

## RESEARCH ARTICLE

# Surrogate-Assisted Adaptive Controller Tuning for BLDC Motors: An Experimental Evaluation

Alam Gabriel Rojas-López<sup>a</sup>, Miguel Gabriel Villarreal-Cervantes<sup>a</sup>, Alejandro Rodríguez-Molina<sup>b</sup>, Daniel Molina-Pérez<sup>c</sup> y Gerardo Hernández-Hernández<sup>b</sup>

<sup>a</sup>Instituto Politécnico Nacional - Centro de Innovación y Desarrollo Tecnológico en Cómputo, Postgraduate Department, Av. Juan de Dios Bátiz, Gustavo A. Madero, 07700, Mexico City, Mexico;

<sup>b</sup>Universidad Autónoma de la Ciudad de México, Colegio de Ciencia y Tecnología, Av. La Corona, Gustavo A. Madero, 06720, Mexico City, Mexico;

<sup>c</sup>Department of Computer Science at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Av. Instituto Politécnico Nacional, Gustavo A. Madero, 07360, Mexico City, Mexico

## ARTICLE HISTORY

Compiled October 8, 2026

## ABSTRACT

Adaptive controller tuning via metaheuristic optimization provides robust compensation against parametric uncertainties and external disturbances, but its high computational cost limits its practical applicability in fast-dynamic systems. This work proposes an experimentally validated surrogate-assisted framework for metaheuristic-based indirect adaptive controller tuning, reducing computational cost while maintaining effective control performance. The methodology adopts a surrogate-assisted integration framework that enables the interchangeability and comparison of surrogate-model strategies. In this work, experimental validation is performed on a Brushless Direct Current motor using three surrogate model techniques: Response Surface Method (RSM), Radial Basis Function (RBF), and Gaussian Process Regression (GPR). A statistical comparative analysis across experiments of increasing complexity evaluates trade-offs between control performance and computational cost relative to a conventional metaheuristic-based tuning strategy. Results confirm computational savings without degrading closed-loop performance. In particular, the RSM variant achieved up to a 43-fold reduction in computational burden, while the RBF variant delivered superior dynamic performance. The results demonstrate the interchangeability of the evaluated surrogate-model strategies within the proposed framework and highlight the practical trade-offs associated with their selection in fast-response adaptive control tuning

## KEYWORDS

Adaptive controller tuning; Surrogate model; Metaheuristic optimization; Real-time optimization

## 1. Introduction

A significant concern in modern mechatronic systems is their ability to cope with adverse conditions through an appropriate control strategy. In some cases, these con-

ditions can be anticipated during the design phase by considering typically overlooked interactions, such as component friction or variable loads. However, a much larger set of adverse conditions remains unpredictable, including parametric uncertainties and external disturbances (Bolton 2003). Regardless of the control strategy employed, performance depends primarily on fine-tuning the controller’s parameters. One of the most comprehensive classifications of controller tuning strategies divides them into four main categories (Reynoso-Meza et al. 2014; Rodríguez-Molina et al. 2020).

The first category is analytical tuning, which utilizes the system model’s dynamics. Through mathematical analysis, such as the root locus method (Alyoussef and Kaya 2023) or Lyapunov stability analysis (Zaki, Rashid, and Masud 2024), the controller is tuned before system operation. The second category is heuristic tuning, in which experience-based rules estimate controller parameters prior to execution based on the system’s behavior, as in the Ziegler-Nichols or Cohen-Coon methods (Ziegler and Nichols 1942; Cohen and Coon 1953). The primary limitation of both analytical and heuristic approaches is their heavy reliance on the completeness of the dynamic model, making them susceptible to unmodeled adverse conditions. Moreover, they do not guarantee a desired performance level, and parameter computation can become overly complex.

The third category corresponds to the optimally offline controller tuning approach (OOCTA), in which the tuning process is formulated as an optimization problem and solved prior to execution (Paredes-Ballesteros et al. 2025). Although OOCTA still relies on an accurate dynamic model, it offers performance guarantees. Nonetheless, since the controller parameters remain fixed during operation, their adaptability to unforeseen adverse conditions is inherently limited.

The fourth approach corresponds to adaptive tuning (Landau et al. 2011), where the controller parameters are updated iteratively during system execution. This approach is further divided into direct adaptive and indirect adaptive strategies. In direct adaptive tuning, a fixed reference model is employed to measure the discrepancy between the system response and the model, adjusting only the controller parameters accordingly (Zhu, Gao, and Huang 2022; Aliman et al. 2022). In contrast, the indirect adaptive controller tuning approach (IACTA) updates both the reference model and the controller parameters during system operation, making it one of the most effective tools for handling adverse conditions (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2023). However, IACTA has the significant drawback of a high computational burden, limiting its experimental use to relatively simple systems, such as those with a low number of states or slow dynamic responses.

In recent years, pattern recognition techniques have been introduced to mitigate IACTA’s computational cost. These methods employ real-time system data to iteratively construct a surrogate model (SM) that demands considerably less computational effort than the actual dynamic model. Beyond reducing computational cost, the effectiveness of a surrogate model in adaptive controller tuning strategies relies primarily on its capability to reproduce the system’s behavior. Consider Fig. 1 as an example, where the system output  $y$  is controlled to follow a reference value. When adverse conditions, such as parametric uncertainties or external disturbances, affect the system, adaptive controller tuning seeks to determine new controller parameters that compensate for their effects and maintain the desired control performance. The surrogate model can be leveraged for this purpose by collecting and reproducing multiple samples over an identification time window  $I_w$  to characterize the system behavior under such conditions. Fig. 1 depicts different surrogate models that attempt to replicate the real system during the identification time window  $I_w$ , rather than the full,

costly, explicit dynamic model, which may be computationally expensive or even unavailable. Notably, each surrogate model is based on different modeling paradigms (Jin 2011), including geometric or interpolation-based methods (e.g., RSM and RBF), statistical or probabilistic techniques (e.g., GPR), logical approximations (e.g., decision trees), and connectionist approaches (e.g., ANN) (Jiang, Zhou, and Shao 2020; Forrester, Sóbester, and Keane 2008), which generate distinct approximations. As a consequence, their ability to capture the relevant system dynamics, maintain control performance, and reduce computational burden may differ substantially, highlighting the importance of systematically analyzing and comparing different surrogate-model families within adaptive controller tuning frameworks.

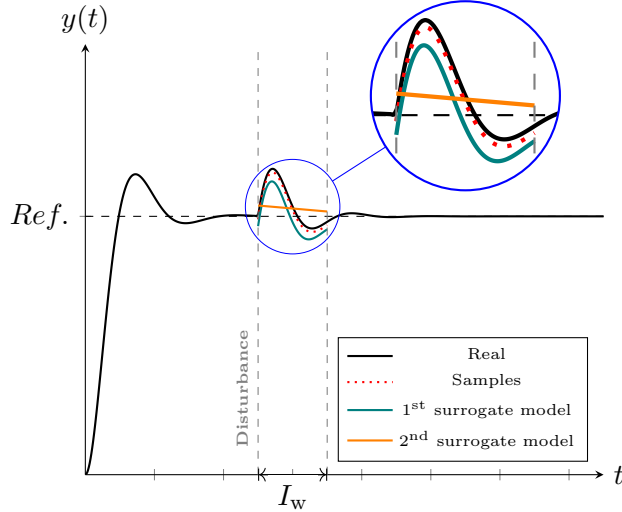


Figure 1.: Surrogate models tracking the system response during a disturbance.

For instance, (Šafarič et al. 2020) utilizes a single-layer artificial neural network (ANN) to build a SM of a one-degree-of-freedom mass-spring robotic system. The approach is compared against a proportional-integral (PI) controller tuned via Oocta using different evolutionary algorithms (EAs). Although the results are encouraging, the study focuses solely on reducing the computational burden associated with EAs rather than assessing the broader impact of surrogate modeling on IACTA performance. Without comparisons against other IACTA approaches or surrogate modeling techniques, the extent to which the ANN-based surrogate model reduces computational cost remains uncertain.

One of the most widely used surrogate models (SMs) in adaptive controller tuning is Gaussian process regression (GPR). Numerous studies combine GPR with Bayesian optimization (BO) for adaptive tuning, primarily because of their natural synergy and the premise that BO requires fewer function evaluations than traditional evolutionary algorithms (EAs). However, (Lan et al. 2022) highlights that, beyond the number of function evaluations, other factors, such as the complexity of the objective function, also significantly affect the convergence speed of optimization algorithms.

A particularly active research area that combines GPR and BO is physical human-robot interaction (pHRI), where the unpredictability of human behavior presents a significant challenge. Examples include (Roveda, Forgione, and Piga 2020a,b), where the parameters of a Cartesian impedance controller for a seven-degrees-of-freedom redundant robot manipulator are adaptively tuned. Similarly, (Tsfazgi et al. 2023)

applies this combination to adaptively tune a mixed proportional-derivative and computed torque controller (PD and CTC) for a two-degrees-of-freedom robot. Another instance is (Almasalmah et al. 2024), where a model predictive controller (MPC) is adaptively tuned using BO and GPR in a haptic system. While these contributions report promising results, their scope is limited to variations in the optimization algorithm, primarily comparing BO with its variants or traditional Oocta strategies, without exploring alternative SMs or explicitly evaluating the computational load.

Beyond pHRI applications, the GPR and BO combination has also been employed in complex industrial processes. For example, (Chen and Cheng 2021) presents an online tuning approach for the controller of an ABB IRB140 robot in tight-tolerance peg-in-hole and torque converter assembly tasks. Additionally, SMs, particularly GPR, have been applied in industrial scenarios where the dynamic model is either unknown or too complex to be practically feasible. Representative cases include (Lu et al. 2021; Xu et al. 2024; Cho, Shao, and Mesbah 2024), where controllers for heating, ventilation, and air conditioning systems, building thermal processes, and plasma-assisted thin-film deposition are adaptively tuned. Further examples can be found in (Zhang et al. 2024; Qiu et al. 2025), where controllers for chemical continuous stirred tank reactors are adaptively tuned. Notably, these contributions continue to focus solely on comparisons between BO and its variants or conventional Oocta strategies, without exploring alternative SMs or quantifying the impact on computational burden.

It is also important to note that GPR is not exclusively coupled with BO, but has also been combined with evolutionary algorithms. For instance, in (Chen et al. 2024), GPR is used alongside the non-dominated sorting genetic algorithm II (NSGA-II) for the adaptive tuning of a PD controller in a three-degrees-of-freedom system. However, this work again limits its scope to comparisons with impedance controllers tuned via conventional Oocta, without considering alternative SMs or analyzing computational cost reductions.

Nonetheless, GPR is not the only SM employed in adaptive controller tuning problems. Radial basis function (RBF) strategies have also shown promising results, particularly in pHRI applications. For example, (Roveda et al. 2023a,b; Maccarini et al. 2022; Franceschi et al. 2023) implement RBFs combined with game theory and preference-based optimization algorithms to adaptively tune the parameters of a Cartesian impedance controller for a seven-degrees-of-freedom redundant robot manipulator. However, these studies remain limited to comparisons with conventional Oocta strategies and do not assess RBF’s ability to reduce computational load or benchmark it against other SMs.

It is also noteworthy that both GPR- and RBF-based contributions tend to classify system complexity mainly by the number of states or degrees of freedom. Yet, other factors, such as the system’s response time, significantly influence complexity, especially in adaptive controller tuning problems, where the adaptation mechanism must operate rapidly enough to compensate for adverse conditions during fast dynamic transitions (Lishchenko, Gushchin, and Larshin 2024).

Another widely used SM in adaptive controller tuning problems is the response surface method (RSM), which employs multivariate, multitarget polynomial regression. Previous applications of the RSM, conducted by the authors of the present work, are reported in a simulation environment in (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2024; Rodríguez-Molina et al. 2024; Rojas-López et al. 2024a,b, 2025). The RSM is combined with evolutionary algorithms (EAs) for the adaptive tuning of controller parameters in a brushless direct current (BLDC) motor, a two-degrees-of-freedom mechanism, an omnidirectional mobile robot, and a three-degrees-

of-freedom planar serial robot, respectively. Notably, in (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2024), an in-depth analysis is conducted by varying the degree of the polynomial regression and testing different EAs (Differential Evolution, Genetic Algorithm, and Particle Swarm Optimization) at each evaluated degree. The results showed that changing the degree of the polynomial regression has a greater effect on the control system’s performance than the EA implemented. This suggests that selecting the surrogate model is more important than selecting the optimization algorithm, limiting the choice of optimizer mainly to existing computational capabilities. Despite these contributions, these works may exhibit limited scalability, as a large number of samples are required to accurately estimate the response surface, thereby increasing computational cost and reducing efficiency in real-time scenarios.

These studies provide prior evidence that surrogate-based adaptive controller tuning can be formulated across different mechatronic systems while preserving the general surrogate-assisted optimization concept. They also illustrate that surrogate models have been successfully combined with different optimization strategies, indicating that surrogate-assisted tuning is not intrinsically restricted to a single optimization algorithm. However, these studies do not establish that different surrogate-model families can be systematically interchanged and compared within a common experimental tuning framework. It is important to note that the aforementioned studies employing ANN, GPR, RBF, or RSM approaches for adaptive controller tuning are generally tailored to specific systems or applications, with each surrogate-model family being treated as an independent, standalone solution governed by its own modeling assumptions and operational principles. Consequently, there remains a need for a systematic methodology that facilitates the integration and comparative evaluation of different surrogate-model families within a common adaptive controller tuning framework. This need hinders the establishment of comparable and reproducible evaluation procedures across different surrogate-model families, despite the fact that they may exhibit heterogeneous modeling behaviors and distinct strengths and limitations arising from their underlying assumptions regarding function approximation and uncertainty representation.

Beyond these methodological limitations, the reported contributions tend to emphasize different aspects depending on the particular experimental conditions and SM employed, which limits a comprehensive understanding of the practical implications and trade-offs associated with surrogate-model selection in indirect adaptive controller tuning problems. Moreover, the associated computational cost is either not systematically analyzed or simply assumed to be fulfilled within the adaptive cycle of the closed-loop system, often without thorough validation. Previous studies have also mainly overlooked high-speed systems in their experimental validations, applying their approaches primarily to slow-dynamic processes or relying solely on simulation-based evidence. Consequently, due to the lack of a common methodology, systematic comparison, computational-cost validation, and experimental evidence on high-speed systems, the practical applicability of these approaches to real-world scenarios, where response time is a critical factor in system variability, remains unclear.

