A Learning-based Online Controller Tuning Method for Electric Motors using Gaussian Processes

Zhenxiao Yin\textsuperscript{1}, Xiaobing Dai\textsuperscript{1}, Zewen Yang\textsuperscript{1}, Dianxun Xiao\textsuperscript{1}, Filippo Menolascina\textsuperscript{1}, Sandra Hirche\textsuperscript{1}, and Hang Zhao\textsuperscript{1}

\textsuperscript{1}Affiliation not available

January 25, 2024
A Learning-based Online Controller Tuning Method for Electric Motors using Gaussian Processes

Zhenxiao Yin†, Student Member, IEEE, Xiaobing Dai†, Zewen Yang, Dianxun Xiao, Member, IEEE, Filippo Menolascina, Sandra Hirche, Fellow, IEEE, Hang Zhao*, Member, IEEE

Abstract—In industrial applications, Proportional-Integral (PI) controllers are frequently employed for controlling Permanent Magnet Synchronous Motors (PMSMs) due to their fast response rate and comprehensibility. However, their control performance may deteriorate with unforeseen environmental disturbances and uncertainties. To enhance the adaptivity of the controller, Gaussian Process Regression (GPR), a machine learning technique, is used to mitigate the impact of unknown components in system dynamics in this paper. In particular, GPR is adopted to autonomously tune the parameters of the PI controller, composing a novel GPR-based PI (GPR-PI) controller that maintains both interpretability and safety, because of its theoretical prediction error bound. Moreover, the stability of the system is guaranteed under the sufficiently small designed learning rate of the PI coefficients in the gradient descent rule, indicating the trade-off between stability and adaptivity. Then, the GPR-based PI is compared with the NN-based PI and demonstrates the priority of enlarging the stabilized speed range. Ultimately, the experiments validate the efficacy of the GPR-PI approach, showcasing a significant reduction of over 60% in response time when compared to the standard PI controller.

Index Terms—Gaussian Process Regression, Machine Learning Control, Permanent Magnet Synchronous Motor, Parameter Tuning, Online Learning.

I. INTRODUCTION

PERSISTENT magnet synchronous motors (PMSMs) have the characteristics of high-power density, speed, and smoothness [1]–[3], and therefore are widely employed in various scenarios, such as underwater vehicles [4], surgical manipulators [5]. In most fields of application, the PMSM control task is set as speed-tracking, where a guaranteed and explainable tracking performance is essential for safe operation [6]. For speed tracking, proportional-integral (PI) controllers are usually adopted due to their simple structures and easy implementations. However, the performance, such as acceptable overshooting [7] and fast convergence [8], under fixed fine-tuned PI coefficients degrades due to unexpected spontaneous environmental disturbance [9]. The severe consequences of large overshooting and slow convergence range from damaging the motors [10] to causing personal injury [11]. Therefore, an adaptive controller is required to guarantee the desired control performance under an unknown environment.

In industrial applications, ensuring the stability of Permanent Magnet Synchronous Motors (PMSMs) across all speed ranges typically relies on the utilization of a controller parameter look-up table [12]–[14]. However, the construction of such a parameter table demands an extensive series of laboratory experiments and is theoretically impossible to encompass all conceivable scenarios [15]. As alternative approaches, heuristic methods using computational intelligence, such as genetic algorithms [16] and particle swarm optimization [17], are explored to identify a small set of control parameters by analyzing certain predefined cases to cover the most common cases. However, the optimality of the obtained parameter set is restricted by the given set of control tasks and scenarios [16], [17], making it hard to apply in a new operating environment. Moreover, it is based on (quasi-)steady states and ignores the transition process, making the controller unsuitable for varying environments, such as the different torques and the temperature effects on the motor parameters [18]. Therefore, dynamic adaptive control strategies with less human design efforts for varying operation status of PMSMs are desired.

The dynamic adaptive PI controller usually updates the PI parameters using the gradient descent strategy, which requires the relationship between the tracking error and the PI parameters [19]. However, such a relationship, including the influence of unmolded PMSMs parts and the external environment, is generally unknown [20]. To infer the unknown relationship, data-driven strategies, e.g., machine learning techniques, are commonly employed to tune the parameters of controllers. To this end, neural network (NN) based controller tuning is widely studied, whose control performance relies strongly on the accuracy level of nonlinear features of neural networks [21]. Accurate features as initial value require sufficient knowledge of the unknown system or well-trained hyperparameters [22], inducing large human efforts. Furthermore, the hyperparameter optimization process lacks transparency and guarantee [23]. Gaussian process regression (GPR), another supervised machine learning method, [24] is commonly used for control in state space, due to its flexibility for model inference [25], [26] and the existence of the theoretical error bound [27]. The GPR method examined in [28] is employed for predicting the unknown system dynamics, followed by parameter adjustment for the controller, which is accomplished
through optimization of an objective function associated with tracking error [29]. However, except for the lack of stability analysis, the optimization-based PI tuning is hard to apply for higher frequency control for PMSMs due to its high computational load. In our work, instead of estimating the unknown dynamics, the GPR is used to infer the Jacobian function related to the output speed and input current of PMSMs’ model, which is then directly applied to the PI parameter update law with an acceptable computational load.

