



Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.sciencedirect.com/journal/european-journal-of-control

A hybrid algorithm for iterative adaptation of feedforward controllers: An application on electromechanical switches

Eloy Serrano-Seco^{ID}*, Eduardo Moya-Lasheras^{ID}, Edgar Ramirez-Laboreo^{ID}

Departamento de Informatica e Ingenieria de Sistemas (DIIS) and Instituto de Investigacion en Ingenieria de Aragon (I3A), Universidad de Zaragoza, Zaragoza, 50018, Spain

ARTICLE INFO

Recommended by T. Parisini

Keywords:

Mechatronics
Adaptive control
Iterative methods
Run-to-run control
Optimization algorithms
Adaptive coordinate descent

ABSTRACT

Electromechanical switching devices such as relays, solenoid valves, and contactors offer several technical and economic advantages that make them widely used in industry. However, uncontrolled operations result in undesirable impact-related phenomena at the end of the stroke. As a solution, different soft-landing controls have been proposed. Among them, feedforward control with iterative techniques that adapt its parameters is a solution when real-time feedback is not available. However, these techniques typically require a large number of operations to converge or are computationally intensive, which limits a real implementation. In this paper, we present a new algorithm for the iterative adaptation that is able to eventually adapt the search coordinate system and to reduce the search dimensional size in order to accelerate convergence. Moreover, it automatically toggles between a derivative-free and a gradient-based method to balance exploration and exploitation. To demonstrate the high potential of the proposal, each novel part of the algorithm is compared with a state-of-the-art approach via simulation.

1. Introduction

Today, solenoid valves (Angadi & Jackson, 2022) and electromechanical relays (Gergić & Hercog, 2019) are used in virtually all industries, ranging from household appliances and automotive applications to robotics and medical devices. In general, the basic operating principle of all electromechanical switching devices is similar: when electrical energy is applied, a magnetic force accelerates a moving component to the end of the stroke. This causes undesirable phenomena, including bouncing and violent impacts, which result in premature device wear and acoustic noise. In an effort to solve or reduce these phenomena, several control strategies have been proposed, generally with the same objective: to reach the final position with zero velocity. Among these strategies are those based on backstepping control (Al Saaideh et al., 2022), sliding-mode control (Deschaux et al., 2018), extremum-seeking adaptive control (Benosman & Atunç, 2015), or iterative learning control (Moya-Lasheras & Sagues, 2024).

Like other authors (Braun et al., 2018), some of our previous works employ a feedforward controller for two main reasons. First, a dynamic property of these systems, differential flatness, allows us to easily design the controller by model inversion. Secondly, feedforward control provides immediate responses to reference changes and is able to compensate for known disturbances. Despite its advantages, it alone is not robust to design errors, modeling errors, or system changes.

To address these limitations, various complementary strategies exist, including conventional feedback controllers with observers (Schroedter et al., 2018), learning algorithms (Grotjahn & Heimann, 2002), and parameter adjustments based on measurable variables (Yeh & Hsu, 1999). Nevertheless, all the previously mentioned controllers are dependent on feedback of the variable to be controlled. However, in some cases, these variables cannot be measured or observed due to economic or technical constraints.

Therefore, we explored an alternative (Moya-Lasheras et al., 2023) based on run-to-run controllers. The main idea is to iteratively update the feedforward controller parameters from measurable variables that, even though they cannot be directly used to observe and control the variable of interest, they provide a performance index for each iteration (i.e., switching operation). In a first approach, the iterative adaptation law was implemented using a Pattern Search (Lewis & Torczon, 2000) algorithm. Although the method is computationally light and accurate, it requires too many evaluations to converge. Specifically, it needs $2q + 1$ evaluations (where q is the search space dimension) to determine whether to move to a new point. Our latest work Ramirez-Laboreo et al. (2024) demonstrated that convergence speed can be improved through sensitivity-based parameter reduction. However, it did not offer an automated approach for determining the number of parameters to be reduced for a given problem, among other limitations.

* Corresponding author.

E-mail address: eserranoseco@unizar.es (E. Serrano-Seco).

<https://doi.org/10.1016/j.ejcon.2025.101305>

Received 6 June 2025; Accepted 9 July 2025

Available online 23 July 2025

0947-3580/© 2025 The Authors. Published by Elsevier Ltd on behalf of European Control Association. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

In this line, [Loshchilov et al. \(2011\)](#) presents an Adaptive Coordinate Descent algorithm. The strategy involves periodically updating the coordinate system by a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and Adaptive Encoding to decompose the problem into as many one-dimensional problems as dimensions in the general problem. Although it is an interesting idea, as concluded by the authors in [Hansen and Ostermeier \(2001\)](#), the function evaluations needed are about $10q$, $30q$ for a real-world search problem, and $100q^2$ for complete adaptation. Given that each evaluation involves a switching operation, the number of switching operations with unsatisfactory performance would be excessively high.

In terms of one-dimensional search, the authors of [Loshchilov et al. \(2011\)](#) suggest derivative-free methods or the use of gradients. Gradient methods are a powerful tool, especially if the objective function is known. A similar alternative are subgradient methods, since they can work with approximations based on the value of the function to be optimized across the search space. Another option are the sign gradient descent methods, first introduced in the RProp (Resilient Propagation) algorithm ([Moulay et al., 2019](#)). Despite being technically a gradient-based method, the RProp algorithm has a low computational load, as it only needs to calculate the sign of the gradient, not the gradient itself. Nevertheless, adjusting the hyperparameters of these algorithms can be a challenging task. In contrast, some gradient descent methods implement an adaptive step size without the need for hyperparameters.