### 1.1. *Contributions*

Considering the context in the previous section, this work addresses existing knowledge gaps in the need for a formal methodology to implement diverse surrogate models into metaheuristic-based indirect adaptive controller tuning problems, the experimental validation of the methodology in fast-response mechatronic systems, and the absence

of systematic comparative analyses among different surrogate models (SMs) within a common surrogate-based control tuning framework. Based on these gaps, the main contributions of this paper are summarized as follows:

- An experimentally validated surrogate-assisted online optimization methodology for metaheuristic-based indirect adaptive controller tuning, formulated to accommodate alternative surrogate-model strategies within a common identification-prediction-tuning framework.
- A detailed analysis of the characteristics, limitations, and implementation considerations of the proposed methodology, covering three SM techniques: response surface method (RSM), radial basis function (RBF), and Gaussian process regression (GPR).
- A statistical analysis across three experiments of increasing complexity on a brushless direct current (BLDC) motor under identical conditions quantifying the trade-offs between control performance and computational cost, providing practical guidelines for surrogate-model selection in the tested application.

The proposed framework uses a two-level structure: the first level builds a surrogate model of the plant, and the second uses it to optimally tune the controller online. The formulation separates the surrogate-model strategy from the general identification, prediction, and controller-tuning procedures, allowing alternative surrogate-model techniques to be incorporated within the same framework. By keeping the plant, controller structure, optimization algorithm, and hardware platform fixed, the experimental design isolates the effect of surrogate-model selection on computational burden and closed-loop performance. Experimental results on a BLDC motor demonstrate the real-time feasibility of the methodology and show a reduced computational burden compared with explicit dynamic model-based adaptive tuning. To the best of the authors' knowledge, experimental studies systematically comparing multiple surrogate-model families within a common online adaptive controller-tuning framework remain scarce. The present work contributes to the literature by providing a controlled experimental benchmark of three representative surrogate-model families (RSM, RBF, and GPR) on a BLDC motor platform. The results therefore provide experimental evidence of surrogate-model interchangeability within the tested BLDC motor tuning framework.

## 1.2. *Structure of the Work*

The remainder of this document is structured as follows. Section 2 introduces the proposed surrogate-model-based indirect adaptive controller tuning approach, detailing its main characteristics and implementation considerations, along with three variants corresponding to different surrogate models. Section 3 describes the case study, which involves a real-world fast-response system, specifically a brushless direct current (BLDC) motor. Section 4 presents the experimental results obtained from three experiments of increasing complexity, along with a statistical analysis that highlights the advantages of the proposed methodology. Finally, Section 5 summarizes the main conclusions, highlights key findings, and outlines potential future research directions.

## 2. Surrogate-model-based indirect adaptive controller tuning proposal

To better illustrate the operation of surrogate models within indirect adaptive controller tuning strategies, Fig. 2 is provided. The red blocks represent a traditional closed-loop controller operating at a fixed sampling frequency with a sampling period  $\delta t_l$ . This configuration is common in various control strategies, where the system outputs are measured and compared to a reference value for regulation or tracking purposes. The main concept of surrogate-model-based indirect adaptive controller tuning approaches (SIACTAs) is the inclusion of a parallel loop, where information is continuously stored in a dataset. A pattern recognition technique utilizes this data to construct a surrogate model that replicates the system's behavior. This surrogate model is then used in a prediction stage, where simulations are performed to determine the new optimal controller parameter vector  $\mathbf{k}^*$ . Both stages operate at a fixed update frequency defined by the update period  $\Delta T_U$ .

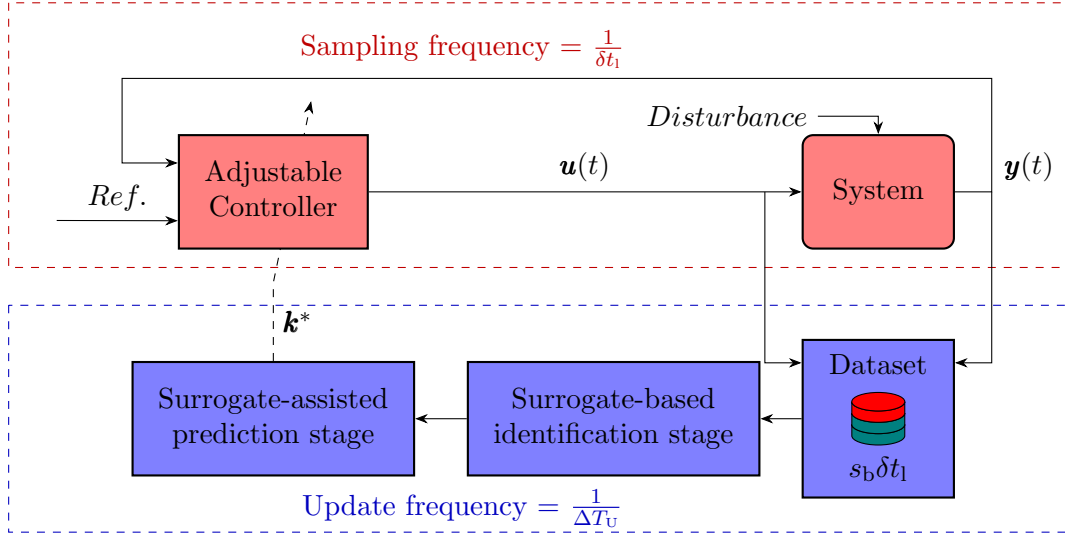


Figure 2.: SIACTA implementation scheme.

Fig. 3 is provided to show the time-based operation of SIACTA. Consider that the system's information (states, outputs, and control signals) is sampled at a period  $\delta t_l$  seconds. When the current time  $t_c$  reaches the beginning of an update period  $\Delta T_U$ , the stored information collected during an identification time window  $I_w = [t_c - s_b \delta t_l, t_c]$ , which includes  $s_b$  backward steps up to the latest sample, is employed to construct a surrogate model. This surrogate model is then used to simulate the system's behavior over a prediction time window  $P_w = [t_c, t_c + s_f \delta t_l]$ , which spans  $s_f$  forward steps from the current time. Using these predictions, a new controller parameter vector is determined to improve the target performance criterion. This process is repeated iteratively every  $\Delta T_U$ , where each  $I_w$  captures updated system information and each  $P_w$  identifies controller parameters to compensate for adverse conditions, such as parametric uncertainties or external disturbances.

### 2.1. Surrogate-model-based Identification Stage

Let (1) represent the system's dynamic model, where  $\mathbf{x} \in \mathbb{R}^{n_x}$ ,  $\mathbf{y} \in \mathbb{R}^{n_y}$ , and  $\mathbf{u} \in \mathbb{R}^{n_u}$  denote the state, output, and control signal vectors, respectively, with dimensions  $n_x$ ,

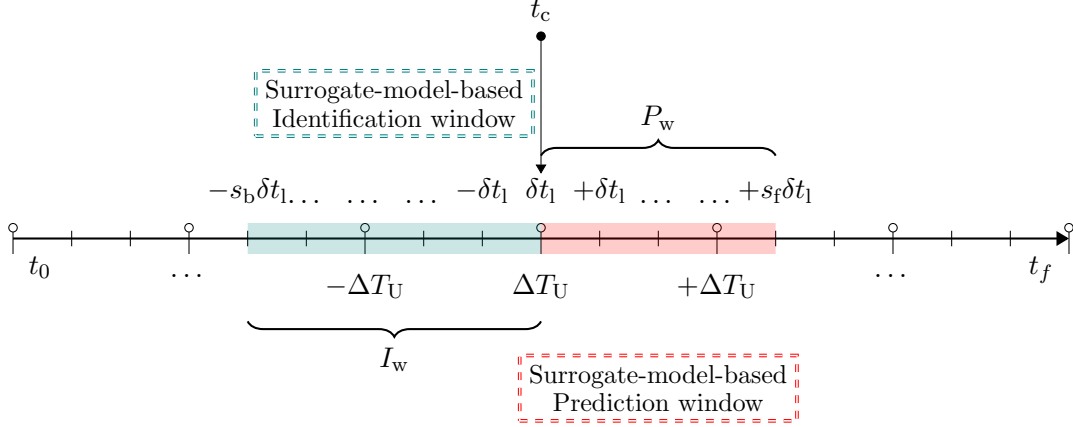


Figure 3.: SIACTA operation over time.

$n_y$ , and  $n_u$ . Additionally,  $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$  is the matrix that relates the system states to its outputs, under the assumption that the full system information is measurable, and the time relationship  $\mathbf{y}(t+1) = \int \dot{\mathbf{x}}(t) dt$  holds.

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{1}$$

This time-based relationship is leveraged to construct a dataset, as shown in Table 1. This dataset contains all the information corresponding to the identification time window  $I_w$ . On the one hand,  $\tilde{\mathbf{y}}_l \in \mathbb{R}^{n_y}$ ,  $\forall l \in 1, \dots, s_b$ , represents the  $l$ -th target vector, which contains the system's output vector sampled at  $t_c - l\delta t_1$ . On the other hand,  $\tilde{\mathbf{x}}_l \in \mathbb{R}^{n_x + n_y}$  denotes the feature vector, which includes the corresponding system states and control signals at each sampling instant within  $I_w$ .

Table 1.: Dataset creation example.

| Target vectors                                               | Feature vectors                                                                                                |
|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| $\tilde{\mathbf{y}}_1 = \mathbf{y}(t_c - \delta t_1)$        | $\tilde{\mathbf{x}}_1 = [\mathbf{x}(t_c - \delta t_1)^\top, \mathbf{u}(t_c - \delta t_1)^\top]^\top$           |
| $\vdots$                                                     | $\vdots$                                                                                                       |
| $\tilde{\mathbf{y}}_l = \mathbf{y}(t_c - l\delta t_1)$       | $\tilde{\mathbf{x}}_l = [\mathbf{x}(t_c - l\delta t_1)^\top, \mathbf{u}(t_c - l\delta t_1)^\top]^\top$         |
| $\vdots$                                                     | $\vdots$                                                                                                       |
| $\tilde{\mathbf{y}}_{s_b} = \mathbf{y}(t_c - s_b\delta t_1)$ | $\tilde{\mathbf{x}}_{s_b} = [\mathbf{x}(t_c - s_b\delta t_1)^\top, \mathbf{u}(t_c - s_b\delta t_1)^\top]^\top$ |

Using the information contained in this dataset, different surrogate modeling strategies based on supervised machine learning formulation for continuous-variable regression (or continuous mappings) can be applied, given their similar properties (Forrester, Sóbester, and Keane 2008). The following subsections describe the surrogate modeling strategies implemented throughout this work: Response Surface Method, Radial Basis Functions, and Gaussian Process Regression. Notably, these strategies are selected for their popularity in the literature and their rapid response. Consequently, more time-consuming approaches are not considered due to the fast-response requirements of the case study presented in Section 3. It is essential to note that these strategies share certain standard features and, consequently, use similar nomenclature. However, some

variables may differ in dimensionality, ranging from vectors to matrices. A summary of the main characteristics, underlying assumptions, and specific requirements of each strategy is provided at the end of this section.

### 2.1.1. Response Surface Method

The first SM strategy is the Response Surface Method (RSM), which is selected for its geometric interpolation principles. Owing to its simplicity, RSM often serves as a baseline or is integrated within other SMs approaches (Myers, Montgomery, and Anderson-Cook 2016). Its polynomial formulations also allow fast evaluation, making it particularly suitable for real-time adaptive control applications. The RSM approximates the system's dynamic model (1) using a multivariate, multi-target polynomial function (Myers et al. 2012; Chelladurai et al. 2021; Khuri and Mukhopadhyay 2010). Before detailing the RSM operational procedure, it is important to note that prior work (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2024; Rodríguez-Molina et al. 2024) has evaluated multiple polynomial degrees, concluding that a second-degree polynomial yields the best performance for SIACTA problems. This is attributed to the inherent mathematical relationship between polynomial degrees, which depends on whether the degree is even or odd (Stewart, Clegg, and Watson 2020; Borwein and Erdélyi 1995). Consequently, the present work adopts a second-degree polynomial formulation to describe and implement the RSM.

