to capture challenges typical in robust tracking and adaptive coordination among dynamically coupled subsystems. To evaluate the effectiveness of the proposed output tracking control scheme, which integrates an α-scaling mechanism with a PI framework, a numerical study is carried out under both model-based and data-driven implementations. The study centers on two primary aspects: (i) whether the closed-loop spectral radius $\rho(\hat{A}_{\alpha_j})$ remains strictly inside the Schur stability region (i.e., the unit circle) throughout the α-iteration process as α_j is driven toward 1; and (ii) the tracking performance of the plant output y with respect to the reference trajectory y_d, along with the convergence behavior of the tracking error $y_e = y - y_d$.

5.1. System setup

The mass-spring-damper system represents a class of mechanical subsystems commonly found in industrial equipment (e.g., positioning stages, suspension systems). Reliable tracking control in such systems directly affects product quality, energy efficiency, and maintenance intervals. Accordingly, the state-space representation given in Eqs. (1) and (2) is instantiated as:

$$x(k+1) = \begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{C_d}{M} \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} u(k) \quad (58)$$

$$y(k) = [1 \quad 0]x(k) \quad (59)$$

where $M = 1$ kg denotes the mass, $K_s = 1$ N/m the spring constant, and $C_d = 0.1215$ N s/m the damping coefficient. The weighting matrices in the performance index Eqs. (18) are selected as $Q = 10$ and $\hat{R} = 1$. The reference trajectory $y_s(k)$ is generated by the autonomous exosystem

$$x_s(k+1) = -x_s(k) \quad (60)$$

$$y_s(k) = x_s(k) \quad (61)$$

with initial condition $y_s(0) = 5$. The internal model parameters F, G, and T appearing in Eqs. (6a) and (6b) are set to -1, 1, and 1, respectively. The initial feedback gain matrix is chosen as $\hat{K}_{\text{init}} = [0,0,0]$, which completes the configuration of the simulation environment.

5.2. Results

This subsection presents the results in Figs. 8–11, including the tracking performance and scaling evolution obtained for both the model-based and data-driven control architectures.

The tracking performance of the model-based scheme after applying the learned control with diagonal scaling is illustrated in Fig. 8. Both the output y and reference output y_d remain bounded over the entire simulation horizon, indicating that the closed-loop system is stable at the origin. The evolution of the scaling factor α and the closed-loop spectral radius $\rho(\hat{A}_{\alpha_j})$ during the α-iteration under the model-based implementation is shown in Fig. 9. The blue curve shows that α decreases from its initial value to approximately 1 within about 20 iterations, driven by the proposed crossing-triggered reset mechanism and sign-based direction encoding. The orange curve represents $\rho(\hat{A}_{\alpha_j})$, which consistently remains below the Schur stability boundary. The red dashed line marks the theoretical baseline $\rho(\hat{A}_{\alpha_j}) = \alpha = 1$, serving as a reference for the desired stability condition. Notably, $\rho(\hat{A}_{\alpha_j})$ follows a smooth, bounded trajectory free of oscillatory overshoot, confirming that the auxiliary damping parameter μ_v and the adaptive update rule effectively suppress repeated crossings in the vicinity of $\alpha = 1$. The steady descent of α and the well-regulated spectral radius jointly demonstrate that the proposed iterative scaling mechanism reliably produces a well-conditioned initial stabilizing feedback gain matrix for the subsequent PI phase. The closed-loop spectral radius remains strictly below unity throughout the iteration, confirming that the system never experiences destabilizing transients. This property is essential for preventing mechanical stress accumulation and ensuring predictable wear, thereby facilitating condition-based maintenance.