This paper presents a new Run-to-Run controller based on an Adaptive Coordinates algorithm (R2R-AC) in order to automate the improvement process described in [Ramirez-Laboreo et al. \(2024\)](#) and to enhance the performance of the iterative adaptation law. This new algorithm leverages the controller sensitivity with respect to its parameters to calculate an alternative search basis that decomposes the initial q -dimensional problem into q one-dimensional problems to optimize on the descending coordinate with the highest improvement potential. We analyze and compare three versions of the algorithm that differ in how the step size is computed: one based on derivative-free methods, another based on gradient-based algorithms, and a hybrid one that toggles between the other two to enhance the exploration–exploitation tradeoff.

The paper is organized as follows. Section 2 provides a concise overview of the dynamic and control model that has prompted the development of the proposed algorithm. Section 3 develops the iterative adaptation law of the R2R-AC strategy and discusses the different versions previously mentioned. Section 4 contains simulation results that demonstrate the functionality of our proposals and the comparison with a state-of-the-art feedforward run-to-run controller. Finally, the conclusions are discussed in Section 5.

2. Background of the control system

In this section, we briefly describe the system dynamics and control where the need for the proposed new algorithm has arisen. For a more detailed explanation, readers are referred to our works ([Moya-Lasheras et al., 2023](#); [Ramirez-Laboreo et al., 2024](#)).

2.1. System dynamics

The system is modeled as a single-coil reluctance actuator, subject to two types of forces: passive elastic forces—typically modeled as ideal springs—and a magnetic force. The magnetic force is generated when current flows through the coil, causing an inner fixed core to become magnetized and attract the movable core. The typical method of supplying the actuator with power is by providing a voltage. We describe the dynamics of the system using a state-space model, where the voltage u is the input to our system, and the position z , velocity v

and magnetic flux linkage λ are the state variables. The state equations are defined as

$$\dot{z} = v, \quad (1)$$

$$\dot{v} = \frac{1}{m} \left(-k_s (z - z_s) - \frac{1}{2} \lambda^2 \frac{\partial \mathcal{R}}{\partial z} \right), \quad (2)$$

$$\dot{\lambda} = -R \lambda \mathcal{R}(z, \lambda) + u, \quad (3)$$

where m , k_s , z_s , R , and \mathcal{R} are the moving mass, the spring stiffness, the spring resting position, the coil resistance, and an auxiliary function based on the magnetic reluctance concept, respectively. This auxiliary function considers the magnetic saturation and flux fringing phenomena,

$$\mathcal{R}(z, \lambda) = \frac{\kappa_1}{1 - |\lambda|/\kappa_2} + \kappa_3 + \frac{\kappa_4 z}{1 + \kappa_5 z \log(\kappa_6/z)}, \quad (4)$$

where κ_1 , κ_2 , κ_3 , κ_4 , κ_5 , and κ_6 are positive constants. Overall, the system dynamics depends on $q = 9$ uncertain parameters, which can be grouped in the parameter vector p .

$$p = [k_s \ z_s \ m \ \kappa_1 \ \kappa_2 \ \kappa_3 \ \kappa_4 \ \kappa_5 \ \kappa_6]^\top. \quad (5)$$

Note that the resistance R is treated independently as a parameter without uncertainty, as it can be easily measured.

2.2. Control

The control structure used in this study is schematized in [Fig. 1](#). This is an iterative control design for a real-world scenario with two particularities:

- The variable to be controlled, the position z , cannot be fed back for several reasons. Firstly, a position sensor is more expensive than the switching devices. Secondly, a protective housing impedes access to the component whose position needs to be known. In addition, a real-time estimate is also unavailable.
- Errors in the model parameters are not negligible. These devices are produced at a low cost, with relaxed manufacturing tolerances, causing variability in the value of the parameters.

Due to the first point, the control strategy is focused on a feedforward controller. It is designed by model inversion, taking advantage of a structural property shared by such devices: differential flatness ([Lévine, 2011](#)). Considering that the objective is soft-landing control, as in our previous works, $z(t)$ is the desired trajectory, designed as a 5th-degree polynomial from z_0 , the initial mechanical limit of motion of the moving component to be controlled, to z_f , the final limit. The trajectory is defined over the time interval $[t_0, t_f]$, subject to boundary conditions of zero initial and final velocities and accelerations. In short, the feedforward control term, u_{ff} , is defined as a function of $z(t)$, its derivatives, and the parameter vector p . Although this applies to a specific case, it can be generalized as $u_{ff} = u_{ff}(t, \theta)$ for parametric controllers where θ represents any vector of control parameters. In our case θ is a normalized version of p .