The approximation of the identification time window  $I_w$  information to a second-degree polynomial is performed according to (2), where  $\tilde{y}_{l,k}$  denotes the  $k$ -th element,  $\forall k \in \{1, \dots, n_y\}$ , of the  $l$ -th target vector, which is approximated through a second-degree polynomial. In this formulation, the first-degree terms consider each element of the  $l$ -th feature vector independently, i.e.,  $\tilde{x}_{l,1} + \dots + \tilde{x}_{l,i} + \dots + \tilde{x}_{l,n_x+n_u}$ ,  $\forall i \in \{1, \dots, n_x+n_u\}$ . In contrast, the second-degree terms include all possible permutations involving the products of elements within the  $l$ -th feature vector, namely,  $\tilde{x}_{l,1}\tilde{x}_{l,1} + \dots + \tilde{x}_{l,i}\tilde{x}_{l,j} + \dots + \tilde{x}_{l,n_x+n_u}\tilde{x}_{l,n_x+n_u}$ ,  $\forall i, j \in \{1, \dots, n_x+n_u\}$ . Each polynomial term is associated with its corresponding  $\beta$  coefficient.

$$\tilde{y}_{l,k} = \underset{\text{bias}}{\beta_{0,k}} + \underset{1^{\text{st}} \text{ degree terms}}{\sum_{i=1}^{n_x+n_u} \beta_{i,k} \tilde{x}_{l,i}} + \underset{2^{\text{nd}} \text{ degree terms}}{\sum_{i=1}^{n_x+n_u} \sum_{j \geq i}^{n_x+n_u} \beta_{i,j,k} \tilde{x}_{l,i} \tilde{x}_{l,j}} \quad (2)$$

For simplicity, the polynomial expression can be rewritten in matrix form as shown in (3), which consists of the following components:

- The target matrix  $\overline{\mathbf{Y}} \in \mathbb{R}^{s_b \times n_y}$  contains the output information of the real system collected within the identification time window.
- The feature matrix  $\overline{\mathbf{X}} \in \mathbb{R}^{s_b \times \nu}$  includes the corresponding system state and control signal information. Each row of  $\overline{\mathbf{X}}$  is composed of the unit element and variables that multiply the  $\beta$  coefficients, corresponding to the bias, first-degree, and second-degree terms and expressed as  $\overline{\mathbf{x}} = [1, \text{1st-degree terms}, \text{2nd-degree terms}]^T \in \mathbb{R}^\nu$ . The total number of terms in the second-degree polynomial (including the bias) is calculated as  $\nu = \frac{(n_x+n_u+1)(n_x+n_u+2)}{2}$ .
- The coefficient matrix  $\overline{\boldsymbol{\beta}} \in \mathbb{R}^{\nu \times n_y}$  contains the coefficients associated with the second-degree polynomials. For instance, the related coefficient vector for the

first polynomial (2) is grouped as  $\bar{\beta}_0 \in \mathbb{R}^\nu$ .

$$\begin{aligned} \begin{bmatrix} \tilde{\mathbf{y}}_1^\top \\ \vdots \\ \tilde{\mathbf{y}}_l^\top \\ \vdots \\ \tilde{\mathbf{y}}_{s_b}^\top \end{bmatrix} &= \begin{bmatrix} \bar{\mathbf{x}}_1^\top \\ \vdots \\ \bar{\mathbf{x}}_l^\top \\ \vdots \\ \bar{\mathbf{x}}_{s_b}^\top \end{bmatrix} \begin{bmatrix} \bar{\beta}_0, \dots, \bar{\beta}_{n_y} \end{bmatrix} \\ &\triangleq \bar{\mathbf{Y}} = \bar{\mathbf{X}} \bar{\boldsymbol{\beta}} \end{aligned} \quad (3)$$

Since the target matrix  $\bar{\mathbf{Y}}$  and the feature matrix  $\bar{\mathbf{X}}$  contain data sampled directly from the system, the surrogate model can approximate the system's dynamic model (1) solely through the computation of the optimal coefficient matrix. This is obtained using the Moore-Penrose pseudo-inverse, expressed as  $\bar{\boldsymbol{\beta}}^* = (\bar{\mathbf{X}}^\top \bar{\mathbf{X}})^{-1} \bar{\mathbf{X}}^\top \bar{\mathbf{Y}}$ . Once the optimal coefficient matrix  $\bar{\boldsymbol{\beta}}^*$  is computed, the final RSM-based surrogate model is defined as in (4), where  $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$  and  $\hat{\mathbf{x}} \in \mathbb{R}^\nu$  represent the surrogate target and feature vectors, respectively. It is important to note that the surrogate feature vector  $\hat{\mathbf{x}}$  is constructed following the second-degree polynomial formulation, similarly to  $\bar{\mathbf{x}}$ .

$$\hat{\mathbf{y}}^\top = \hat{\mathbf{x}}^\top \bar{\boldsymbol{\beta}}^* \quad (4)$$

### 2.1.2. Radial Basis Function

The second surrogate modeling strategy examined is the Radial Basis Function (RBF), widely regarded as one of the simplest kernel-based extensions of geometric interpolation methods (Buhmann 2000). The use of a “kernel” provides adaptability and flexibility to the regression process, enabling the representation of nonlinear relationships while maintaining analytical simplicity. Nevertheless, kernel computation increases the computational load of the regression process and depends critically on the correct selection and parameterization of the kernel (Bishop and Nasrabadi 2006). Specifically, RBF utilizes an unbiased first-degree polynomial (Jiang, Zhou, and Shao 2020). It is worth noting that, in applications outside the SIACTA framework, extensions to higher-degree biased polynomials have been proposed; however, such extensions significantly increase the computational burden. The RBF approach is applied to the information collected during the identification time window  $I_w$ , as described in (5), where  $\tilde{y}_{l,k}$  denotes the  $k$ -th element,  $\forall k \in 1, \dots, n_y$ , of the  $l$ -th target vector, approximated through a first-degree polynomial. Each term of this polynomial is computed based on all possible  $s_b$  combinations of feature vectors from the  $I_w$  dataset, through a distance-dependent function  $\phi_{j,i}(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_i)$  between the  $\tilde{\mathbf{x}}_i$  feature vector and the fixed  $\tilde{\mathbf{x}}_j$  vector,  $\forall i, j \in 1, \dots, s_b$ . Additionally,  $\omega_{k,i}$  denotes the coefficient that relates the respective  $i$ -th polynomial term to the  $k$ -th element of the target vector.

$$\tilde{y}_{l,k} = \sum_{i=1}^{s_b} \omega_{k,i} \phi_{j,i}(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_i) \quad (5)$$

Expressing this formulation in matrix form results in (6), which consists of the following components:

- The target matrix  $\bar{\mathbf{Y}} \in \mathbb{R}^{s_b \times n_y}$  contains the output information of the real system, obtained within the identification time window.
- The feature matrix  $\bar{\Phi} \in \mathbb{R}^{s_b \times s_b}$  includes the corresponding system states and control signal information, structured through all possible pairwise combinations of feature vectors from the  $I_w$  dataset. In this work, a normalized Gaussian radial basis function, as defined in (7), is employed as the distance-dependent function, where  $\gamma \in (0, \infty) \subset \mathbb{R}$  represents the scale factor influencing the dispersion of the function. It is important to note that, unlike RSM, the RBF modeling strategy uses feature vectors without additional polynomial expansions, resulting in a vector length of  $\nu = n_x + n_u$ .
- The coefficient matrix  $\bar{\Omega} \in \mathbb{R}^{s_b \times n_y}$  contains the coefficients corresponding to the first-degree polynomials. For instance, the related coefficient vector for the first polynomial (5) is grouped as  $\omega_1 \in \mathbb{R}^\nu$ .

$$\begin{bmatrix} \tilde{\mathbf{y}}_1^\top \\ \vdots \\ \tilde{\mathbf{y}}_l^\top \\ \vdots \\ \tilde{\mathbf{y}}_{s_b}^\top \end{bmatrix} = \begin{bmatrix} \phi_{1,1}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_1) & \dots & \phi_{1,s_b}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_{s_b}) \\ \vdots & & \vdots \\ \phi_{s_b,1}(\tilde{\mathbf{x}}_{s_b}, \tilde{\mathbf{x}}_1) & \dots & \phi_{s_b,s_b}(\tilde{\mathbf{x}}_{s_b}, \tilde{\mathbf{x}}_{s_b}) \end{bmatrix} [\omega_1, \dots, \omega_{n_y}] \quad (6)$$

$$\triangleq \bar{\mathbf{Y}} = \bar{\Phi} \bar{\Omega}$$

$$\phi_{j,i}(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_i) = \left( \frac{1}{(2\pi\gamma^2)^{\nu/2}} \right) e^{-\frac{\|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i\|^2}{\gamma^2}} \quad (7)$$

Similarly, the RBF surrogate model approximates the system's dynamic model by computing its optimal coefficient matrix. Since the RBF feature matrix is square, the optimal coefficient matrix is obtained through a direct inverse, given by  $\Omega^* = \bar{\Phi}^{-1} \bar{\mathbf{Y}}$ . Subsequently, the final RBF surrogate model is expressed as in (8), where  $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$  denotes the surrogate target vector. This result is obtained from the surrogate distance-dependent function vector  $\hat{\phi} = [\phi_1(\hat{\mathbf{x}}, \tilde{\mathbf{x}}_1), \dots, \phi_{s_b}(\hat{\mathbf{x}}, \tilde{\mathbf{x}}_{s_b})]^\top \in \mathbb{R}^{s_b}$ , which is computed by evaluating the distances between the  $s_b$  feature vectors from the identification dataset and the new surrogate feature vector  $\hat{\mathbf{x}} \in \mathbb{R}^\nu$ .

$$\hat{\mathbf{y}}^\top = \hat{\phi}^\top \Omega^* \quad (8)$$

### 2.1.3. Gaussian Process Regression

The third SM strategy tested is Gaussian Process Regression (GPR), which leverages probabilistic principles to model system behavior and uncertainties (Williams and Rasmussen 2006). GPR provides a flexible, non-parametric regression framework that offers not only predictions but also uncertainty estimates, making it particularly suitable for data-driven approximation and, hence, adaptive control applications. However, its computational cost can increase with data size. The GPR uses the information in the  $I_w$  dataset as probabilistic data within a Gaussian-distribution framework (Ebden

2015; Schulz, Speekenbrink, and Krause 2018). This approach begins by computing the Gram matrix  $\mathbf{K} \in \mathbb{R}^{s_b \times s_b}$ , as defined in (9), where each term  $\phi_{j,i}(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_i)$  is calculated using a Gaussian distribution-based function. To ensure consistency and enable a fair comparison, this work employs the same formulation as in (7). Consequently, the Gram matrix in GPR is equivalent to the feature matrix used in RBF. The key distinction lies in their interpretation: in GPR, the matrix is regarded as representing covariance information, whereas in RBF, it captures geometrical relationships between feature vectors.

$$\mathbf{K} = \begin{bmatrix} \phi_{1,1}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_1) & \dots & \phi_{1,s_b}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_{s_b}) \\ \vdots & \phi_{l,l}(\tilde{\mathbf{x}}_l, \tilde{\mathbf{x}}_l) & \vdots \\ \phi_{s_b,1}(\tilde{\mathbf{x}}_{s_b}, \tilde{\mathbf{x}}_1) & \dots & \phi_{s_b,s_b}(\tilde{\mathbf{x}}_{s_b}, \tilde{\mathbf{x}}_{s_b}) \end{bmatrix} \quad (9)$$

Once the Gram matrix is computed, the GPR strategy approximates each element  $\hat{y}_k, \forall k \in 1, \dots, n_y$ , of the surrogate target vector  $\hat{\mathbf{y}}$  as defined in (10), where the predicted value is obtained within a Gaussian distribution, characterized by a predictive mean  $\mu_k$  (11) and a predictive variance  $\sigma^2$  (12). To compute these predictive quantities, GPR requires the covariance vector between the surrogate feature vector  $\hat{\mathbf{x}}$  and each of the  $s_b$  feature vectors, given by  $\mathbf{k}_* = [\phi(\hat{\mathbf{x}}, \tilde{\mathbf{x}}_1), \dots, \phi(\hat{\mathbf{x}}, \tilde{\mathbf{x}}_{s_b})]^\top \in \mathbb{R}^{s_b}$ , along with the self-covariance  $k_{**} = \phi(\hat{\mathbf{x}}, \hat{\mathbf{x}})$ . Notably, the predictive mean  $\mu_k$  is computed using all the information gathered within  $I_w$ , specifically related to the  $k$ -th element of the dataset's target vector, i.e.,  $\check{\mathbf{y}}_k = [\tilde{y}_{1,k}, \dots, \tilde{y}_{l,k}, \dots, \tilde{y}_{s_b,k}]^\top \in \mathbb{R}^{s_b}$ . Remarkably, this approach can be extended to use multiple surrogate input vectors to predict multiple values of the  $\hat{y}_k$  surrogate output. Nonetheless, in a dynamic system, the temporal relationship between the system's inputs and outputs limits the GPR strategy to a single prediction at a time. It is noteworthy that the surrogate feature vector  $\hat{\mathbf{x}} \in \mathbb{R}^\nu$  (with  $\nu = n_x + n_u$ ) is used to predict all the  $k$ -th  $\forall k \in \{1, \dots, n_y\}$  elements of the surrogate target vector  $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$ . This characteristic makes it only necessary to compute each predictive  $\mu_k$  till the surrogate output vector  $\hat{\mathbf{y}}$  is completed.

$$\hat{y}_k \sim \mathcal{N}(\mu_k, \sigma^2) \quad (10)$$

$$\mu_k = \mathbf{k}_*^\top \mathbf{K}^{-1} \check{\mathbf{y}}_k \quad (11)$$

$$\sigma^2 = k_{**} - \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{k}_* \quad (12)$$

## 2.2. Surrogate-model-based Prediction Stage

The prediction stage is performed once the surrogate model is obtained using a supervised machine learning technique. This stage involves formulating the controller tuning problem as an optimization problem, as shown in (13), where  $\mathbf{k} \in \mathbb{R}^{n_k}$  denotes the design variable vector associated with the controller's parameters (as illustrated in Fig. 2). Additionally,  $\mathcal{J}_p$  denotes the objective functional associated with the performance

criterion to be optimized. Typically, this criterion is defined through a performance error function  $\varepsilon_P(\mathbf{r}, \hat{\mathbf{y}})|_{t \in P_w}$ , which evaluates the error between the controller's reference vector  $\mathbf{r}$  and the surrogate target vector  $\hat{\mathbf{y}}$ , within the prediction time window  $P_w$ . Any performance index, such as those listed in Table 2, can be used to define  $\varepsilon_P$ . However, alternative criteria may also be implemented depending on the application.