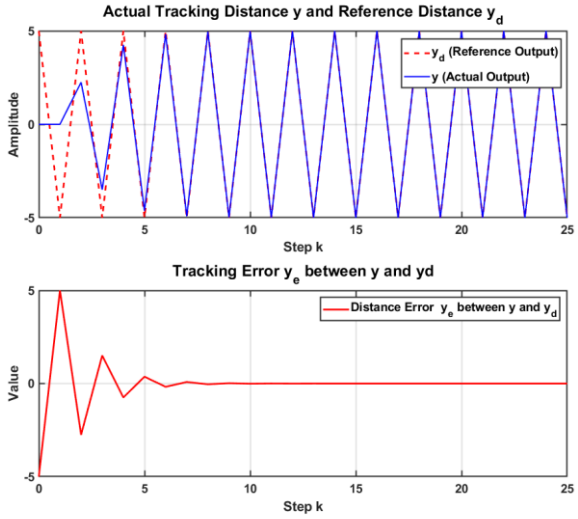

Figure 8. Model-based control: output y, reference y_d, and tracking error y_e.

The tracking performance of the data-driven control scheme after PI has converged is presented in Fig. 10. The upper subplot displays the reference output y_d (red dashed line) and the actual plant output y (blue solid line) over 100 time steps. The reference trajectory is a wave alternating between $+5$ and -5, generated by the autonomous exosystem in Eq. (60). The actual output y rapidly converges to y_d within the first few time steps and maintains accurate tracking thereafter, exhibiting no discernible steady-state error or phase lag. The lower subplot shows the tracking error $y_e = y - y_d$ (red solid line), which converges to zero and remains within a negligible band. These results confirm that the data-driven controller, designed without explicit model knowledge and relying solely on measured input-

output data, achieves asymptotic tracking of the reference signal while preserving robustness. The transient response further highlights the effectiveness of the α-scaling mechanism in furnishing a well-initialized feedback gain matrix for the subsequent iteration.

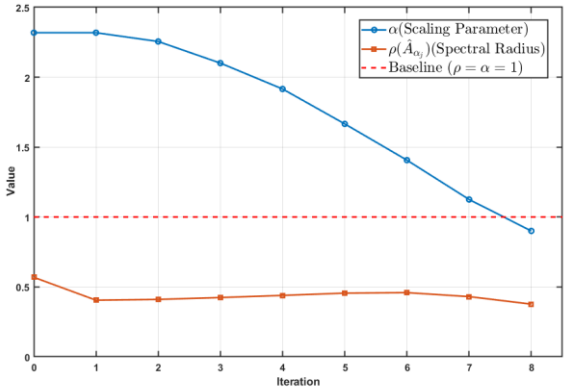

Figure 9. Model-based control: evolution of the scaling factor α and the closed-loop spectral radius $\rho(\hat{A}_{\alpha_j})$.

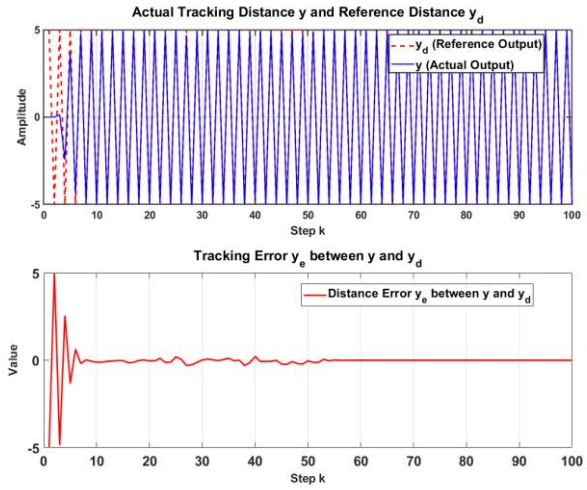

Figure 10. Data-driven control: output y, reference y_d, and tracking error y_e.