Due to the second point, including a feedback loop to adapt the control parameters θ is essential. Some previous works propose associating a measurement related to the control objective to a cost value J . In a real-world application, due to measurement difficulties, J may be computed based on indirect measurements associated with the impacts ([Moya-Lasheras et al., 2023](#); [Winkel et al., 2023](#)). In simulation, however, the impact velocity v_c could be directly used as feedback,

$$J = |v_c|. \quad (6)$$

At this point, it is reasonable to assume that the process which relates θ to J is unknown or difficult to work with analytically. In response to this, the proposal is to use a black-box optimizer as the iterative adaptation law to minimize J along switchings. Since we are focusing on a real-world application, the optimizer must be not only accurate,

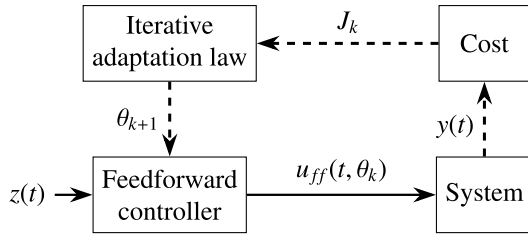


Fig. 1. General control diagram. The subscript k denotes the variables of the k th evaluation of the run-to-run adaptation law. The feedforward block computes u_{ff} from the parameter vector θ and the desired reference signal r . The adaptation law updates the feedforward parameters θ using the cost J , which is derived from the measurable output y .

but also computationally light and able to quickly discard poorly performing evaluations to minimize unsatisfactory switching operations. Therefore, to speed up convergence, a state-of-the-art feedforward run-to-run controller based on a Pattern Search algorithm (Ramirez-Laboreo et al., 2024) shows a strategy to reduce the number of search dimension, without relying on further cost function evaluations. Assuming that larger changes in the control action translate into larger changes in the cost value, this strategy is based on a local sensitivity analysis of smooth (differentiable) controllers. The Fisher matrix, $F(\theta)$, which can be computed from the sensitivity of the controller with respect to θ ,

$$F(\theta) = \int_{t_0}^{t_f} \left[\left(\frac{\partial u_{ff}(t, \theta)}{\partial \theta} \right)^T \left(\frac{\partial u_{ff}(t, \theta)}{\partial \theta} \right) \right] d\tau, \quad (7)$$

is evaluated at a nominal point θ^{nom} . Since $F(\theta^{\text{nom}})$ is symmetric and positive semidefinite, the eigendecomposition and the singular value decomposition coincide. That is, the Fisher matrix can be expressed as $F(\theta^{\text{nom}}) = V \Lambda V^T$, where $V \in \mathbb{R}^{q \times q}$ is the orthonormal matrix with the eigenvectors (or singular vectors) of $F(\theta^{\text{nom}})$ as columns and $\Lambda \in \mathbb{R}^{q \times q}$ is the diagonal matrix of the corresponding eigenvalues (or singular values).

Finally, a transformation of the known vector θ to a new vector $X \in \mathbb{R}^q$ is defined in terms of the basis change matrix V as

$$\theta = \theta^{\text{nom}} + V(X - X^{\text{nom}}) \iff X = X^{\text{nom}} + V^T(\theta - \theta^{\text{nom}}), \quad (8)$$

where X^{nom} is the nominal value of X , which can be chosen arbitrarily.

Thanks to this procedure, the controller is parametrized in such a way that the correlation between the sensitivities with respect to the new parameters in vector X is low. In addition, Λ provides information about these sensitivities, enabling the exclusion of coordinates with negligible sensitivity from the optimization process.

3. New algorithm

This section is divided into two parts. The first one, following the idea presented in Loshchilov et al. (2011), introduces a new algorithm which solves the open questions posed in Ramirez-Laboreo et al. (2024), i.e., what is the appropriate number of dimensions to optimize in each situation and, since the analysis is local, how often the alternative coordinate system should be recalculated. The second part presents a classical derivative-free method, a gradient-based method, and a hybrid methodology that enables toggling between the two options to calculate the step size of the previous algorithm.

3.1. Iterative adaptation law of the R2R-AC

The behavior of the algorithm is presented in Algorithm 1. The new algorithm must fulfill two requirements: it must eventually upgrade the alternative coordinate system and it must select automatically the number of search dimensions. We propose to convert the complete optimization of the θ -problem into successive X -optimization problems.

Each X -optimization problem optimizes X in a new canonical basis initialized at $X^{\text{nom}} = \mathbf{0}_q$. This decision simplifies the transformation of X into θ as

$$\theta = \theta^{\text{nom}} + VX, \quad (9)$$

with the only eventual needs to update θ^{nom} as the best evaluated θ (Algorithm 1, line 18) and update the transformation matrix V (line 2) as shown in the previous section (according to Ramirez-Laboreo et al. (2024)). From now on, for clarity, since we work mainly in the X -optimization space, with a slight abuse of notation, we denote the cost — obtained for a value θ that depends on X , (9) — as a direct function of X : $J(X) = J$.

Once the strategy for updating the alternative coordinate system has been selected, the remaining tasks are to determine the timing of the update event and the search problem dimensions. The proposed solution is to first perform an exploration process to find the coordinate with the greatest descent and then to exploit this coordinate. For the exploration process, V should be ordered such that the associated eigenvalues are arranged from largest to smallest, i.e.,

$$\Lambda_{(1,1)} \geq \Lambda_{(2,2)} \geq \dots \geq \Lambda_{(q,q)}, \quad (10)$$

where the subscripts in parentheses represent the position in the matrix. This step permits the ordering of the coordinates from the highest to the lowest sensitivity of the control action to the parameters X . The proposal for selecting the coordinate of greatest descent is based on pattern search methods: the origin of the coordinate system (line 13) and two points along each coordinate (line 7), X^+ and X^- , symmetrically located at a distance δ from the origin coordinate, are evaluated,