Returning to the optimization problem (13), it is noted that the formulation includes inequality constraints  $g_i(\mathbf{k}) \leq 0$ , equality constraints  $h_j(\mathbf{k}) = 0$ , and boundary limits defined by  $\mathbf{k}_{\min}$  and  $\mathbf{k}_{\max}$ . Furthermore, it is essential to highlight that the optimization problem is subject to the surrogate model relationship  $\hat{\mathbf{y}} = \hat{f}(\hat{\mathbf{x}}, \mathfrak{C}_o, \mathbf{k})$ , where  $\hat{f}$  denotes the surrogate modeling strategy employed. In the present study,  $\hat{f}$  is instantiated using RSM, RBF, or GPR, while the remaining prediction and optimization formulation is kept unchanged. This formulation provides the methodological basis for evaluating alternative surrogate-model strategies and lays the foundation towards a common surrogate-based adaptive controller-tuning framework. The coefficient term  $\mathfrak{C}_o$  represents the model-specific parameters, corresponding to the second-degree coefficient matrix  $\bar{\boldsymbol{\beta}}^*$  for RSM, the first-degree coefficient matrix  $\bar{\boldsymbol{\Omega}}^*$  for RBF, and the predictive mean  $\mu_k$  and variance  $\sigma^2$  for GPR.

$$\begin{aligned} & \min_{\mathbf{k} \in \mathbb{R}^{n_k}} \mathcal{J}_P \\ \text{s.t.} \quad & \begin{cases} \hat{\mathbf{y}} = \hat{f}(\hat{\mathbf{x}}, \mathfrak{C}_o, \mathbf{k}) \\ g_i(\mathbf{k}) \leq 0, \quad i = 1, \dots, r \\ h_j(\mathbf{k}) = 0, \quad j = 1, \dots, s \\ \mathbf{k}_{\min} \leq \mathbf{k} \leq \mathbf{k}_{\max} \end{cases} \end{aligned} \quad (13)$$

Table 2.: Popular Performance Indices.

| Indexes | Functional                                                      |
|---------|-----------------------------------------------------------------|
| IAE     | $\int_{t \in P_w}  \mathbf{r}(t) - \hat{\mathbf{y}}(t)  dt$     |
| ISE     | $\int_{t \in P_w} (\mathbf{r}(t) - \hat{\mathbf{y}}(t))^2 dt$   |
| ITAE    | $\int_{t \in P_w} t  \mathbf{r}(t) - \hat{\mathbf{y}}(t)  dt$   |
| ITSE    | $\int_{t \in P_w} t (\mathbf{r}(t) - \hat{\mathbf{y}}(t))^2 dt$ |

To solve the surrogate-model-based optimization problems (13), it is necessary to consider the intuitive aspects of SIACTA implementations. The solution to the optimization problem may vary at each update interval  $\Delta T_U$  because disturbances or uncertainties can alter the problem's conditions. Deterministic optimization algorithms are generally ineffective in this context, as the initial search point is unknown at each interval due to the system's dynamics. Moreover, gradient-based deterministic algorithms are limited by their requirement for continuous objective functions and constraints. Therefore, the optimization problem within SIACTA is addressed using bio-inspired algorithms in the present work. Such algorithms are suitable for the considered problem because they do not require an explicit dynamic model, can handle discrete and continuous search spaces, and operate over a defined search space without requiring a specific starting point. More generally, alternative optimization algorithms may also be incorporated into the prediction stage, provided that they can solve the surrogate-model-based optimization problem within the available update interval  $\Delta T_U$ .

Thus, the choice of optimizer is treated as an implementation-dependent component of the adaptive control tuning framework rather than as an intrinsic restriction of the surrogate-model formulation.

In this work, the online differential evolution (ODE) (Rodríguez-Molina et al. 2022) algorithm is employed as a fixed optimization component to isolate the influence of surrogate-model selection on the computational and control-performance outcomes. The online implementation involves incorporating the best solution from the current update period,  $\Delta T_U$ , into the initial population for the next update period,  $\Delta T_U + 1$ . This strategy assumes that the optimal solution does not experience significant shifts between consecutive intervals, thereby improving the algorithm’s exploration efficiency. It is important to clarify that the present work focuses exclusively on mono-objective optimization problems. Furthermore, this choice avoids the additional computational burden and *a posteriori* decision-making processes associated with multi-objective optimization, which typically require preference-handling mechanisms, as discussed in (Al-Tashi et al. 2020; Chen et al. 2020; Lin et al. 2021). Consequently, the first research step prioritizes a simplified mono-objective formulation suitable for unknown dynamic systems.

An additional consideration in time-based implementations is the occurrence of undesired fluctuations, remarkably abrupt changes in the solution of the optimization problem. To mitigate this effect, the present work incorporates a discrete-time implementation of a Resistor-Capacitor (RC) low-pass filter, expressed as  $\mathbf{k}(t + \Delta T_U) = d_f \mathbf{k}(t) + (1 - d_f) \mathbf{k}(t - \Delta T_U)$ . This filter is applied to the controller parameters  $\mathbf{k}$  at each update interval, following the approach in (Villarreal-Cervantes et al. 2017), where  $d_f$  denotes the discrete smoothing factor. This filtering strategy reduces undesired oscillations in system performance by smoothing transitions between consecutive controller parameter updates.

### 2.3. Summary of SIACTA Operation

Fig. 4 is provided to facilitate a clearer understanding of the SIACTA operation. It can be observed that two parallel threads are required. The first thread, referred to as the “closed-loop thread,” implements a traditional closed-loop control strategy with a sampling period  $\delta t_l$ . In addition to computing the control signal from the sampled system outputs, this thread collects system information into a dataset  $\mathcal{D}$  comprising  $s_b$  samples, including the system’s state, control signal, and output vectors. The second thread, referred to as the “adaptive thread,” operates at a slower update period  $\Delta T_U$  and is responsible for executing both the identification and prediction stages. The identification stage employs the collected dataset to compute the surrogate model, using any of the strategies presented in Section 2.1, where the central objective is to determine the optimal coefficient set  $\mathcal{C}_o$  for the surrogate model function  $\hat{f}(\cdot)$ . Once the surrogate model is obtained, it is used to formulate the optimization problem in the prediction stage, to determine the new controller parameter vector that optimizes the desired criterion, evaluated through the prediction objective function  $\mathcal{J}_P$ . The optimal controller parameters obtained from the prediction stage are then filtered and applied in the closed-loop thread.

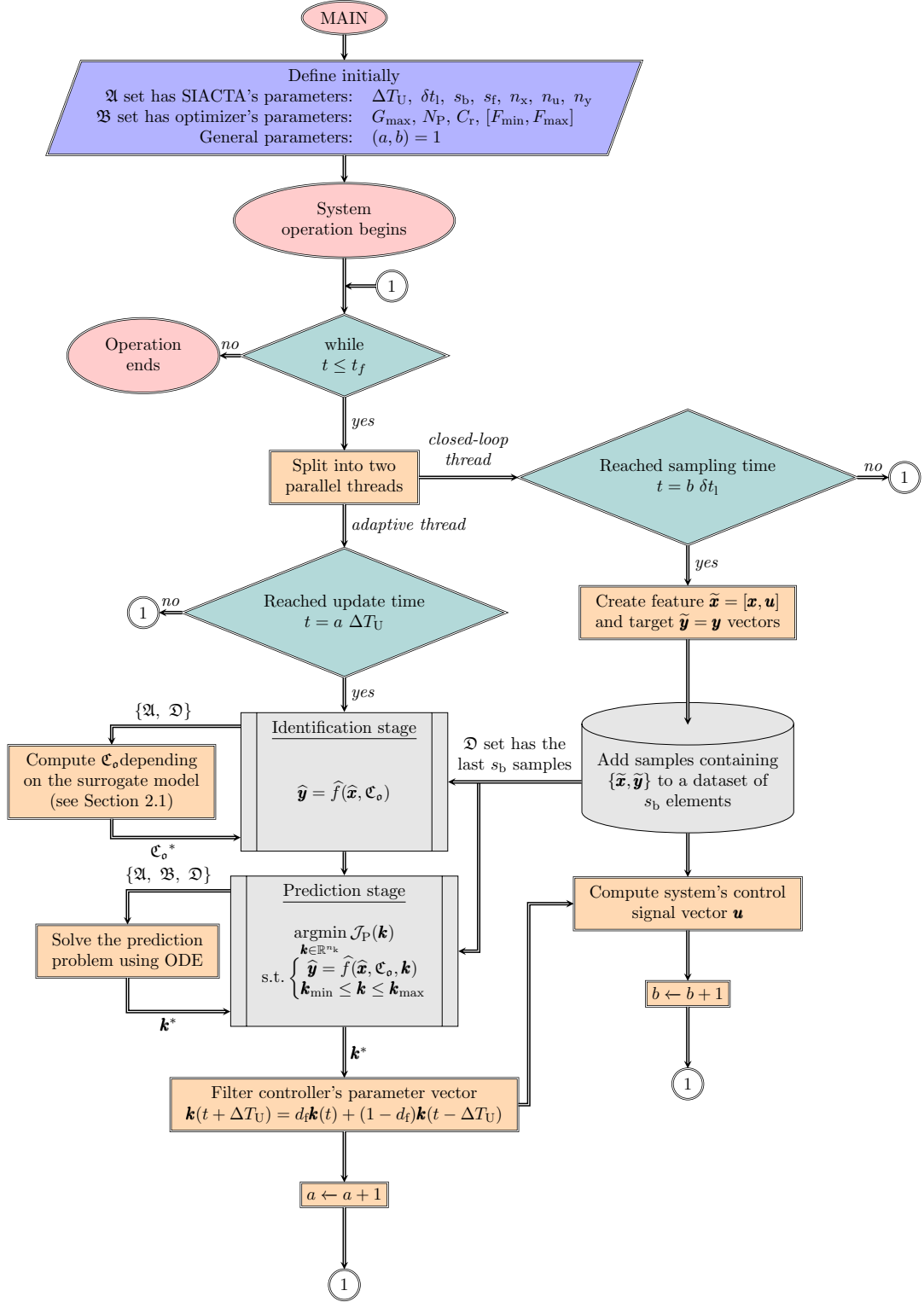


Figure 4.: SIACTA operation summary.

## 2.4. SIACTA's Characteristics and Considerations

The following information is provided to assist researchers and practitioners in understanding and facilitating the implementation of SIACTA using any of the previously described methods:

- *General information of SIACTA*
  - To implement the SIACTA framework and instantiate its workflow in a given mechatronic system, the required information is summarized in Table 3. This information covers system-level dimensions and implementation parameters, including the numbers of state and output variables, the dimension of the control signal parameter, and the timing constraints for sampling, identification, prediction, and optimization. By leveraging these generic dimensions, the proposal provides a common framework for integrating and evaluating different surrogate-model approaches without redesigning the core strategy. Consequently, this formulation facilitates adaptation to alternative application settings. However, the suitability and real-time feasibility of a particular implementation depend on the target system's characteristics and require application-specific validation.

Table 3.: General parameters for SIACTA implementation.

|                                      | Concerning                             | Parameter                               |
|--------------------------------------|----------------------------------------|-----------------------------------------|
| System characteristics               | Size of system's states vector         | $n_x$                                   |
|                                      | Size of system's control signal vector | $n_u$                                   |
|                                      | Size of system's output vector         | $n_y$                                   |
| Time-related parameters              | Sampling period                        | $\delta t_l$                            |
|                                      | Update period <sup>a</sup>             | $\Delta T_U$                            |
|                                      | Backward steps                         | $s_b$                                   |
|                                      | Forward steps                          | $s_f$                                   |
|                                      | Identification time window             | $I_w$                                   |
|                                      | Prediction time window                 | $P_w$                                   |
| Optimization problem characteristics | Size of controller's parameter vector  | $n_k$                                   |
|                                      | Metaheuristic optimizer parameters     | Population size, Max. generations, etc. |

- Regardless of the surrogate modeling strategy, all approaches rely exclusively on the dataset information collected during the identification time window  $I_w$ . This dataset includes the system's states, outputs, and control signals. Consequently, the surrogate model can only simulate or predict the behavior of these variables, and any other state, output, or control signal not included during the identification stage cannot be predicted.
- Since SIACTA uses a fixed sampling period  $\delta t_l$  to define the temporal relationship between samples, the surrogate model must employ the same sampling period  $\delta t_l$  for forward-step simulations within the prediction stage.
- Alternative optimization algorithms beyond the one used in this work must solve the surrogate-model-based optimization problem for the prediction stage within the update period  $\Delta T_U$ . Although SIACTA reduces the computational burden, the optimizer's suitability depends on the algorithm's computational complexity and the computational system's resources to handle the load.
- If the sampling period  $\delta t_l$  is too short, it may lead to redundant or uninformative data.

mative samples, as parametric uncertainties and external disturbances may not significantly affect the system within such short intervals. Conversely, if  $\delta t_1$  is too long, relevant system information may be omitted, and adverse conditions might go uncompensated. The choice of  $\delta t_1$  should reflect the system's dynamic responses, satisfying at least the Nyquist-Shannon sampling criterion, where fast-dynamic systems require shorter sampling periods and slow-dynamic systems allow more extended periods to reconstruct the model.

- The number of backward steps  $s_b$ , which defines the identification time window  $I_w$ , must be sufficient to ensure representativeness of the dataset. However, increasing  $s_b$  increases computational costs during surrogate model construction, introducing a trade-off between data representativeness and computational efficiency.
- The number of forward steps  $s_f$ , which defines the prediction time window  $P_w$ , must be adequate to allow the optimization algorithm to differentiate between candidate solutions. Nonetheless, a larger  $P_w$  increases the computational load during the prediction stage.
- It is advisable to conduct a sensitivity analysis on the time-related parameters  $\delta t_1$ ,  $\Delta T_U$ ,  $I_w$ , and  $P_w$ , as their selection directly influences both performance and computational demand. This sensitivity analysis should consider the computational capabilities of the target system.
- *SIACTA with response surface method (SIACTA-RSM)*
  - Any polynomial degree can be employed within the RSM; however, it is important to consider that increasing the polynomial degree directly increases the computational burden associated with model construction and evaluation.
  - Since the RSM formulation reduces to solving a system of equations, the number of samples  $s_b$  must be greater than or equal to the number of polynomial terms  $\nu$ , that is,  $s_b \geq \nu$ . Specifically, in the context of RSM,  $\nu$  is computed as  $\nu = \frac{(n_x + n_u + 1)(n_x + n_u + 2)}{2}$ , accounting for all bias, first-degree, and second-degree terms.
- *SIACTA with radial basis function (SIACTA-RBF)*
  - The RBF method does not require a strict minimum number of samples  $s_b$  to perform the regression; however, it is recommended to use at least  $s_b > 2$  to ensure sufficient information sharing among samples (Fasshauer 2007). As previously discussed, a small number of samples leads to poor regression accuracy, while an excessive number of samples increases the computational burden.
  - The method is sensitive to the scale factor  $\gamma$  of the Gaussian kernel. A small  $\gamma$  value tends to produce localized “bumps” in the regression surface, while a large  $\gamma$  value causes the regression to approximate a smooth, flat plane.
- *SIACTA with Gaussian process regression (SIACTA-GPR)*
  - The GPR method does not provide a strict lower bound on the number of samples  $s_b$  required for regression. However, it is generally advisable to use  $s_b > 2$  to ensure a meaningful estimation of covariance relationships among samples (Fasshauer 2007). A reduced number of samples limits the predictive capacity of GPR, while increasing  $s_b$  improves model generalization at the cost of greater computational effort due to the need for matrix inversions.
  - The GPR performance is highly influenced by the scale factor  $\gamma$ , which

directly impacts the covariance structure. A minimal  $\gamma$  results in rapidly decaying covariances, leading to overfitting to isolated data points and reducing the generalization capability. In contrast, a high  $\gamma$  induces dominant long-range correlations, potentially leading to overly smoothed predictions with reduced sensitivity to local dynamics.