The evolution of α and $\rho(\hat{A}_{\alpha_j})$ during the α-iteration under the data-driven implementation is illustrated in Fig. 11. The blue circles represent α, which decreases monotonically and smoothly from its initial value to approximately 1 within about 30 iterations, confirming the effectiveness of the crossing-triggered reset mechanism and sign-based direction encoding even in the absence of an explicit plant model. The red squares denote $\rho(\hat{A}_{\alpha_j})$, which remains strictly below the Schur stability boundary throughout the entire iterative process. The black dashed line indicates the baseline condition $\rho(\hat{A}_{\alpha_j}) = \alpha = 1$ for reference. The inset provides a magnified view of the convergence phase (iterations 27 to 28), where both α and $\rho(\hat{A}_{\alpha_j})$ stabilize near their nominal values without exhibiting repeated oscillations or overshoot. These results demonstrate that the proposed α-scaling strategy, when executed in a data-driven fashion, reliably produces a well-conditioned initial stabilizing feedback gain for the subsequent PI phase.

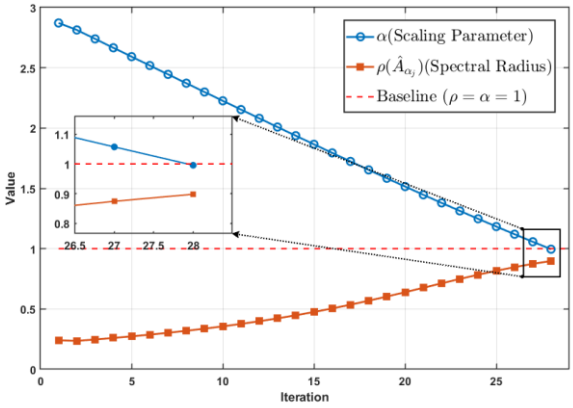

Figure 11. Data-driven control: evolution of the scaling factor α and the closed-loop spectral radius $\rho(\hat{A}_{\alpha_j})$.

6. Conclusion

This paper addresses robust tracking for linear DT systems via RL-based control with a diagonal scaling strategy. The contribution is the generalization of conventional scalar convergence-rate scaling to a diagonal matrix parametrization. This enriches the eigenvalue assignment space and enables fine-grained spectral tuning without a nominal model. The resulting data-driven controller guarantees asymptotic tracking while preserving robustness against unmodeled dynamics without model knowledge or system identification. Simulations under both model-based and data-driven settings confirm that the diagonal-scaling mechanism reliably drives initial gains (even destabilizing) to a well-conditioned stabilizing solution, with the closed-loop spectral radius strictly inside the unit circle throughout adaptation. Tracking error vanishes upon PI convergence. Future work will explore extensions to fault-tolerant control and integration with predictive maintenance frameworks, further strengthening the link between control design and system dependability.