$$X^+ \leftarrow X^b + \delta \cdot e_d, \quad (11)$$

$$X^- \leftarrow X^b - \delta \cdot e_d, \quad (12)$$

where X^b is the point associated with the lowest cost value, whose value is $\mathbf{0}_q$ at the start of each X -optimization problem, $\delta \in \mathbb{R}$ is the step size, and $e_d \in \mathbb{R}^q$ is the unit vector that defines the direction of the $d \in [1, q]$ coordinate of the canonical basis. This pattern is evaluated sequentially coordinate by coordinate (lines 4–10) until a cost-improving coordinate has been found. At this moment the exploration process ends and the exploitation process begins by a line-search (lines 11–16). The algorithm continues to look for lower cost points along the corresponding direction and orientation by a method that embraces the philosophy of sign gradient descent algorithms,

$$X^{\text{next}} \leftarrow X^b + \text{sgn} \left(X_{(d)}^b \right) \delta \cdot e_d, \quad (13)$$

where X^{next} is the next point to be evaluated, $\text{sgn}(\cdot)$ is the sign operator, and $X_{(d)}^b$ is the d -th component of X^b . Note that (13) only applies after a better point has been found, so in this case, $X_{(d)}^b \neq 0$.

When the evaluation of X^{next} does not improve the cost, the present X -optimization problem ends, θ^{nom} and V are updated and the X -optimization problem is reset (line 18). Thus, it is not necessary to complete the pattern to shift it, and the number of dimensions of the problem is automatically reduced to the minimum allowed to obtain improvements. An exception applies to this sequence, when the algorithm has moved on the first coordinate (line 17), θ is not updated, so neither is V , and the evaluation of the second coordinate continues around the best point found on the first coordinate. The main reason is not to transform the algorithm into a single gradient search method in which the descending coordinate is calculated through the coordinate that further modifies u_{ff} , because the convergence speed may decrease due to limited information and reduced opportunities to directly identify a new best point.

To complete the algorithm, it only remains to define how to upgrade the step size (lines 9 and 15). This is discussed in the following subsection.

Algorithm 1 Iterative adaptation law of the R2R-AC

Initialize: δ , θ^{nom} , $d \leftarrow 0$, $X^b \leftarrow \mathbf{0}_q$, $J(X^{\text{next}}) \leftarrow \infty$

1: **while** true **do**

\triangleright Alternative coordinate system

2: $V \leftarrow$ sorted eigenvectors of $\mathcal{F}(\theta^{\text{nom}})$

\triangleright Best descent coordinate exploration

3: Evaluate $J(X^b)$

4: **while** $J(X^{\text{next}}) > J(X^b)$ **do**

5: $d \leftarrow (d \bmod q) + 1$ \triangleright Next d

6: $X^+ \leftarrow X^b + \delta \cdot e_d$; $X^- \leftarrow X^b - \delta \cdot e_d$

7: Evaluate $J(X^+)$ and $J(X^-)$

8: $X^{\text{next}} \leftarrow \arg \min_{X \in \{X^+, X^-\}} J(X)$

9: Update δ \triangleright Algorithm 2

10: **end while**

\triangleright Descent coordinate exploitation: Line-search

11: **while** $J(X^{\text{next}}) < J(X^b)$ **do**

12: $X^b \leftarrow X^{\text{next}}$

13: $X^{\text{next}} \leftarrow X^b + \text{sgn}\left(X_{(d)}^b\right) \delta \cdot e_d$,

14: Evaluate $J(X^{\text{next}})$

15: Update δ \triangleright Algorithm 2

16: **end while**

\triangleright Reset the X optimization problem

17: **if** $d \neq 1$ **then**

18: $\theta^{\text{nom}} \leftarrow \theta^{\text{nom}} + V X^b$; $d \leftarrow 0$; $X^b \leftarrow \mathbf{0}_q$

19: **end if**

20: **end while**

3.2. Step-size update: hybrid method

The new algorithm enables the q -dimensional problem to be reduced to the behavior of a one-dimensional problem. For the sake of simplicity, this subsection assumes that a descending shift always occurs. For clarity, we work with \hat{e}_d , the unit vector of the desired direction and orientation.

The fact that there is no analytical information on the relationship between the cost value J and the parameters to optimize X , together with the need of a computationally light and fast convergence process, has led us to the use of direct search methods based on updating a step size δ . We explore two approaches: derivative-free methods and adaptive methods based on the objective function.

In the basic derivative method, the step size decreases as the number of evaluations increases; however, other methods allow for expanding the step size if certain conditions are met. The latter methods are used in algorithms as Pattern Search and RProp, among others. When it appears the process is progressing in the right direction, i.e., the cost improves, the step size should be increased (expansion) to reach the possibly distant optimum point more quickly. Conversely, when the process has reached a minimum, i.e., the cost does not improve, the step size should be decreased (contraction) to allow for an approach to the minimum cost and reduce fluctuations. In short, the next point to evaluate X^{next} can be calculated as

$$X^{\text{next}} = X^b + \delta^{\text{DF}} \cdot \hat{e}_d, \quad (14)$$

where X^b is the point considered best, δ^{DF} is the derivative-free step size, which is eventually updated as

$$\delta^{\text{DF}} \leftarrow \begin{cases} \max(\alpha_{\text{con}} \delta^{\text{DF}}, \delta_{\text{min}}^{\text{DF}}) & \text{if } J(X^{\text{next}}) > J(X^b) \\ \min(\alpha_{\text{exp}} \delta^{\text{DF}}, \delta_{\text{max}}^{\text{DF}}) & \text{if } J(X^{\text{next}}) \leq J(X^b) \end{cases}, \quad (15)$$

where α_{con} and α_{exp} are the contraction and expansion constants such that $0 < \alpha_{\text{con}} < 1 < \alpha_{\text{exp}}$, and $\delta_{\text{min}}^{\text{DF}}$ and $\delta_{\text{max}}^{\text{DF}}$ are the minimum and maximum allowed step sizes, respectively. Unfortunately, the tuning of α values suffers from a trade-off between faster convergence and the