### 3. Case study: BLDC motor

The present work considers a brushless direct current (BLDC) motor as the case study, selected for its widespread use in industrial applications and inherent control complexity. This complexity arises from its fast-response behavior, which makes it difficult to compensate for adverse conditions such as parametric uncertainties and external disturbances.

#### 3.1. Dynamic model

Fig. 5 presents the schematic of a BLDC motor, which includes both the three-phase motor and the inverter. The inverter's commutation logic converts the DC voltage input into a quasi-sinusoidal three-phase AC voltage to drive the motor.

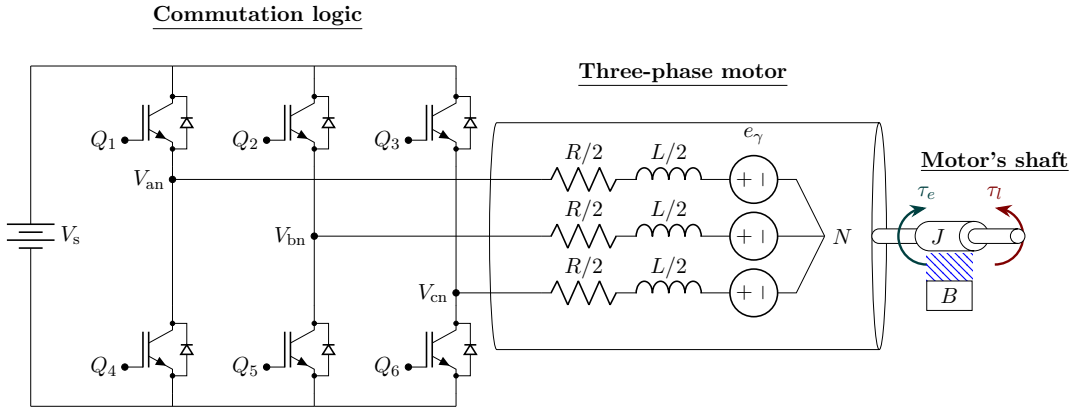


Figure 5.: BLDC motor diagram.

The dynamic model of the BLDC motor comprises both electrical and mechanical components. The electrical model describes the energy flow through each of the  $\gamma \in \{a, b, c\}$  phases, as defined in (14). In this model,  $i_\gamma$  and  $V_{\gamma n} = \frac{V_s}{2}\eta_\gamma$  denote the phase current and phase voltage, respectively, where  $V_s$  is the DC voltage supply applied to the three-phase inverter, and  $\eta_\gamma$  is a position-dependent function. The parameters  $R$  and  $L$  represent the line (phase-to-phase) resistance and inductance, respectively. Additionally,  $e_\gamma = \frac{k_e}{2}v\rho_\gamma$  defines the back-electromotive force (back-EMF) of the  $\gamma$  phase, where  $k_e$  is the electromagnetic constant,  $v$  is the angular velocity of the BLDC motor, and  $\rho_\gamma$  is a position-dependent function. The functions  $\eta_\gamma$  and  $\rho_\gamma$  account for the phase shift of  $\frac{2\pi}{3}$  between phases, such that  $\eta_a = \eta(\theta)$ ,  $\rho_a = \rho(\theta)$ ,  $\eta_b = \eta(\theta - \frac{2\pi}{3})$ ,  $\rho_b = \rho(\theta - \frac{2\pi}{3})$ , and  $\eta_c = \eta(\theta - \frac{4\pi}{3})$ ,  $\rho_c = \rho(\theta - \frac{4\pi}{3})$ , where  $\eta(\theta)$  and  $\rho(\theta)$  are defined in (15) and (16), respectively.

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \begin{bmatrix} L/2 & 0 & 0 \\ 0 & L/2 & 0 \\ 0 & 0 & L/2 \end{bmatrix} \begin{bmatrix} \dot{i}_a \\ \dot{i}_b \\ \dot{i}_c \end{bmatrix} + \begin{bmatrix} R/2 & 0 & 0 \\ 0 & R/2 & 0 \\ 0 & 0 & R/2 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (14)$$

$$\eta(\theta) = \begin{cases} 1, & \text{if } 0 \leq \theta < \frac{2\pi}{3} \\ 0, & \text{if } \frac{2\pi}{3} \leq \theta < \pi \\ -1, & \text{if } \pi \leq \theta < \frac{5\pi}{3} \\ 0, & \text{if } \frac{5\pi}{3} \leq \theta < 2\pi \end{cases} \quad (15)$$

$$\rho(\theta) = \begin{cases} 1, & \text{if } 0 \leq \theta < \frac{2\pi}{3} \\ 1 - \frac{6}{\pi} \left( \theta - \frac{2\pi}{3} \right), & \text{if } \frac{2\pi}{3} \leq \theta < \pi \\ -1, & \text{if } \pi \leq \theta < \frac{5\pi}{3} \\ -1 + \frac{6}{\pi} \left( \theta - \frac{5\pi}{3} \right), & \text{if } \frac{5\pi}{3} \leq \theta < 2\pi \end{cases} \quad (16)$$

The electric flow described in (14) generates an electromagnetic torque  $\tau_e$ , defined in (17), where  $k_t$  is the mechanical constant. This electromagnetic torque  $\tau_e$  produces mechanical motion that drives an external load  $\tau_l$  through the motor's shaft. The interaction between the electromagnetic torque and the external load is governed by (18) and (19), where  $J$  and  $B$  denote the shaft's inertia and viscous friction coefficients, respectively, and  $P$  represents the number of pole pairs.

$$\tau_e = k_t (i_a \rho_a + i_b \rho_b + i_c \rho_c) \quad (17)$$

$$\tau_e - \tau_l = J \frac{dv}{dt} + Bv \quad (18)$$

$$\frac{d\theta}{dt} = Pv \quad (19)$$

The phase model is transformed into a line (phase-to-phase) model by considering the current relationships  $i_c = -i_a - i_b$ , and the line voltages defined as  $V_{ab} = V_{an} - V_{bn}$ ,  $V_{bc} = V_{bn} - V_{cn}$ , and  $V_{ca} = V_{cn} - V_{an}$ . This transformation is applied since, in many real-world BLDC motor systems, direct phase-to-neutral measurements are not feasible. The complete dynamic model of the BLDC motor is expressed in (20), where the state vector is defined as  $\mathbf{x} = [x_1, x_2, x_3, x_4]^\top = [i_a, i_b, v, \theta]^\top \in \mathbb{R}^4$ , the control signal is  $u = V_s$ , and the parameter vector is  $\mathbf{p} = [p_1, \dots, p_6]^\top = [\frac{R}{L}, \frac{k_e}{6L}, \frac{k_t}{J}, \frac{B}{J}, \frac{1}{6L}, P]^\top \in \mathbb{R}^6$ . The output vector is  $\mathbf{y} \in \mathbb{R}^4$ , with  $\mathbf{C} = \mathbf{I}_4 \in \mathbb{R}^{4 \times 4}$  representing the identity matrix, assuming all states are measurable. Finally,  $\mathbf{D} = [0, 0, 1, 0]^\top \in \mathbb{R}^4$  denotes the external load vector applied to the BLDC motor.

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{D}\tau_l \\
&= f(\mathbf{x}, u, \mathbf{p}) \\
\mathbf{y} &= \mathbf{C}\mathbf{x}
\end{aligned} \tag{20}$$

where,

$$\mathbf{A} = \begin{bmatrix} -p_1 & 0 & -p_2(2 * \rho_a - \rho_b - \rho_c) & 0 \\ 0 & -p_1 & p_2(\rho_a - 2\rho_b + \rho_c) & 0 \\ p_3(\rho_a - \rho_c) & p_3(\rho_b - \rho_c) & -p_4 & 0 \\ 0 & 0 & p_6 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} p_5(2\eta_a - \eta_b - \eta_c) \\ p_5(-\eta_a + 2\eta_b - \eta_c) \\ 0 \\ 0 \end{bmatrix}.$$

In the current work, the control signal  $u(t)$  is generated using a Proportional-Integral (PI) controller, defined as  $u(t) = k_p e(t) + k_i \int e(t) dt$ , where  $e(t) = v_d - x_3$  represents the speed regulation error, given by the difference between the desired angular speed  $v_d$  and the actual angular speed  $x_3$ . The proportional and integral gains,  $k_p$  and  $k_i$ , respectively, form the controller's parameter vector  $\mathbf{k} = [k_p, k_i]^\top \in \mathbb{R}^2$ .

### 3.2. Experimental platform description

Fig. 6 illustrates the scheme of the experimental platform, detailing its main blocks. The Mechanical block consists of a BLDC motor (Maxon EC 90 flat, model 607327) and an unbalanced weight system. The nominal specifications of the motor are provided in Table 4. The unbalanced weight system connects the BLDC motor shaft to a transfer mechanism composed of a gear-timing belt assembly with a 1:1 ratio. The control objective is to regulate the BLDC motor's angular speed, considering unknown rotational inertia and timing belt stiffness as parametric uncertainties. The BLDC motor is powered (brown lines in Fig. 6) by a DC-to-AC voltage inverter (MAXON ESCON 50/5). The Electronic block is located between the voltage inverter and the BLDC motor, where the three-phase currents and phase-to-phase voltages are sensed and conditioned. Current measurements are performed using LEM LTSR-6-NP sensors, and voltage measurements are performed using LEM LV-25-P sensors. All signals are filtered through a second-order Sallen-Key low-pass filter with a cut-off frequency of 1 kHz. The conditioned current and voltage signals (blue lines in Fig. 6), along with the BLDC motor's encoder and Hall-effect signals (green lines), are acquired by the Control block. This block consists of a Sensoray 826 PCIe multifunction analog/digital I/O board, interfaced with a computer running at 3.2 GHz and equipped with 16 GB of RAM. The control block executes the SIACTA approach, implemented in C++ with Visual Studio 2021, and generates the regulated control signal (orange line in Fig. 6) that is sent to the voltage inverter. This configuration is flexible, allowing alternative control strategies to be implemented if needed.

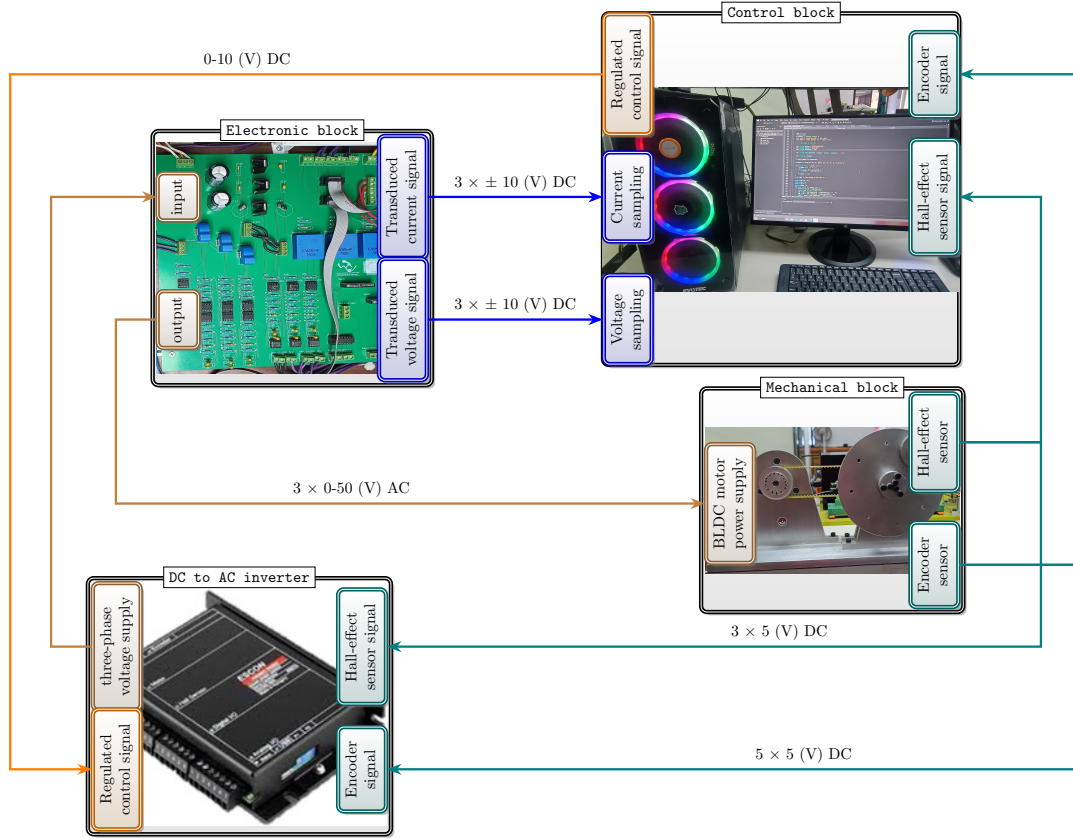


Figure 6.: Experimental platform scheme.