References

1. Zhou X, Wan Z. Path planning and motion control of robotic arm based on neural network. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2025; 27(4): 205794. https://doi.org/10.17531/ein/205794.
2. Kocer Ozturk M, Khaniyev T. Optimal maintenance policy for a Markov deteriorating system under reliability limit. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2024; 26(4): 190865. https://doi.org/10.17531/ein/190865.
3. Josephin Shermila P, Anu Disney D, Reeda Lenus C, Niruban R. Efficiency and reliability: Optimization of energy management in electric vehicles apply monarch butterfly algorithm and fuzzy logic control. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2025; 27(3): 200691. https://doi.org/10.17531/ein/200691.
4. Chai TY. Operational optimization and feedback control for complex industrial processes. *Acta Automatica Sinica* 2013; 39(11): 1744-1757. https://doi.org/10.3724/SP.J.1004.2013.01744.
5. Siciliano B, Khatib O (eds). *Springer Handbook of Robotics*, 2nd ed. Cham: Springer International Publishing; 2016.
6. Francis BA, Wonham WM. The internal model principle of control theory. *Automatica* 1976; 12(5): 457-465. https://doi.org/10.1016/0005-1098(76)90006-6.
7. Huang J. *Nonlinear Output Regulation: Theory and Applications*. Philadelphia: Society for Industrial and Applied Mathematics; 2004. https://doi.org/10.1137/1.9780898718683.
8. Isidori A, Byrnes CI. Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control* 1990; 35(2): 131-140. https://doi.org/10.1109/9.45168.
9. Garcia de Marina H, Cao M, Jayawardhana B. Controlling rigid formations of mobile agents under inconsistent measurements. *IEEE Transactions on Robotics* 2015; 31(1): 31-39. https://doi.org/10.1109/TRO.2014.2373145.
10. Zhang C, Lin W, Ke D, Sun Y. Smoothing tie-line power fluctuations for industrial microgrids by demand side control: An output regulation approach. *IEEE Transactions on Power Systems* 2019; 34(5): 3716-3728. https://doi.org/10.1109/TPWRS.2019.2906252.
11. Zhang J, Yang D, Zhang H, Su H. Adaptive secure practical fault-tolerant output regulation of multiagent systems with DoS attacks by asynchronous communications. *IEEE Transactions on Network Science and Engineering* 2023; 10(6): 4046-4055. https://doi.org/10.1109/TNSE.2023.3281899.
12. Li L, Rong D, Fu J, Guo Q. Multi-level event-triggered scheme of output regulation under long-duration DoS attacks in switched systems with dissipativity. *IEEE Transactions on Network Science and Engineering* 2024; 11(2): 1631-1641. https://doi.org/10.1109/TNSE.2023.3327591.
13. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press; 2018.
14. Lewis FL, Vrabie DL, Syrmos VL. *Optimal Control*, 3rd ed. Hoboken, NJ: John Wiley & Sons; 2012.
15. Lewis FL, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine* 2009; 9(3): 32-50. https://doi.org/10.1109/MCAS.2009.933854.
16. Zhang H, Liu D, Luo Y, Wang D. Adaptive Dynamic Programming for Control: Algorithms and Stability. London: Springer; 2013.
17. Jiang Y, Jiang ZP. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* 2012; 48(10): 2699-2704. https://doi.org/10.1016/j.automatica.2012.06.096.
18. Vrabie D, Vamvoudakis KG, Lewis FL. *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. London: Institution of Engineering and Technology; 2012.
19. Chen C, Xie L, Xie K, Lewis FL, Liu Y, Xie S. Learning the continuous-time optimal decision law from discrete-time rewards. *National Science Open* 2024; 3(5): 20230054. https://doi.org/10.1360/nso/20230054.
20. Kiumarsi B, Lewis FL. Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems* 2015; 26(1): 140-151. https://doi.org/10.1109/TNNLS.2014.2358227.
21. Kiumarsi B, Lewis FL, Modares H, Karimpour A, Naghibi-Sistani MB. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica* 2014; 50(4): 1167-1175. https://doi.org/10.1016/j.automatica.2014.02.015.
22. Lewis FL, Vamvoudakis KG. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 2011; 41(1): 14-25. https://doi.org/10.1109/TSMCB.2010.2043839.