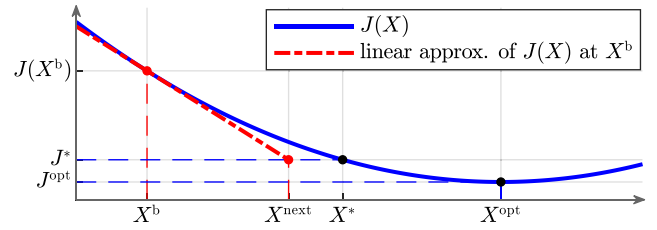


Fig. 2. Geometric interpretation of the first-order method adopted.

likelihood of reaching a local minima, which tends to be higher as the dimensionality of the problem increases.

In contrast, adaptive methods based on the objective function are able to dynamically fit the step size, often utilizing gradient information in first-order methods. A typical geometric interpretation of these methods is illustrated in Fig. 2 for an ideal situation with a convex objective function. The aim is to reach a point X^* with a lower cost J^* from a known point X^b , for our algorithm, the best evaluated point. To this end, the step size is approximated as the ratio of the cost difference $J(X^b) - J^*$ to the gradient g at X^b . To determine J^* , common approaches include treating it as a constant equal to J^{opt} , the cost of the optimal point X^{opt} , i.e., the minimum value of the objective function. Even if the real value is unknown, $J^* = 0$ is adequate on many situations, but it is usually a strong assumption. Alternatively, J^* can vary by iteration (Algorithm 2, line 5), which is often more practical, as large step sizes can be more detrimental than an advantage, leading to divergence. As shown in Fig. 2, larger step sizes introduce greater approximation errors, potentially affecting convergence.

For our problem, our non-static search coordinate system complicates the collection of information on g , as a reminder, the gradient with respect to X . Thus, we propose a reinterpretation of this method by working with \tilde{g} , an average slope of the trajectory followed by the algorithm that represents the average improvement capacity. In addition, given the possible non-smoothed decreasing cost, we work with \tilde{J} , an average value of the cost. These are computed as

$$\tilde{J} \leftarrow \beta \tilde{J} + (1 - \beta) J(X_k), \quad (16)$$

$$\tilde{g} \leftarrow \sqrt{\beta \tilde{g}^2 + (1 - \beta) \left(\frac{J(X_k) - J(X_{k-1})}{\|X_k - X_{k-1}\|} \right)^2}, \quad (17)$$

where $\beta < 1$ is a positive constant that acts as a decay factor, and the subscript k refers to the evaluation number. With these adjustments, the next point to be evaluated and the value of the gradient-based step size δ^{GB} are calculated, respectively, as

$$X^{\text{next}} = X^b + \delta^{\text{GB}} \cdot \hat{e}_d, \quad (18)$$

$$\delta^{\text{GB}} = \frac{(\tilde{J} - J^*)}{\tilde{g}}. \quad (19)$$

Analyzing the equation, an additional advantage of slope filtering is that it mitigates the problem of excessively oscillating step sizes.

Algorithm 2 Process to update the step size δ

Initialize: \tilde{J} , \tilde{g} , β , δ^{DF} , α_{con} , α_{exp} , $\delta_{\text{min}}^{\text{DF}}$, $\delta_{\text{max}}^{\text{DF}}$, J^*

1: **for** $k \leftarrow 1$ to num. evaluations **do**

2: Update \tilde{J} and \tilde{g} \triangleright Eq. (16) and Eq. (17)

3: **if** δ must be updated **then**

4: Update δ^{DF} and δ^{GB} \triangleright Eq. (15) and Eq. (19)

5: Update J^*

6: $\delta \leftarrow \begin{cases} \delta^{\text{DF}} & \text{if } \tilde{J} \leq J^* \\ \delta^{\text{GB}} & \text{if } \tilde{J} > J^* \end{cases}$

7: **end if**

8: **end for**

Table 1

Nominal parameter values.

k_s	55 N/m	κ_5	1320 m ⁻¹
z_s	0.015 m	κ_6	$9.73 \cdot 10^{-3}$ m
m	$1.6 \cdot 10^{-3}$ kg	R	50 Ω
κ_1	1.35 H ⁻¹	z_0	10^{-3} m
κ_2	0.0229 Wb	z_f	0
κ_3	3.88 H ⁻¹	t_0	0
κ_4	$7.67 \cdot 10^4$ H ⁻¹ /m	t_f	$3.5 \cdot 10^{-3}$ s

Table 2

Initial hyperparameters of the control strategies.

Control	δ^{DF}	$\delta_{\min}^{\text{DF}}$	$\delta_{\max}^{\text{DF}}$	α_{con}	α_{exp}	β
R2R-PS+	0.2	$2 \cdot 10^{-10}$	2	0.5	2	–
R2R-AC	0.2	$2 \cdot 10^{-10}$	2	0.7	1.1	0.8

This concept is also used in several stochastic gradient descent algorithms, including RMSProp (Ma et al., 2022). Despite the adaptability of gradient-based methods, they are highly susceptible to the geometry of the objective function and the initial evaluations.

Our hybrid version for calculating δ tries to take advantage of the good performances of both strategies and avoid their drawbacks by toggling between (15) and (19) based on whether \bar{J} is less than or greater than J^* , respectively (line 6). In this form the first resource of the algorithm is (15), but, when convergence slows significantly or the evaluated point is far from the optimal point, the strategy toggles to (19). The update of the step size δ is summarized in Algorithm 2.