Table 4.: BLDC motor's nominal values.

| Parameter     | Name                          | Value                         |
|---------------|-------------------------------|-------------------------------|
| $R$           | Line resistance               | 0.844 ( $\Omega$ )            |
| $L$           | Line inductance               | 0.00107 (H)                   |
| $k_e$         | Electromagnetic constant      | 0.207 (V·s/rad)               |
| $k_t$         | Torque constant               | 0.231 (N·m/A)                 |
| $B$           | Magnetic friction coefficient | 0.000181437 (N·m·s)           |
| $J$           | Inertia coefficient           | 0.000506 (kg·m <sup>2</sup> ) |
| $P$           | Pairs of poles                | 11                            |
| $\tau_{\max}$ | Maximum continuous torque     | 0.964 (Nm)                    |

## 4. Experimentation

### 4.1. Considerations

According to the BLDC motor dynamic model (20), the following dimensions are established:  $n_x = 4$ ,  $n_u = 1$ , and  $n_y = 4$ , corresponding to the sizes of the system's state, control signal, and output vectors, respectively. Table 5 summarizes the time-related parameters and the optimizer parameters. The time-related parameters are selected based on a sensitivity analysis that accounts for the computational capabilities of the experimental platform, choosing the smallest feasible values that ensure proper system operation. Regarding the optimization stage, the online differential evolution (ODE) algorithm is employed, specifically in its best/1/bin variant. The online adaptation incorporates the last known optimal individual into the initial population for the current search cycle, thereby facilitating the algorithm's exploration. The optimizer and its parameters are kept fixed across all surrogate-model variants to isolate the effect of surrogate-model selection on computational burden and closed-loop performance. The optimizer parameters are adopted from (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2023; Rojas-López et al. 2025), as they have shown effectiveness in adaptive controller tuning problems.

Table 5.: SIACTA implementation parameters.

|                                      | Parameter                              | value                               |
|--------------------------------------|----------------------------------------|-------------------------------------|
| System characteristics               | Size of system's states vector         | $n_x = 4$                           |
|                                      | Size of system's control signal vector | $n_u = 1$                           |
|                                      | Size of system's output vector         | $n_y = 4$                           |
| Time-related parameters              | Sampling period                        | $\delta t_l = 100$ ( $\mu$ s)       |
|                                      | Update period <sup>a</sup>             | $\Delta T_U = 10$ (ms)              |
|                                      | Backward steps                         | $s_b = 100$                         |
|                                      | Forward steps                          | $s_f = 10$                          |
|                                      | Identification time window             | $I_w = 10$ (ms)                     |
|                                      | Prediction time window                 | $P_w = 500$ ( $\mu$ s)              |
| Optimization problem characteristics | ODE's Population size                  | $N_P = 20$                          |
|                                      | ODE's Max. generations                 | $G_{\max} = 20$                     |
|                                      | ODE's Crossover rate                   | $C_r = 0.6$                         |
|                                      | ODE's Mutation factor range            | $[F_{\min}, F_{\max}] = [0.3, 0.9]$ |
|                                      | Size of controller's parameter vector  | $n_k = 2$                           |

<sup>a</sup>SIACTA-GPR requires  $\Delta T_U = 250$  (ms) since its process is significantly slower.

The prediction stage optimization problem employed in this work is defined in (21), where the objective functional  $\mathcal{J}_P = \int_{t \in P_w} |v_d(t) - \hat{y}_3(t)| dt$  represents the integral absolute error (IAE) between the desired angular speed  $v_d$  and the surrogate model's predicted angular speed  $\hat{y}_3$  within the prediction time window  $P_w$ . The selected controller is a proportional-integral (PI) controller, characterized by the parameter vector  $\mathbf{k} = [k_p, k_i]^\top \in \mathbb{R}^2$ . The optimization problem is bounded by lower and upper limits  $\mathbf{k}_{\min} = 0.9 \mathbf{k}_0$  and  $\mathbf{k}_{\max} = 1.1 \mathbf{k}_0$ , respectively, where  $\mathbf{k}_0$  is a reference parameter vector precomputed offline under ideal operating conditions.

$$\begin{aligned} & \min_{\mathbf{k} \in \mathbb{R}^{n_2}} \mathcal{J}_P \\ \text{s.t. } & \begin{cases} \hat{\mathbf{y}} = \hat{f}(\hat{\mathbf{x}}, \mathcal{C}_o, \mathbf{k}) \\ \mathbf{k}_{\min} \leq \mathbf{k} \leq \mathbf{k}_{\max} \end{cases} \end{aligned} \quad (21)$$

## 4.2. Results

The SIACTA-RSM, SIACTA-RBF, and SIACTA-GPR variants are compared across three experiments, with 30 executions for each variant in each experiment. The comparison also includes the IACTA strategy (Rojas-López, Villarreal-Cervantes, and Rodríguez-Molina 2023) to assess computational burden and the optimal offline controller tuning approach (OOCTA) (Paredes-Ballesteros et al. 2025) as a baseline reference for control performance. In the first experiment, no weights are attached to the disc. Although the disc remains balanced, parametric uncertainties persist due to the timing belt's elongation. The second and third experiments introduce unbalanced weights to the disc: the second experiment involves moving a pair of 0.077 kg discs. In comparison, the third experiment involves a pair of 0.099 kg discs (see Fig. 7). For each control strategy, thirty executions of duration  $t_f = 10$  s are performed.

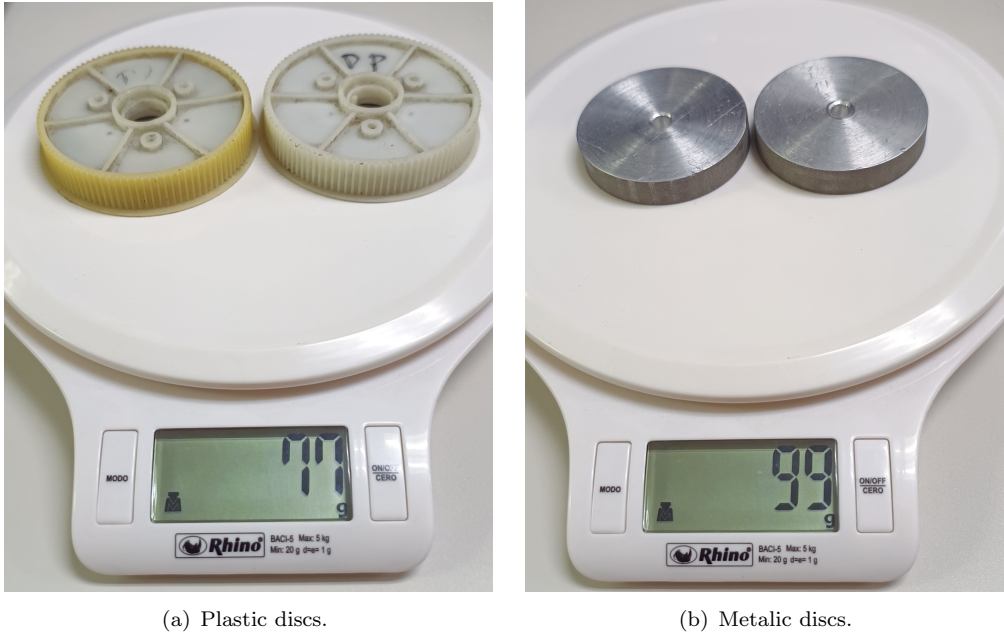


Figure 7.: Used external weights.

In each execution, three evaluation criteria are considered. The first criterion assesses system performance, measured by IAE over the entire execution period. The second criterion quantifies the computational burden by evaluating the average computational time (ACT) required by each control strategy to complete the stages within an update period  $\Delta T_U$ . The third criterion analyzes the normalized trade-off (NTO) between system performance and computational burden, providing a combined perspective of efficiency and control effectiveness. The next information is provided in order to clarify the NTO criterion. The NTO criterion for the  $i$ -th execution, named  $NTO_i$ , is computed through (22), where the terms  $IAE_i^m$  and  $ACT_i^m$  are the IAE and ACT evaluations of the respective  $m$ -th controller tuning approach, and  $IAE_{\min}$ ,  $ACT_{\min}$ ,  $IAE_{\max}$ , and  $ACT_{\max}$  are the respective minimal and maximal evaluations among all the implemented controller tuning approaches.

$$\text{NTO}_i = \sqrt{\left(\frac{\text{IAE}_i^m - \text{IAE}_{\min}}{\text{IAE}_{\max} - \text{IAE}_{\min}}\right)^2 + \left(\frac{\text{ACT}_i^m - \text{ACT}_{\min}}{\text{ACT}_{\max} - \text{ACT}_{\min}}\right)^2} \quad (22)$$

#### 4.2.1. Graphical overview

Fig. 8 presents the results of each experiment, illustrating the execution closest to the mean performance among the thirty runs for each control strategy. It is observed that the SIACTA-RBF variant achieves faster compensation across all three experiments. Additionally, it exhibits lower overshoot during the system's start-up phase compared with the other control strategies implemented. Beyond these observations, further distinctions between the control strategies are not immediately evident from the plotted results.

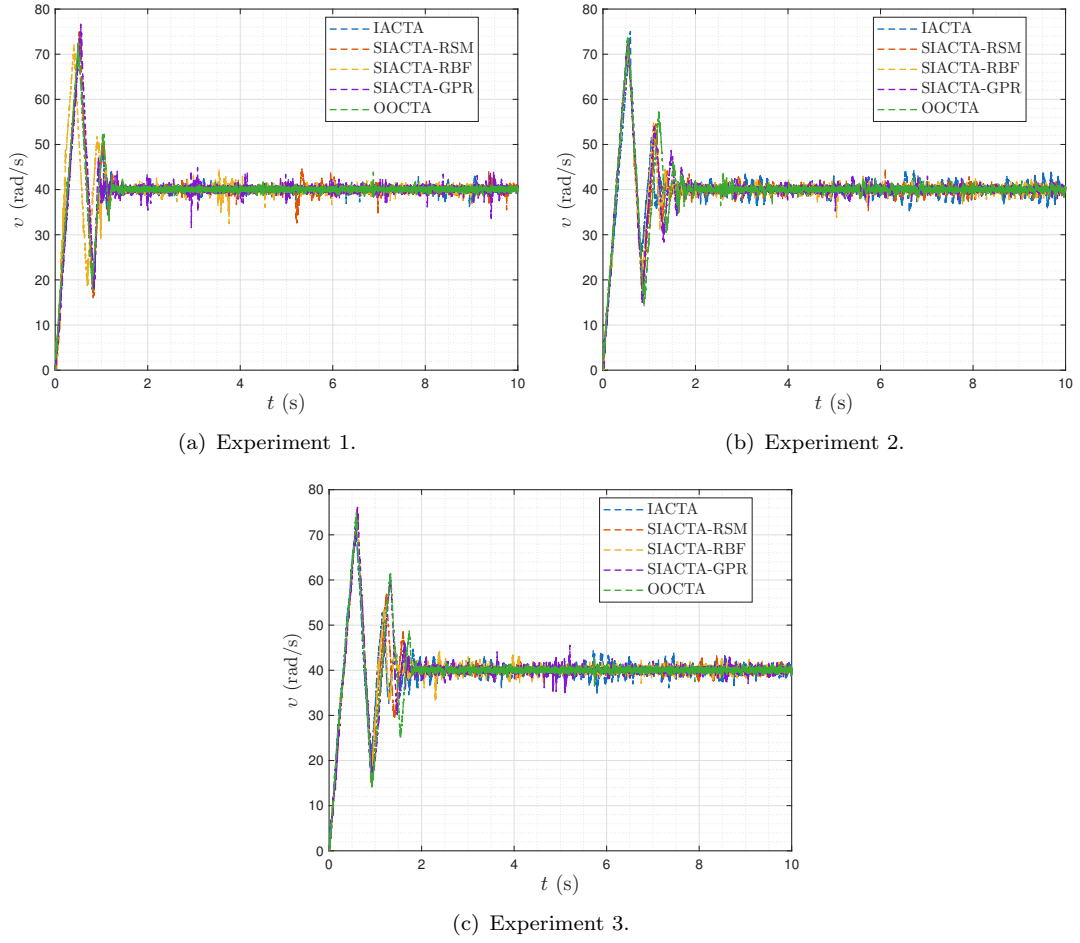


Figure 8.: Speed regulation outcomes.

#### 4.2.2. Statistical analysis

Tables 6, 7, and 8 present the statistical outcomes, both descriptive and inferential, corresponding to the first, second, and third experiments, respectively. Each table

is organized into three sections of rows, where the first, second, and third sections report the results for the IAE, ACT, and NTO criteria, respectively. The second column identifies the control tuning strategy evaluated for each criterion. The third to sixth columns report the best, mean, worst, and standard deviation values across the thirty executions for each strategy. The seventh to eleventh columns (shaded in grey) summarize the results of the Wilcoxon rank-sum tests, performed with a significance level of  $\alpha = 0.05$ . The symbols  $+$ ,  $-$ , and  $\approx$  indicate whether the strategy in the row label is statistically superior, inferior, or statistically similar to the strategy in the column header. It is essential to note that the ACT and NTO criteria are not evaluated for OOctA (indicated as n.a.), as OOctA operates offline. Additionally, the symbol “—” is used to denote pairwise comparisons of a control tuning strategy with itself.

The following highlights are provided to facilitate a clearer understanding of the obtained results:

- Experiment 1:
  - *Regarding IAE*: All tested controller tuning strategies yield statistically similar outcomes, indicating comparable control performance. However, the SIActA-RBF variant is the only adaptive strategy that achieves a lower mean IAE than the traditional offline strategy, OOctA, reducing it by 4.70%.
  - *Regarding ACT*: SIActA-RSM shows the lowest computational burden among the adaptive strategies, outperforming IActA, SIActA-GPR, and SIActA-RBF, followed by SIActA-RBF as the second-best. Specifically, SIActA-RSM achieves a mean ACT that is 43.5 times faster than IActA, while SIActA-RBF achieves a mean ACT up to 1.5 times faster than IActA. Notably, SIActA-GPR results in the highest computational burden, confirming that not all surrogate model strategies necessarily reduce computational cost.
  - *Regarding NTO*: SIActA-RBF and SIActA-RSM exhibit statistically similar trade-off performance compared to the traditional adaptive strategy IActA. Nevertheless, SIActA-RBF achieves a notable 25.67% reduction in the mean NTO value relative to IActA.
- Experiment 2:
  - *Regarding IAE*: The SIActA-RBF and SIActA-RSM proposals, along with the traditional IActA strategy, outperform the conventional offline strategy, OOctA. In contrast, SIActA-GPR remains statistically similar to OOctA. It is noteworthy that the performance of SIActA-RSM decreases as experimental complexity increases, and it is outperformed by SIActA-RBF and IActA, which remain statistically similar. Furthermore, SIActA-RBF achieves a 2.62% reduction in the mean IAE compared to IActA.
  - *Regarding ACT*: The outcomes for this criterion remain consistent with those observed in the first experiment. This is expected, as the computational burden is associated with the internal operations of each controller tuning strategy, specifically the identification and prediction stages, rather than with the external experimental conditions. Therefore, ACT results remain constant for the BLDC motor regardless of experiment complexity.
  - *Regarding NTO*: The NTO results follow a similar trend to those observed in the first experiment. This indicates that while SIActA-RBF and IActA improve in performance with increasing experiment complexity, the sub-

stantial computational burden reduction achieved by SIACTA-RSM remains sufficient to maintain a competitive trade-off.