Furthermore, the performance of learning-based control hinges on the accuracy of the GPR prediction model [30], [31], highlighting the significance of appropriate hyperparameters and the quantity of training samples [32]. Attaining a desirable overall prediction accuracy necessitates a substantial amount of training data, which poses challenges for practical implementation in PMSMs’ control applications in motor control units (MCU). A promising approach to enhance local prediction accuracy with manageable data sets relevant to specific tasks is online learning, as proposed in [33]. Nevertheless, streaming data collection leads to a linear growth in dataset size over time, posing challenges in terms of infinite data storage requirements [34]. To tackle this issue, online learning is implemented to improve the control performance. As the training dataset relies on the most recent data, the accuracy pertaining to the current situation is also high. The main contributions of this work include:

1) Developing a tuning scheme based on GPR for PI controller in real-time using Lyapunov stability theory;
2) Stability analysis of the PMSMs using the proposed online learning-based control method with guaranteed control performance;
3) Comparing the GPR-based controller with NN based controller. The stabilized speed range is enlarged using GPR demonstrated by simulation.
4) Demonstration of the effectiveness of the proposed learning-based controller in real experiments. The response time decreases by more than 60% compared to the basic PI controller under uncertainties.

The remainder of the paper is structured as follows. Section II describes the problem setting and the foundation of GPR. The GPR-based PI controller is proposed in Section III with a guarantee of stability. Simulations and experiments are provided to demonstrate the effectiveness of the proposed control strategy in Section IV. Finally, Section V concludes the paper.

II. PRELIMINARIES AND PROBLEM SETTING

A. Notation

The set of natural numbers excluding zero is indicated as \( \mathbb{N}^+ \), while real positive numbers, both including and excluding zero, are denoted as \( \mathbb{R}^+ \) and \( \mathbb{R}_{a,+} \), respectively. Unless explicitly stated, the symbol \(| \cdot |\) signifies the absolute value, and \(| \cdot |\) denotes the Euclidean norm. The \( i \)-th component of a vector \( \mathbf{a} \) is denoted as \( a_i \), while \( a_{ij} \) indicates the element located at the intersection of the \( i \)-th row and the \( j \)-th column within the matrix \( \mathbf{A} \). The smallest and biggest eigenvalues of a matrix are represented by \( \lambda_{\min} \) and \( \lambda_{\max} \), respectively.

B. System Description

In this paper, we consider the speed control of PMSMs, whose dynamics is described in a discrete-time form as follows

\[
\omega_{k+1} = f(\omega_k, u_k),
\]

where the rotation speed \( \omega_k \in \mathbb{W} = [\omega, \bar{\omega}] \subset \mathbb{R} \) is regarded the state of the system. The control input \( u_k \in \mathbb{U} = [u, \bar{u}] \subset \mathbb{R} \) is defined as the increment of input current of the PMSM, and the unknown function \( f(\cdot) : \mathbb{W} \times \mathbb{U} \to \mathbb{W} \) is defined as the closed-loop controlled system by the PI-controller for the PMSM. Considering the actual limitation of the motor speed and current, the domains \( \mathbb{W} \) and \( \mathbb{U} \) are compact. Note that the dynamic function \( f(\cdot) \) encodes unknown environmental effects, such as external loads, and unmodeled parts of the system, which are hard to directly derive from the first principles. Therefore, the dynamic function is assumed to be unknown but satisfies the following assumption.

Assumption 1: The unknown function \( f(\cdot) : \mathbb{W} \to \mathbb{W} \), where \( \bar{\omega} = [\omega, \bar{\omega}]^T \in \mathbb{R}^2 \), \( f(\cdot) \) is at least \( C^2 \) continuous w.r.t \( u \). Furthermore, the Jacobian function defined as \( \tilde{h}(\bar{\omega}) = \partial f(\bar{\omega})/\partial u \) is Lipschitz continuous with Lipschitz constant \( L_{\tilde{h}} \in \mathbb{R}^+ \), i.e., \( \| \tilde{h}(\bar{\omega}) \| \leq L_{\tilde{h}} \). Furthermore, the Jacobian function \( h(\cdot) \) is positive and bounded, i.e. there exist \( \bar{h}, \tilde{h} \in \mathbb{R}^+ \) such that \( \| h(\bar{\omega}) \| \leq [\bar{h}, \tilde{h}], \forall \omega \in \mathbb{W} \), \forall \omega \in \mathbb{W} \) and \( u \in \mathbb{U} \).