4. Simulated results

In this section, to illustrate the benefits of the new algorithm, we analyze the improvements achieved over our previous work (Ramirez-Laboreo et al., 2024). To assess the improvements introduced by R2R-AC and the hybrid step size, the following control strategies are evaluated:

- **R2R-PS+**: the best solution in Ramirez-Laboreo et al. (2024); it uses for the iterative adaptation law a Pattern Search algorithm, with a single initial basis change (9) and four dimensions.
- **R2R-AC (DF)**: proposed R2R-AC in which $\delta \leftarrow \delta^{\text{DF}}$.
- **R2R-AC (GB)**: proposed R2R-AC in which $\delta \leftarrow \delta^{\text{GB}}$.
- **R2R-AC (Hy)**: proposed R2R-AC in which δ is computed using the Algorithm 2, the complete proposal.

These strategies have been tested through simulation on the problem presented in Section 2. Recall that the detailed methodology underlying the design of the feedforward controller, as well as the formulation of the desired trajectory, can be found in Ramirez-Laboreo et al. (2024). It is assumed that the model acting in the role of the real system and the feedforward controller are based on exactly the same equations. However, it is reasonable to expect discrepancies between the model parameters identification and the nominal (initial) feedforward controller parameters (see Table 1). To emulate these errors, two Monte Carlo analyses of 10 000 trials, and 300 switching operations in each trial are performed for each run-to-run strategy. For each Monte Carlo analysis, a test set with a different p -vector, (5), for each trial is generated. The first test set is associated with a situation where the errors are small, and the second with a situation where the errors are larger. To ensure fair comparisons, these test sets are common for all algorithms. The initial hyperparameters required of each algorithm (see Table 2) have been adjusted to optimize the results of the test where the errors are small. The same hyperparameters have been applied to the other test.

4.1. Small errors in the initial parameters

For this situation, each component of each p -vector of the real system model is randomly and independently perturbed up to 5%, i.e., the parameters of the real device under consideration vary with a uniform probability distribution between 95% and 105% of the values in Table 1.

Fig. 3 shows the results of the first analysis. The graphs represent the evolution of the cost, J , with respect to each evaluation or switching operation. Due to the large number of simulations required to capture the variability of the parameters across devices, the results are presented by the median (P_{50}) and the 10th and 90th percentiles (P_{10} and P_{90} , respectively) of the distribution of values obtained for the 10 000 simulated experiments. For reference, the cost of a switching operation without control, namely with a 30 V constant activation, is also plotted. To demonstrate the improvement introduced by R2R-AC, Fig. 3(a) (the best solution in Ramirez-Laboreo et al. (2024)) and 3(b) must be compared, as they differ only in the search strategy; both methods compute the step size using the same derivative-free approach. As can be seen, while the results at the end are quite similar, the new R2R-AC (DF) strategy shows a notable improvement in the convergence speed. While our previous feedforward run-to-run controller requires 300 switching operations for 90% of the trials to converge, R2R-AC (DF) requires only approximately 70. The same applies to 50% and 10% of the trials, for which the required number of switching operations is reduced by approximately half.

The other two strategies require the definition of J^* . For R2R-AC (GB), since gradient-based methods are sensitive to excessively large step sizes, J^* is variable and is calculated as the minimum evaluated cost, J^{\min} , multiplied by a positive constant $\gamma < 1$. With this strategy, the convergence speed also increases, but the final results are worse in this case. For R2R-AC (Hy), the expression of J^* (see Fig. 3(d)) is derived from a smoothed P_{90} behavior of R2R-AC (DF) under the constraint that the initial value matches the cost obtained from an evaluation without control. In this way, processes below the 90th percentile should remain unchanged, as can be seen when Figs. 3(b) and 3(d) are compared, while those above the 90th percentile should improve their performance when the hybrid method is applied to calculate the step size. For this reason, the 97.5th percentil ($P_{97.5}$) is also represented in these figures. Comparing this index, with R2R-AC (Hy) the target value is almost reached at the end of the trials, while with R2R-AC (DF) the convergence continues but progresses at a slow pace. Figs. 4(a) and 4(b) illustrate the effect of the hybrid strategy on the J evolution of two individual processes, one with slow convergence and another that converges to an unacceptable cost, respectively.

4.2. Larger errors in the initial parameters

Analogous to the previous subsection, the p test set has been generated with perturbations up to 25% instead of 5%.

Fig. 5 shows the results of the second analysis. As in the previous case, P_{10} , P_{50} and P_{90} of the distribution of values obtained for the 10 000 simulated experiments are shown. Each control strategy, i.e., R2R-PS+, R2R-AC (DF), R2R-AC (GB), and R2R-AC (Hy), uses the same hyperparameters as those employed in the previous subsection. This includes the definition of J^* across evaluations: for R2R-AC (GB), $J^* \leftarrow J^{\min} \cdot \gamma$, and for R2R-AC (Hy), J^* is the smoothed behavior of P_{90} with the set of nominal parameters perturbed up to 5%.

As can be seen, the improvement of R2R-AC, compared to Ramirez-Laboreo et al. (2024) is considerable. In contrast to cases with initial low p error, R2R-AC (GB) is able to offer better performance than R2R-AC (DF). However, our complete proposal, R2R-AC (Hy), achieves the best results regardless of the error level. As a summary, Table 3 compares P_{90} of J of the 10 000 simulated experiments at the 300th switching operation in both situations.