23. Kiumarsi B, Lewis FL, Naghibi-Sistani MB, Karimpour A. Optimal tracking control of unknown discrete-time linear systems using input-output measured data. *IEEE Transactions on Cybernetics* 2015; 45(12): 2770-2779. https://doi.org/10.1109/TCYB.2014.2384016.
24. Chen C, Xie L, Xie K, Lewis FL, Xie S. Adaptive optimal output tracking of continuous-time systems via output-feedback-based reinforcement learning. *Automatica* 2022; 146: 110581. https://doi.org/10.1016/j.automatica.2022.110581.
25. Jiang Y, Fan J, Gao W, Chai T, Lewis FL. Cooperative adaptive optimal output regulation of nonlinear discrete-time multi-agent systems. *Automatica* 2020; 121: 109149. https://doi.org/10.1016/j.automatica.2020.109149.
26. Huang C, Chen C, Xie K, Li Z, Xie S. Adaptive output synchronization with designated convergence rate of multiagent systems based on off-policy reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2024; 54(8): 4667-4678. https://doi.org/10.1109/TSMC.2024.3387395.
27. Gao W, Jiang ZP. Adaptive dynamic programming and adaptive optimal output regulation of linear systems. *IEEE Transactions on Automatic Control* 2016; 61(12): 4164-4169. https://doi.org/10.1109/TAC.2016.2548662.
28. Jiang Y, Kiumarsi B, Fan J, Chai T, Li J, Lewis FL. Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning. *IEEE Transactions on Cybernetics* 2020; 50(7): 3147-3156. https://doi.org/10.1109/TCYB.2018.2890046.
29. Bin M, Marconi L, Teel AR. Adaptive output regulation for linear systems via discrete-time identifiers. *Automatica* 2019; 105: 422-432. https://doi.org/10.1016/j.automatica.2019.04.019.
30. Zhao F, Gao W, Liu T, Jiang ZP. Adaptive optimal output regulation of linear discrete-time systems based on event-triggered output-feedback. *Automatica* 2022; 137: 110103. https://doi.org/10.1016/j.automatica.2021.110103.
31. Wang Z, Wang Y, Kowalczuk Z. Adaptive optimal discrete-time output-feedback using an internal model principle and adaptive dynamic programming. *IEEE/CAA Journal of Automatica Sinica* 2024; 11(1): 131-140. https://doi.org/10.1109/JAS.2023.123759.
32. Xia C, Dong Y, Wang C, Xu S. Data-driven output regulation control for constrained linear systems. *Science China Information Sciences* 2025; 68(3): 132209. https://doi.org/10.1007/s11432-024-4208-3.
33. Lin L, Huang J. Data-driven optimal output regulation for continuous-time linear systems via internal model principle. *IEEE Transactions on Automatic Control* 2025; 70(6): 4202-4208. https://doi.org/10.1109/TAC.2025.3535281.
34. Jiang Y, Chai T, Chen G. Output feedback-based adaptive optimal output regulation for continuous-time strict-feedback nonlinear systems. *IEEE Transactions on Automatic Control* 2025; 70(2): 767-782. https://doi.org/10.1109/TAC.2024.3441668.
35. Liu Z, Li C. Optimal adaptive output regulation of discrete-time nonlinear stochastic systems. *SIAM Journal on Control and Optimization* 2025; 63(4): 2369-2396. https://doi.org/10.1137/24M1679197.
36. Liu T, Huang J. Adaptive cooperative output regulation of discrete-time linear multi-agent systems by a distributed feedback control law. *IEEE Transactions on Automatic Control* 2018; 63(12): 4383-4390. https://doi.org/10.1109/TAC.2018.2823266.
37. Li S, Wang F, Er MJ, Yang Z. Approximate output regulation of discrete-time stochastic multiagent systems subject to heterogeneous and unknown dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2022; 52(10): 6373-6382. https://doi.org/10.1109/TSMC.2022.3145354.
38. Chen C, Xie L, Jiang Y, Xie K, Xie S. Robust output regulation and reinforcement learning-based output tracking design for unknown linear discrete-time systems. *IEEE Transactions on Automatic Control* 2023; 68(4): 2391-2398. https://doi.org/10.1109/TAC.2022.3172590.
39. Liang D, Dong Y, Wang C, Zhai G. Data-driven cooperative output regulation of linear discrete-time multiagent systems with unknown dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2024; 54(8): 5025-5034. https://doi.org/10.1109/TSMC.2024.3390388.
40. Chen C, Xie LH. A data-driven prescribed convergence rate design for robust tracking of discrete-time systems. *Journal of Guangdong University of Technology* 2021; 38(6): 29-34. https://doi.org/10.12052/gdutxb.210105.
41. Jiang Y, Fan JL, Chai TY. Data-driven optimal output regulation with assured convergence rate. *Acta Automatica Sinica* 2022; 48(4): 980-991. https://doi.org/10.16383/j.aas.c200932.
42. Chen C, Lewis FL, Li B. Homotopic policy iteration-based learning design for unknown linear continuous-time systems. *Automatica* 2022; 138: 110153. https://doi.org/10.1016/j.automatica.2021.110153.