According to the physical law of PMSMs, we consider \( \nabla_\omega \partial f(\bar{\omega})/\partial u \) is proportional to flux linkage \( \psi_f \in \mathbb{R}_{p,+} \) and sampling time of speed loop \( T_s \in \mathbb{R}_{+} \), which are positive and bounded, inverse proportional to inertia \( J \in \mathbb{R}_{+} \), which is also positive and bounded [35]. The pole pair is fixed after the PMSM is manufactured, which is positive and also positively related to the Jacobian Term. Even when the temperature or torque changes during operation, these parameters satisfy the noted condition that these parameters are positive and bounded [36]. However, some of the above mentioned parameters are changing during the operation, which will affect the control process. Thus, the real-time data-driven method is needed to help the controller learn the real change of the system.

The control task is to track a pre-defined speed trajectory \( \omega_{r,k} \in \mathbb{W} \), such that the tracking error \( e_k = \omega_k - \omega_{r,k} \in \mathbb{E} \) is forced to be small. For this purpose, a PI controller incorporated with the tracking error is designed in a truncated integration form of [37] as

\[
\begin{align*}
    u_k = K_{p,k}e_k + K_{i,k}(e_k + e_{k-1}),
\end{align*}
\]

where the control gains \( K_{p,k} = K_{p,k}(e_k, e_{k-1}) \) and \( K_{i,k} = K_{i,k}(e_k, e_{k-1}, e_{k-2}) \) for the sake of simplicity. Note that the error domain \( \mathbb{E} \subset \mathbb{R} \) is compact due to the compactness of \( \mathbb{W} \) for the actual rotation speed and its reference, particularly, \( \mathbb{E} = [-2u_{max}, 2u_{max}] \) with \( u_{max} = \max(|u|, |\bar{u}|) \).

Using the proposed PI controller (2) and shifting \( \omega_k \) as \( \omega_k = e_k + \omega_{r,k} \), the Jacobian function is reformulated as

\[
\begin{align*}
    \tilde{h}(\bar{\omega}_k) &= \tilde{h}(\{e_k + \omega_{r,k}, K_{p,k}e_k + K_{i,k}(e_k + e_{k-1})\}^T) \\
    &= h(\{e_k, e_{k-1}\}^T) = h(x_k)
\end{align*}
\]

with a newly defined function \( h(\cdot) : \mathbb{X} \to \mathbb{W} \) and the corresponding input \( x_k = [e_k, e_{k-1}]^T \in \mathbb{X} \), where \( \mathbb{X} \subset \mathbb{R}^2 \).
is compact. Moreover, the function \( h(\cdot) \) inherits the property of \( \hat{h}(\cdot) \), which is shown as follows.

**Proposition 1:** Consider the dynamics of the speed loop of PMSMs as in (1) satisfying Assumption 1. The system is controlled by (2) with well-defined coefficients \( K_{p,k}, K_{i,k} \in \mathbb{R}_+ \), such that (3) holds for all \( k \in \mathbb{N} \). Then the Jacobian function \( h(x) \) is Lipschitz continuous, which means there exists a positive constant \( L_h \in \mathbb{R}_+ \) such that \( \| \nabla_x h(x) \| \leq L_h \) holds for \( \forall x \in \mathcal{X} \). The valid choice of \( L_h \) satisfies

\[
L_h \geq \max(\sqrt{2}, \sqrt[3]{2(\bar{K}_p + \bar{K}_i)}) L_h
\]

where the positive constants denote \( \bar{K}_p = \max_{k \in \mathbb{N}} K_{p,k} \) and \( \bar{K}_i = \max_{k \in \mathbb{N}} K_{i,k} \).