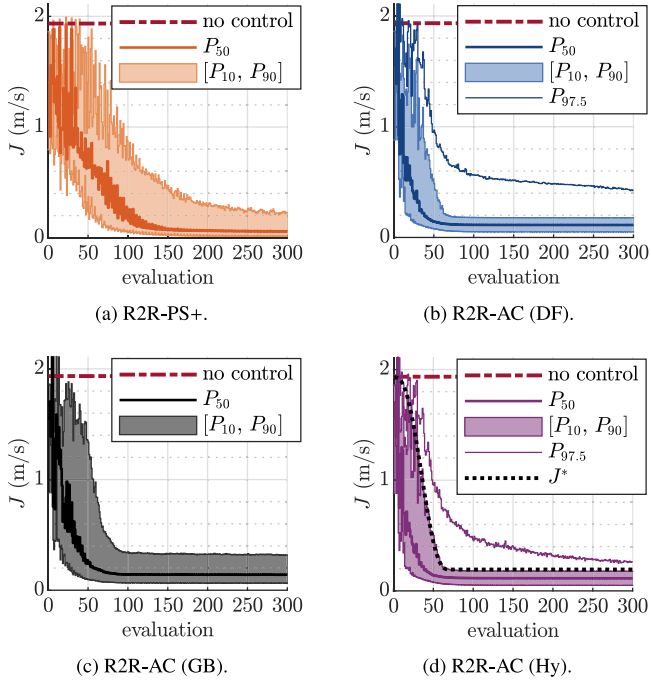


Fig. 3. Cost values with respect to the number of switching operations when parameter perturbations set to 5%. Each graph shows the median (P_{50}) and the 10th and 90th percentiles (P_{10} and P_{90} , respectively) of the distribution of values obtained for the 10000 simulated experiments. The cost without control is also represented. The 97.5th percentile ($P_{97.5}$) is also represented in (b) and (d) to show the hybrid method improvement.

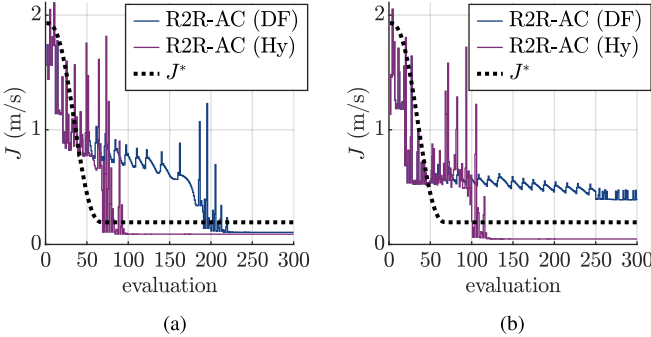


Fig. 4. Effect of the hybrid strategy versus derivative-free strategy. Evolution of J in two specific processes selected as representative. (a) Process with slow convergence. (b) Process with convergence to an unacceptable cost.

5. Conclusions

In this work, we have presented R2R-AC (Hy), a run-to-run control scheme with a new algorithm for iteratively adapting the parameters of a feedforward controller from indirect measurements. However, as outlined, the new algorithm could be used for other differentiable parametrized controllers. The improvement over the previous approach has been achieved both for small initial parameter errors and for larger errors, where the previous technique is not effective. The improvements have been obtained by integrating three concepts into the algorithm: a scheduled basis change based on the sensitivity of the feedforward law, a continuous process of exploration and exploitation, and a hybrid method that toggles between a derivative-free and a gradient-based strategy to calculate the step size. Likewise, this new algorithm automates two aspects of our previous work: the update of the coordinate system and the number of search dimensions to reduce, as the new

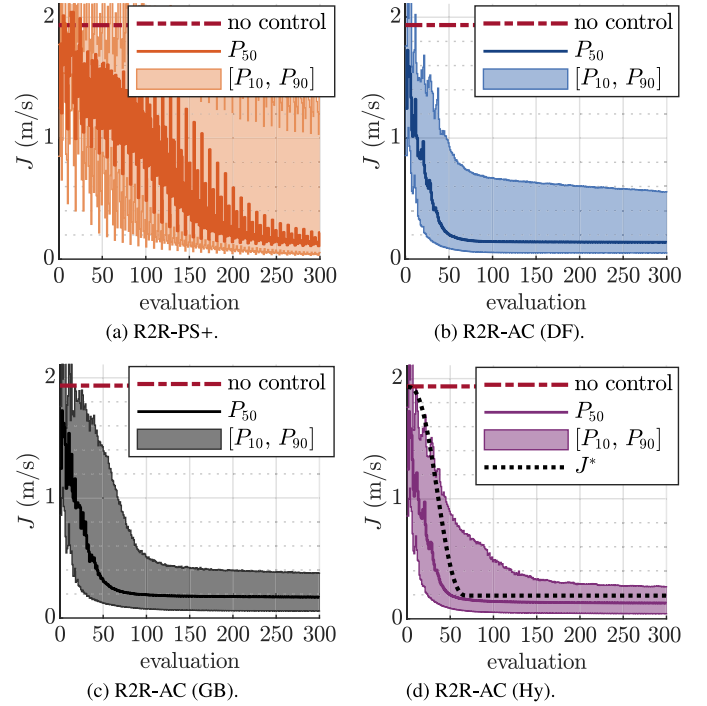


Fig. 5. Cost values with respect to the number of switching operations when parameter perturbations set to 25%. Each graph shows the median (P_{50}) and the 10th and 90th percentiles (P_{10} and P_{90} , respectively) of the distribution of values obtained for the 10000 simulated experiments. The cost without control is also represented.