- Experiment 3:
  - *Regarding IAE*: As the experimental complexity increases, SIACTA-RBF emerges as the most reliable strategy, outperforming all other approaches. Specifically, SIACTA-RBF reduces the mean IAE by 3.34% compared to IACTA.
  - *Regarding ACT*: The established trend persists, where SIACTA-RSM consistently achieves the lowest computational burden, followed by SIACTA-RBF. Both strategies outperform IACTA, while SIACTA-GPR continues to yield the worst computational cost.
  - *Regarding NTO*: SIACTA-RBF outperforms all other adaptive strategies, reducing IACTA's mean NTO by 10.33%. Notably, as experiment complexity increases, the traditional IACTA strategy shows a decline in overall performance. Consequently, SIACTA-RSM, due to its significantly reduced computational burden, achieves NTO results that are statistically comparable to those of IACTA.

Table 6.: Experiment 1 - statistical results.

| Crit. | Strategy   | best           | mean           | worst          | std.          | IACTA | SIACTA-RSM | SIACTA-RBF | SIACTA-GPR | OOCTA |
|-------|------------|----------------|----------------|----------------|---------------|-------|------------|------------|------------|-------|
| IAE   | IACTA      | 18.2739        | 23.4214        | 27.4358        | 2.0202        | —     | ≈          | ≈          | ≈          | ≈     |
|       | SIACTA-RSM | 20.4474        | 24.2730        | 38.3512        | 3.7443        | ≈     | —          | —          | ≈          | ≈     |
|       | SIACTA-RBF | <b>14.5471</b> | <b>21.7215</b> | 27.4732        | 3.2299        | ≈     | +          | —          | ≈          | ≈     |
|       | SIACTA-GPR | 18.2848        | 23.3233        | 33.8894        | 3.0709        | ≈     | ≈          | ≈          | —          | ≈     |
|       | OOCTA      | 20.0551        | 22.7946        | <b>25.5562</b> | <b>1.4646</b> | ≈     | ≈          | ≈          | ≈          | —     |
| ACT   | IACTA      | 10.8028        | 11.0088        | 12.5031        | 0.3065        | —     | —          | —          | +          | n.a.  |
|       | SIACTA-RSM | <b>0.2344</b>  | <b>0.2527</b>  | <b>0.2665</b>  | <b>0.0070</b> | +     | —          | +          | +          | n.a.  |
|       | SIACTA-RBF | 5.6893         | 7.2979         | 10.2021        | 0.9428        | +     | —          | —          | +          | n.a.  |
|       | SIACTA-GPR | 181.7090       | 217.8759       | 271.7040       | 17.7257       | —     | —          | —          | —          | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |
| NTO   | IACTA      | 0.1617         | 0.3751         | <b>0.5429</b>  | <b>0.0842</b> | —     | ≈          | ≈          | +          | n.a.  |
|       | SIACTA-RSM | 0.2479         | 0.4086         | 1.0000         | 0.1573        | ≈     | —          | —          | +          | n.a.  |
|       | SIACTA-RBF | <b>0.0268</b>  | <b>0.3037</b>  | 0.5435         | 0.1329        | ≈     | +          | —          | +          | n.a.  |
|       | SIACTA-GPR | 0.7654         | 0.8891         | 1.1378         | 0.0931        | —     | —          | —          | —          | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |

### 4.3. Discussion

Three experiments were conducted, each introducing increasing levels of adverse conditions to be compensated for. Three surrogate-model-based indirect adaptive controller tuning strategies were evaluated: response surface method (SIACTA-RSM), radial basis function (SIACTA-RBF), and Gaussian process regression (SIACTA-GPR). These were compared with the traditional indirect adaptive controller tuning approach (IACTA), which is based on a generalized dynamic model, and the conventional optimal offline controller tuning approach (OOCTA). The evaluation considered three criteria: system performance, measured by the integral absolute error (IAE); the av-

Table 7.: Experiment 2 - statistical results.

| Crit. | Strategy   | best           | mean           | worst          | std.          | IACTA | SIACTA-RSM | SIACTA-RBF | SIACTA-GPR | OOCTA |
|-------|------------|----------------|----------------|----------------|---------------|-------|------------|------------|------------|-------|
| IAE   | IACTA      | 21.7815        | 25.3688        | 30.3091        | 2.2372        | --    | ≈          | ≈          | ≈          | +     |
|       | SIACTA-RSM | 20.9865        | 26.0983        | 30.7435        | <b>2.0758</b> | ≈     | --         | --         | ≈          | +     |
|       | SIACTA-RBF | 17.4905        | <b>24.7038</b> | <b>30.0957</b> | 2.7996        | ≈     | +          | --         | +          | +     |
|       | SIACTA-GPR | <b>16.8738</b> | 26.6662        | 43.0334        | 4.4321        | ≈     | ≈          | --         | --         | ≈     |
|       | OOCTA      | 22.7976        | 27.7980        | 34.7883        | 2.5046        | --    | --         | --         | ≈          | --    |
| ACT   | IACTA      | 10.7932        | 10.9060        | 11.2521        | 0.1097        | --    | --         | --         | +          | n.a.  |
|       | SIACTA-RSM | <b>0.2270</b>  | <b>0.2485</b>  | <b>0.2838</b>  | <b>0.0124</b> | +     | --         | +          | +          | n.a.  |
|       | SIACTA-RBF | 5.9373         | 6.6635         | 8.3481         | 0.6085        | +     | --         | --         | +          | n.a.  |
|       | SIACTA-GPR | 183.2340       | 208.0580       | 238.3270       | 15.7367       | --    | --         | --         | --         | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |
| NTO   | IACTA      | 0.1930         | 0.3280         | 0.5156         | 0.0846        | --    | ≈          | ≈          | +          | n.a.  |
|       | SIACTA-RSM | 0.1572         | 0.3526         | 0.5302         | <b>0.0794</b> | ≈     | --         | --         | +          | n.a.  |
|       | SIACTA-RBF | <b>0.0336</b>  | <b>0.3009</b>  | <b>0.5061</b>  | 0.1060        | ≈     | +          | --         | +          | n.a.  |
|       | SIACTA-GPR | 0.8448         | 0.9625         | 1.2915         | 0.0886        | --    | --         | --         | --         | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |

Table 8.: Experiment 3 - statistical results.

| Crit. | Strategy   | best           | mean           | worst          | std.          | IACTA | SIACTA-RSM | SIACTA-RBF | SIACTA-GPR | OOCTA |
|-------|------------|----------------|----------------|----------------|---------------|-------|------------|------------|------------|-------|
| IAE   | IACTA      | 21.5339        | 28.1475        | <b>31.4814</b> | <b>2.4408</b> | --    | ≈          | --         | +          | +     |
|       | SIACTA-RSM | 23.1762        | 28.5739        | 34.6438        | 3.0792        | ≈     | --         | --         | +          | +     |
|       | SIACTA-RBF | <b>19.1572</b> | <b>27.2048</b> | 40.3387        | 3.3354        | +     | +          | --         | +          | +     |
|       | SIACTA-GPR | 26.1260        | 30.4315        | 38.9564        | 2.9086        | --    | --         | --         | --         | ≈     |
|       | OOCTA      | 27.1614        | 30.5160        | 46.2315        | 3.6386        | --    | --         | --         | ≈          | --    |
| ACT   | IACTA      | 10.8541        | 10.9843        | 11.2612        | 0.1062        | --    | --         | --         | +          | n.a.  |
|       | SIACTA-RSM | <b>0.2324</b>  | <b>0.2488</b>  | <b>0.2625</b>  | <b>0.0086</b> | +     | --         | +          | +          | n.a.  |
|       | SIACTA-RBF | 5.8122         | 7.1887         | 14.3174        | 1.5911        | +     | --         | --         | +          | n.a.  |
|       | SIACTA-GPR | 210.6180       | 240.6956       | 275.8460       | 17.7548       | --    | --         | --         | --         | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |
| NTO   | IACTA      | 0.1188         | 0.4265         | <b>0.5831</b>  | 0.1143        | --    | ≈          | --         | +          | n.a.  |
|       | SIACTA-RSM | 0.1897         | 0.4446         | 0.7311         | 0.1454        | ≈     | --         | --         | +          | n.a.  |
|       | SIACTA-RBF | <b>0.0511</b>  | <b>0.3824</b>  | 1.0007         | 0.1534        | +     | +          | --         | +          | n.a.  |
|       | SIACTA-GPR | 0.8632         | 1.0287         | 1.2637         | <b>0.0938</b> | --    | --         | --         | --         | n.a.  |
|       | OOCTA      | n.a.           | n.a.           | n.a.           | n.a.          | n.a.  | n.a.       | n.a.       | n.a.       | n.a.  |

erage computation time (ACT) required for the identification and prediction stages; and the normalized trade-off (NTO) between performance and computational burden.

Among the tested strategies, SIACTA-RBF proved to be the most effective, consistently achieving the best NTO by balancing control performance and computational cost. These results establish SIACTA-RBF as the most reliable option for compensating adverse conditions in real-time applications involving BLDC motors. SIACTA-RSM stands out for its significant advantage in reducing computational burden, achieving an ACT up to 43.5 times faster than traditional IACTA while maintaining comparable NTO results.

The SIACTA-GPR strategy failed to match, let alone surpass, the performance of the traditional IACTA strategy in any of the evaluated criteria and was the only adaptive method with performance levels comparable to the baseline offline approach (OOCTA). This behavior is expected because, in this study, it was necessary to extend its update period to  $250ms$  (twenty-five times longer than the other strategies) to allow execution of its identification and prediction stages. While this demonstrates the capability of the proposed framework to integrate diverse surrogate models within the tested BLDC motor platform, it also highlights key real-time constraints of GPR in fast-dynamic systems, suggesting caution for its use in high-speed scenarios.

Furthermore, the literature supports the view that surrogate models and optimization algorithms act as distinct components within adaptive tuning frameworks. Although keeping the ODE optimizer fixed isolates the influence of surrogate-model selection in our evaluation, quantifying the impact of alternative optimizers remains an avenue for future work.

Overall, the results suggest that surrogate-model selection should be treated as an application-dependent design decision, analogous to optimizer selection in related works, where the benefits of model interchangeability must be balanced against the computational requirements of the target system. Consequently, these findings emphasize that the effective deployment of surrogate models relies heavily on system-specific characteristics, such as physical complexity, operating magnitude, and real-time response constraints.

## 5. Conclusions

This work introduces a surrogate-model-based methodology for the indirect adaptive controller tuning approach (SIACTA) based on metaheuristic optimization, supported by real-world experimental evidence. The proposal addresses key limitations in conventional metaheuristic-based IACTAs by significantly reducing the computational burden while preserving control performance under nominal conditions or enhancing it under adverse conditions. The methodology is formulated to accommodate alternative surrogate-model strategies within a common identification-prediction-tuning framework. In this study, the response surface method (RSM), radial basis function (RBF), and Gaussian process regression (GPR) are implemented and experimentally compared under identical system, controller, optimization, and hardware configurations.

The methodology is validated through the speed regulation of a fast-response BLDC motor, on a real-world experimental platform, across three experiments of increasing complexity, incorporating parametric uncertainties and external disturbances. Comparisons were conducted against both the traditional explicit model-based indirect adaptive controller tuning approach (IACTA) and the conventional optimal offline

controller tuning approach (OOCTA).

The results indicate that, as experimental complexity increases, surrogate-assisted strategies emerge as the most reliable and robust strategy among the evaluated alternatives for the tested BLDC motor platform. Among the evaluated variants, SIACTA-RSM excels in reducing computational burden, achieving an average computation time 43 times faster than traditional IACTA. Meanwhile, SIACTA-RBF delivers the best overall performance, consistently outperforming IACTA across all evaluated criteria and delivering the best trade-off between system performance and computational efficiency among the surrogate-model-based strategies.

These outcomes provide evidence of the real-time feasibility of the proposed approach on the tested fast-response BLDC motor platform and show the framework's interchangeability of the surrogate model within the tested platform. Notably, the proposed framework is formulated around a baseline controller configuration and can, from a methodological perspective, be adapted to alternative control structures, surrogate-modeling techniques, optimization algorithms, and platforms, giving the fact that the proposed framework relies on generic system-level information and controller parameters, which facilitates such adaptation without requiring a redesign of the overall structure. Nevertheless, alternative control structures, plants, optimization algorithms, and hardware platforms remain to be experimentally established.

Future work will focus on expanding the methodology to incorporate additional surrogate-modeling techniques and multi-objective optimization frameworks to address multiple performance criteria simultaneously. Further experimental validation on other complex real-world systems and low-capacity embedded platforms is also planned.

## Acknowledgements

Authors acknowledge the support from the Comisión de Operación y Fomento de Actividades Académicas of the Instituto Politécnico Nacional (COFAA-IPN), the Secretaría de Investigación y Posgrado of the IPN (SIP-IPN), the Expert's Network in Robotics and Mechatronics of the IPN, the Colegio de Ciencia y Tecnología (CCyT) of the Universidad Autónoma de la Ciudad de México (UACM), and the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI). The first author acknowledges the support from the SECIHTI in Mexico, through a scholarship to pursue graduate studies at CIDETEC-IPN.