**Proof:** Considering the definition of \( h(\cdot) \) in Eq. (3), the derivative of \( h(\cdot) \) is formulated with the chain rule with respect to \( e_k \) and \( e_{k-1} \) is calculated as follows

\[
\frac{\partial h(x_k)}{\partial e_k} = \frac{\partial h(\bar{\omega}_k)}{\partial \omega_k} \frac{\partial \omega_k}{\partial e_k} + \frac{\partial h(\tilde{\omega}_k)}{\partial \omega_k} \frac{\partial \omega_k}{\partial e_k} + \frac{\partial h(\tilde{\omega}_k)}{\partial e_k} = \frac{\partial h(\bar{\omega}_k)}{\partial \omega_k} (K_{p,k} + K_{i,k}) \frac{\partial \omega_k}{\partial u_k},
\]

Thus, the square of norm of \( \nabla_x h(x_k) \) is written as

\[
\| \nabla_x h(x_k) \|^2 = \left( \frac{\partial h(\bar{\omega}_k)}{\partial \omega_k} \right)^2 \left( K_{p,k} + K_{i,k} \right)^2 + \left( \frac{\partial h(\tilde{\omega}_k)}{\partial \omega_k} \right)^2 \left( K_{p,k} + K_{i,k} \right)^2 + \left( \frac{\partial h(\tilde{\omega}_k)}{\partial e_k} \right)^2 = \| \nabla_x h(x_k) \| \leq \sqrt{2} \sqrt[3]{2(\bar{K}_p + \bar{K}_i) L_h} \| X \|.
\]

Moreover, employing the triangular inequality and Young’s inequality, which results in \( (a + b)^2 \leq (\|a\| + \|b\|)^2 \leq \|a\|^2 + \|b\|^2 + 2\|a\|\|b\| \leq 2(\|a\|^2 + \|b\|^2) \), then we have

\[
\sup_{x \in \mathcal{X}} \| \nabla_x h(x_k) \| \leq \sqrt{2} \sqrt[3]{2(\bar{K}_p + \bar{K}_i) L_h} \| X \|.
\]

with the incremental terms \( \Delta K_{p,k} \) and \( \Delta K_{i,k} \) related to the gradient of \( J_k = e_k^2 / 2 \), which are calculated as

\[
\Delta K_{p,k} = -\lambda_p \frac{\partial J_k}{\partial K_{p,k-1}} = -\lambda_p \frac{\partial J_k}{\partial e_k} \frac{\partial e_k}{\partial u_{k-1}} \frac{\partial u_{k-1}}{\partial K_{p,k-1}}
\]

\[
\Delta K_{i,k} = -\lambda_i \frac{\partial J_k}{\partial K_{i,k-1}} = -\lambda_i \frac{\partial J_k}{\partial e_k} \frac{\partial e_k}{\partial u_{k-1}} \frac{\partial u_{k-1}}{\partial K_{i,k-1}}
\]

respectively, with the facts that \( \partial u_{k-1}/\partial K_{p,k-1} = e_{k-1}, \partial e_k/\partial u_{k-1} = h_{k-1}, \partial u_{k-1}/\partial K_{i,k-1} = e_{k-1} + e_{k-2}, \) where \( h_k = h(x_k) \) for notational simplicity. The positive values \( \lambda_p, \lambda_i \in \mathbb{R}_+ \) denote learning rates, inferring the changing speed of the coefficients, which are usually given as constants in the optimization process, or with the decay rate approaching the minimum point [38]. It is also shown in [39] that the selection of learning rates \( \lambda_p, \lambda_i \) in (10) and (11) influences the stability of the system using machine learning-based controller. This paper employs consistent and reasonable constant learning rates obeying Lyapunov theory so that the tracking error can achieve asymptotic stability. Since the specific dynamical function \( f(\cdot, \cdot) \) is not known, its corresponding gradient \( h(\cdot) \) is not accessible. Therefore, we model it by using the data-driven method, i.e., GPR, where the measurements for predicting \( h(\cdot) \) satisfy the following assumption.

**Assumption 2:** The data set \( \mathbb{D} = \{x_k, y_k\} \) contains \( N \in \mathbb{N} \) samples, where \( x_k = [e_k, e_{k-1}]^T \) in the compact domain \( \mathcal{X} \) is noise-free and \( y_k = h(x_k) + \tilde{v}_k \) is noisy. The composed output noise \( \tilde{v}_k \sim N(0, \sigma_n^2) \), \( \sigma_n \in \mathbb{R}_+ \) is independent, identical, and Gaussian distributed. The omission of noise in \( x_k \) is not a limiting factor, as any noise affecting speed measurements can effectively be accounted for by transferring it to \( y_k \) using methods like Taylor expansion as follows

\[
h(\bar{x}_k) + \tilde{y}_k = h(x_k) + \frac{\partial h}{\partial e_{k-1}} [e_{k-1} + e_{k-2}] v_{k-1} = \tilde{h}(x_k) + v_{k-1} \approx y_k,
\]

considering the non-negative coefficient \( K_{p,k}, K_{i,k} \). Note that the supremum of \( \| \nabla_x h(\bar{\omega}_k) \| \) is the Lipshitz constant \( L_h \) according to its definition in Assumption 1. Then choose \( L_h \) satisfying (4), the relationship with \( \sup_{x \in \mathcal{X}} \| \nabla_x h(x_k) \| \leq L_h \) holds, which concludes the proof.