Table 3

Comparison of P_{90} of J (m/s) in the 300th switching operation.

Perturbation	R2R-PS+	R2R-AC (DF)	R2R-AC (GB)	R2R-AC (Hy)
5%	0.2268	0.1777	0.3178	0.1772
25%	1.2381	0.5541	0.3750	0.2645

algorithm selects the minimum number of dimensions to improve its feedforward controller behavior.

As future work, we would like to address the possibility of an estimation technique for the objective function J^* or to analyze and improve the gradient-based step-size calculation in other scenarios, such as stochastic processes or noisy measurements, by considering alternative approaches from the literature. In addition, we also intend to perform real laboratory tests on different systems to verify that the experimental results agree with those observed in simulation and the generality of the method.

CRedit authorship contribution statement

Eloy Serrano-Seco: Data curation, Writing – original draft, Investigation, Conceptualization, Formal analysis, Software, Methodology, Visualization, Validation. **Eduardo Moya-Lasheras:** Supervision, Writing – review & editing, Conceptualization. **Edgar Ramirez-Laboreo:** Supervision, Funding acquisition, Writing – review & editing, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part via grants CPP2021-008938, PID2021-124137OB-I00, and TED2021-130224B-I00, funded by MCIN/AEI/ 10.13039/501100011033, by the European Union NextGenerationEU/PRTR, and by ERDF A way of making Europe, in part by the Government of Aragón - EU, under grant T45_23R, in part by the “Programa Investigo” funded by the European Union - Next Generation EU, and in part by Fundación Ibercaja and the University of Zaragoza, Spain, via project JIUZ2023-IA-07.

References

- Al Saaideh, M., Boker, A. M., & Al Janaideh, M. (2022). Output-feedback control of electromagnetic actuated micropositioning system with uncertain nonlinearities and unknown gap variation. In *IEEE conf. decision and control* (pp. 2481–2486). IEEE.
- Angadi, S. V., & Jackson, R. L. (2022). A critical review on the solenoid valve reliability, performance and remaining useful life including its industrial applications. *Eng. Fail. Anal.*, 136, Article 106231.
- Benosman, M., & Atınc, G. M. (2015). Extremum seeking-based adaptive control for electromagnetic actuators. *International Journal of Control*, 88(3), 517–530.
- Braun, T., Reuter, J., & Rudolph, J. (2018). Observer design for self-sensing of solenoid actuators with application to soft landing. *IEEE Transactions on Control Systems Technology*, 27(4), 1720–1727.
- Deschaux, F., Gouaisbaut, F., & Ariba, Y. (2018). Nonlinear control for an uncertain electromagnetic actuator. In *IEEE conf. decision and control* (pp. 2316–2321). IEEE.
- Gergiç, B., & Hercog, D. (2019). Design and implementation of a measurement system for high-speed testing of electromechanical relays. *Measurement*, 135, 112–121.
- Grotjahn, M., & Heimann, B. (2002). Model-based feedforward control in industrial robotics. *International Journal of Robotics Research*, 21(1), 45–60.
- Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2), 159–195.
- Lévine, J. (2011). On necessary and sufficient conditions for differential flatness. *Appl. Algebra Eng., Commun. Comput.*, 22(1), 47–90.
- Lewis, R. M., & Torczon, V. (2000). Pattern search methods for linearly constrained minimization. *SIAM J. Optimization*, 10(3), 917–941.
- Loshchilov, I., Schoenauer, M., & Sebag, M. (2011). Adaptive coordinate descent. In *Prod. 13th GECCO* (pp. 885–892).
- Ma, C., Wu, L., & Weinan, E. (2022). A qualitative study of the dynamic behavior for adaptive gradient algorithms. In *Math. sci. mach. learning* (pp. 671–692). PMLR.
- Moulay, E., Léchappé, V., & Plestan, F. (2019). Properties of the sign gradient descent algorithms. *Inf. Sci.*, 492, 29–39.
- Moya-Lasheras, E., Ramirez-Laboreo, E., & Serrano-Seco, E. (2023). Run-to-Run Adaptive Nonlinear Feedforward Control of Electromechanical Switching Devices. *IFAC-PapersOnLine*, 56(2), 5358–5363, 22nd IFAC World Congr.
- Moya-Lasheras, E., & Sagues, C. (2024). Iterative-only learning control for soft landing of short-stroke reluctance actuators. *Control Engineering Practice*, 152, Article 106067.
- Ramirez-Laboreo, E., Moya-Lasheras, E., & Serrano-Seco, E. (2024). Faster run-to-run feedforward control of electromechanical switching devices: A sensitivity-based approach. In *2024 European control conf.* (pp. 1321–1326).
- Schroedter, R., Roth, M., Janschek, K., & Sandner, T. (2018). Flatness-based open-loop and closed-loop control for electrostatic quasi-static microscanners using jerk-limited trajectory design. *Mechatronics*, 56, 318–331.
- Winkel, F., Scholz, P., Wallscheid, O., & Böcker, J. (2023). Reducing contact bouncing of a relay by optimizing the switch signal during run-time. *IEEE Trans. Autom. Sci. Eng.*.
- Yeh, S.-S., & Hsu, P.-L. (1999). An optimal and adaptive design of the feedforward motion controller. *IEEE/ASME Transactions on Mechatronics*, 4(4), 428–439.