## CRedit authorship contribution statement

**Alam Gabriel Rojas-López:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing - original draft, Writing - review & editing, Data curation, Project administration. **Miguel Gabriel Villarreal-Cervantes:** Methodology, Supervision, Resources, Validation, Writing - review & editing, Project administration. **Alejandro Rodríguez Molina:** Conceptualization, Methodology, Validation, Supervision, Funding acquisition, Writing - review & editing. **Daniel Molina-Pérez:** Conceptualization, Methodology, Validation, Supervision, Writing - review & editing. **Gerardo Hernández-Hernández:** Conceptualization, Methodology, Validation, Supervision, Writing - review & editing.

## Disclosure statement

The authors report there are no competing interests to declare.

## Funding

This work was supported by the Secretaría de Investigación y Posgrado (SIP) of the Instituto Politécnico Nacional [grant numbers 20241335, 20250216].

## Data availability statement

The data that support the findings of this study are available from the first and corresponding author, [AGR-L, MGV-C], upon reasonable request.

## References

- Al-Tashi, Qasem, Said Jadid Abdulkadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Al-hussian. 2020. "Approaches to Multi-Objective Feature Selection: A Systematic Literature Review." *IEEE Access* 8: 125076–125096.
- Aliman, Norazam, Rizauddin Ramli, Sallehuddin Mohamed Haris, Mohammad Soleimani Amiri, and Mien Van. 2022. "A robust adaptive-fuzzy-proportional-derivative controller for a rehabilitation lower limb exoskeleton." *Engineering Science and Technology, an International Journal* 35: 101097.
- Almasalmah, Fadi Alyousef, Hassan Omran, Chao Liu, Thibault Poignonec, and Bernard Bayle. 2024. "Auto-Tuning of Model Predictive Control for Bilateral Teleoperation with Bayesian Optimization." *IFAC-PapersOnLine* 58 (30): 85–90.
- Alyoussef, Fadi, and Ibrahim Kaya. 2023. "Simple PI-PD tuning rules based on the centroid of the stability region for controlling unstable and integrating processes." *ISA Transactions* 134: 238–255.
- Bishop, Christopher M, and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. New York, NY: Springer.
- Bolton, William. 2003. *Mechatronics: electronic control systems in mechanical and electrical engineering*. Pearson Education.
- Borwein, Peter, and Tamás Erdélyi. 1995. *Polynomials and polynomial inequalities*. Vol. 161. Springer Science & Business Media.
- Buhmann, M. D. 2000. "Radial basis functions." *Acta Numerica* 9: 1–38.
- Chelladurai, Samson Jerold Samuel, Murugan K., Abhra Pratip Ray, Makarand Upadhyaya, Venugopal Narasimharaj, and Gnanasekaran S. 2021. "Optimization of process parameters using response surface methodology: A review." *Materials Today: Proceedings* 37: 1301–1304. International Conference on Newer Trends and Innovation in Mechanical Engineering: Materials Science.
- Chen, Heping, and Hongtai Cheng. 2021. "Online performance optimization for complex robotic assembly processes." *Journal of Manufacturing Processes* 72: 544–552.
- Chen, Jie, Weiyi Ying, Zhengchun Hua, Baoshi Cao, and Jie Zhao. 2024. "Efficient Impedance Controller Tuning of Robot–Environment Interaction With Improved Transient and Steady-State Performance." *IEEE Transactions on Instrumentation and Measurement* 73: 1–10.
- Chen, Qingda, Jinliang Ding, Shengxiang Yang, and Tianyou Chai. 2020. "A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems." *IEEE Transactions on Evolutionary Computation* 24 (4): 792–806.

- Cho, Kwanghyun, Ketong Shao, and Ali Mesbah. 2024. "Run-indexed time-varying Bayesian optimization with positional encoding for auto-tuning of controllers: Application to a plasma-assisted deposition process with run-to-run drifts." *Computers & Chemical Engineering* 185: 108653.
- Cohen, G. H., and G. A. Coon. 1953. "Theoretical Consideration of Retarded Control." *Journal of Fluids Engineering* 75 (5): 827–834.
- Ebden, Mark. 2015. "Gaussian Processes: A Quick Introduction." .
- Fasshauer, Gregory E. 2007. *Meshfree approximation methods with Matlab (With Cd-rom)*. Vol. 6. World Scientific Publishing Company.
- Forrester, Alexander I. J., András Sóbester, and Andy J. Keane. 2008. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley.
- Franceschi, Paolo, Marco Maccarini, Dario Piga, Manuel Beschi, and Loris Roveda. 2023. "Human Preferences' Optimization in pHRI Collaborative Tasks." In *2023 20th International Conference on Ubiquitous Robots (UR)*, 693–699. IEEE.
- Jiang, Ping, Qi Zhou, and Xinyu Shao. 2020. *Surrogate Model-Based Engineering Design and Optimization*. Springer Singapore.
- Jin, Yaochu. 2011. "Surrogate-assisted evolutionary computation: Recent advances and future challenges." *Swarm and Evolutionary Computation* 1 (2): 61–70.
- Khuri, André I., and Siuli Mukhopadhyay. 2010. "Response surface methodology." *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2): 128–149.
- Lan, Gongjin, Jakub M. Tomczak, Diederik M. Roijers, and A.E. Eiben. 2022. "Time efficiency in optimization with a bayesian-Evolutionary algorithm." *Swarm and Evolutionary Computation* 69: 100970.
- Landau, Ioan Doré, Rogelio Lozano, Mohammed M'Saad, and Alireza Karimi. 2011. *Adaptive Control: Algorithms, Analysis and Applications*. Springer London.
- Lin, Jiabin, Hai-Lin Liu, Kay Chen Tan, and Fangqing Gu. 2021. "An Effective Knowledge Transfer Approach for Multiobjective Multitasking Optimization." *IEEE Transactions on Cybernetics* 51 (6): 3238–3248.
- Lishchenko, Natalia, Anatoly Gushchin, and Vasily Larshin. 2024. "Hierarchical Control in Mechatronic Technological Systems." *Machines* 12 (10): 697.
- Lu, Qiugang, Leonardo D. González, Ranjeet Kumar, and Victor M. Zavala. 2021. "Bayesian optimization with reference models: A case study in MPC for HVAC central plants." *Computers & Chemical Engineering* 154: 107491.
- Maccarini, Marco, Filippo Pura, Dario Piga, Loris Roveda, Lorenzo Mantovani, and Francesco Braghin. 2022. "Preference-Based Optimization of a Human-Robot Collaborative Controller." *IFAC-PapersOnLine* 55 (38): 7–12.
- Myers, Raymond H, Douglas C Montgomery, and Christine M Anderson-Cook. 2016. *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons.
- Myers, Raymond H, Douglas C Montgomery, G Geoffrey Vining, and Timothy J Robinson. 2012. *Generalized linear models: with applications in engineering and the sciences*. John Wiley & Sons.
- Paredes-Ballesteros, Jesús Aldo, Miguel Gabriel Villarreal-Cervantes, Saul Enrique Benitez-Garcia, Alejandro Rodríguez-Molina, Alam Gabriel Rojas-López, and Victor Manuel Silva-García. 2025. "Optimal Tuning of Robot-Environment Interaction Controllers via Differential Evolution: A Case Study on (3,0) Mobile Robots." *Mathematics* 13 (11).
- Qiu, Yuchao, Zuhua Xu, Jun Zhao, Chunyue Song, and Xiaoping Zhu. 2025. "Data-driven controller parameters online tuning method based on model-inherited trust region Bayesian optimization." *Computers & Chemical Engineering* 199: 109141.
- Reynoso-Meza, Gilberto, Xavier Blasco, Javier Sanchis, and Miguel Martínez. 2014. "Controller tuning using evolutionary multi-objective optimisation: Current trends and applications." *Control Engineering Practice* 28: 58–73.
- Rodríguez-Molina, Alejandro, Efrén Mezura-Montes, Miguel G Villarreal-Cervantes, and Mario Aldape-Pérez. 2020. "Multi-objective meta-heuristic optimization in intelligent control: A

- survey on the controller tuning problem.” *Applied Soft Computing* 93: 106342.
- Rodríguez-Molina, Alejandro, Miguel Gabriel Villarreal-Cervantes, Jesús Said Pantoja-García, Alam Gabriel Rojas-López, Eric Hernández-Castillo, and Ricardo Mejía-Rodríguez. 2024. “Metaheuristic Adaptive Control Based on Polynomial Regression and Differential Evolution for Robotic Manipulators.” *Applied Soft Computing* 151: 111116.
- Rodríguez-Molina, Alejandro, Miguel Gabriel Villarreal-Cervantes, Omar Serrano-Pérez, José Solís-Romero, and Ramón Silva-Ortigoza. 2022. “Optimal Tuning of the Speed Control for Brushless DC Motor Based on Chaotic Online Differential Evolution.” *Mathematics* 10 (12).
- Rojas-López, Alam Gabriel, Miguel G. Villarreal-Cervantes, Alejandro Rodríguez-Molina, and Jesús A. Paredes-Ballesteros. 2024a. “Surrogate Adaptive Controller Tuning Based on DE in a 3R Serial Robot: A Comparative Analysis.” In *2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA)*, Vol. 275, Aug, 1–7. IEEE.
- Rojas-López, Alam Gabriel, Miguel G. Villarreal-Cervantes, Alejandro Rodríguez-Molina, and Jesús Aldo Paredes-Ballesteros. 2024b. “Comparative Analysis of Indirect Adaptive Controller Tuning Strategies Using Surrogate and Model-Based Techniques Applied to the Omnidirectional Mobile Robot.” In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, Vol. 275, Jul, 91–96. IEEE.
- Rojas-López, Alam Gabriel, Miguel Gabriel Villarreal-Cervantes, and Alejandro Rodríguez-Molina. 2023. “Optimum Online Controller Tuning Through DE and PSO Algorithms: Comparative Study with BLDC Motor and Offline Controller Tuning Strategy.” In *2023 10th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, Nov, 215–221. IEEE.
- Rojas-López, Alam Gabriel, Miguel Gabriel Villarreal-Cervantes, and Alejandro Rodríguez-Molina. 2024. “Surrogate Indirect Adaptive Controller Tuning Based on Polynomial Response Surface Method and Bioinspired Optimization: Application to the Brushless Direct Current Motor Controller.” *Expert Systems with Applications* 245: 123070.
- Rojas-López, Alam Gabriel, Miguel Gabriel Villarreal-Cervantes, Alejandro Rodríguez-Molina, and Jesús Aldo Paredes-Ballesteros. 2025. “Efficient Online Controller Tuning for Omnidirectional Mobile Robots Using a Multivariate-Multitarget Polynomial Prediction Model and Evolutionary Optimization.” *Biomimetics* 10 (2): 114.
- Roveda, Loris, Marco Forgione, and Dario Piga. 2020a. “Robot control parameters auto-tuning in trajectory tracking applications.” *Control Engineering Practice* 101: 104488.
- Roveda, Loris, Marco Forgione, and Dario Piga. 2020b. “Two-Stage Robot Controller Auto-Tuning Methodology for Trajectory Tracking Applications.” *IFAC-PapersOnLine* 53 (2): 8724–8731.
- Roveda, Loris, Lorenzo Mantovani, Marco Maccarini, Francesco Braghin, and Dario Piga. 2023a. “Optimal physical human–robot collaborative controller with user-centric tuning.” *Control Engineering Practice* 139: 105621.
- Roveda, Loris, Palaniappan Veerappan, Marco Maccarini, Giuseppe Bucca, Arash Ajoudani, and Dario Piga. 2023b. “A human-centric framework for robotic task learning and optimization.” *Journal of Manufacturing Systems* 67: 68–79.
- Schulz, Eric, Maarten Speekenbrink, and Andreas Krause. 2018. “A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions.” *Journal of Mathematical Psychology* 85: 1–16.
- Stewart, James, Daniel K Clegg, and Saleem Watson. 2020. *Calculus: early transcendentals*. Cengage Learning.
- Tesfazgi, Samuel, Armin Lederer, Johannes F. Kunz, Alejandro J. Ordóñez-Conejo, and Sandra Hirche. 2023. “Model-Based Robot Control with Gaussian Process Online Learning: An Experimental Demonstration.” *IFAC-PapersOnLine* 56 (2): 501–506.
- Villarreal-Cervantes, Miguel G., Alejandro Rodríguez-Molina, Consuelo-Varinia García-Mendoza, Ollin Peñaloza-Mejía, and Gabriel Sepúlveda-Cervantes. 2017. “Multi-Objective On-Line Optimization Approach for the DC Motor Controller Tuning Using Differential Evolution.” *IEEE Access* 5: 20393–20407.
- Williams, Christopher KI, and Carl Edward Rasmussen. 2006. *Gaussian processes for machine*

- learning*. Vol. 2. Massachusetts: MIT press Cambridge, MA.
- Xu, Wenjie, Bratislav Svetožarević, Loris Di Natale, Philipp Heer, and Colin N. Jones. 2024. “Data-driven adaptive building thermal controller tuning with constraints: A primal–dual contextual Bayesian optimization approach.” *Applied Energy* 358: 122493.
- Zaki, Hammad, Aamir Rashid, and Usman Masud. 2024. “Lyapunov-based fractional-order PID controller design for coupled nonlinear system.” *Transactions of the Institute of Measurement and Control* 47 (6): 1046–1056.
- Zhang, Lingzhi, Lei Xie, Hongye Su, and Jinfeng Liu. 2024. “Data-driven auto-tuning strategy for RTO-MPC based on Bayesian optimization.” *Computers & Chemical Engineering* 187: 108743.
- Zhu, Ning, Xin-Tong Gao, and Chun-Qing Huang. 2022. “A data-driven approach for on-line auto-tuning of minimum variance PID controller.” *ISA Transactions* 130: 325–342.
- Ziegler, J. G., and N. B. Nichols. 1942. “Optimum Settings for Automatic Controllers.” *Journal of Dynamic Systems, Measurement, and Control* 115 (2B): 220–222.
- Šafarič, Jakob, Primož Bencak, Dušan Fister, Riko Šafarič, and Iztok Fister. 2020. “Use of stochastic nature-inspired population-based algorithms within an online adaptive controller for mechatronic devices.” *Applied Soft Computing* 95: 106559.