Proposition 1 enables us to compute the Jacobian solely using logged tracking errors, offering a method to determine the Jacobian function without requiring additional information beyond the recorded tracking error data.

We consider the discrete time-varying control gains in (2) are updated by

\[
\Delta K_{p,k} = K_{p,k} - K_{p,k-1}, \quad \Delta K_{i,k} = K_{i,k} - K_{i,k-1}.
\]
components in the controller, i.e., \( h(\cdot) \), from the collected samples in this paper. A Gaussian process induces a Gaussian distribution characterized by the mean function \( m(\cdot) : \mathbb{X} \to \mathbb{R} \) and the kernel function \( \kappa(\cdot, \cdot) : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{0,+} \) satisfying the following assumption.

**Assumption 3:** The kernel function \( \kappa(||x - x'||) = \kappa(x, x') \) is chosen as stationary, monotonically decreasing, and Lipschitz continuous. Moreover, \( \kappa(0) = \sigma_f^2 \) with \( \sigma_f > 0 \).

The Lipschitz continuous kernel is a reasonable choice for continuous unknown functions in the compact domain. The monotonic decrease in \( \kappa(\cdot) \) indicates a weaker relationship between the training data and the evaluated point with a larger Euclidean distance. Therefore, this assumption is not restrictive for control applications.

The mean function \( m(\cdot) \) encodes the prior knowledge of the unknown function. Combined with the prior information and the data set, the prediction of Jacobian \( h \) in Eq. (10) and Eq. (11) at \( x_k \) is expressed in

\[
\mu_k(x_k) = m(x_k) + \kappa_X^T(K + \sigma_n^2I_n)^{-1}(y - m_X),
\]
(13)
under the measurement noise with variance \( \sigma_n^2 \) in Assumption 2, and its posterior variance is expressed as

\[
\sigma_k^2(x_k) = \kappa(x_k, x_k) - \kappa_X^T(K + \sigma_n^2I_n)^{-1} \kappa_X.
\]
(14)
where \( \kappa_X = \left[ \kappa(x(1), x), \cdots, \kappa(x(N), x) \right]^T \), \( K = \left[ \kappa(x(i), x(j)) \right]_{i,j=1,\ldots,N} \) and \( y = \left[ y(1), \cdots, y(N) \right]^T \).

The concatenated prior mean is defined as \( m_X = [m(x_1), \cdots, m(x_N)]^T \). Thanks to the theoretical error bound for GPR, the prediction accuracy can be probabilistic bounded as shown in the following lemma.

**Lemma 1:** [30] Consider an unknown function \( h(\cdot) \) satisfying Assumption 1, which is inferred through GPR with the kernel function satisfying Assumption 3. Assume a data set \( \mathbb{D} \) is available with \( N \in \mathbb{N} \) samples satisfying Assumption 2. Pick \( \delta \in (0, 1) \subset \mathbb{R} \), then the prediction error is bounded by:

\[
|h(x_k) - \mu_k(x_k)| \leq \sqrt{\beta(X, \delta, \tau)}\sigma_k(x_k) + \gamma(\delta, \tau) = \eta(x_k)
\]
(15)
with the probability of at least \( 1 - \delta \), in which

\[
\beta(X, \delta, \tau) = 2 \sum_{j=1}^n \log \left( \frac{\sqrt{n}}{2\tau} (\bar{x}_j - \underline{x}_j) + 1 \right) - 2 \log \delta,
\]

\[
\gamma(\delta, \tau) = \frac{\sqrt{\beta(X, \delta, \tau)}}{2} \sigma_L + L_f + \mu_L \tau.
\]
(16)
where \( \bar{x}_j = \max_{x \in \mathbb{X}} |x_j| \) and \( \underline{x}_j = \min_{x \in \mathbb{X}} |x_j| \) are the upper and lower bound of the \( j \)-th dimension of all \( x \in \mathbb{X} \). The positive values \( L_\mu \) and \( L_\sigma \) denote the Lipschitz constant for the posterior mean \( \mu(\cdot) \) and variance \( \sigma(\cdot) \), respectively. The detailed expressions of \( L_\mu \) and \( L_\sigma \) are available in [30].

Lemma 1 provides a data-related prediction error bound for GPR under the assumption that \( h(\cdot) \) is a sample from the corresponding GP with fine-tuned hyperparameters. Despite of the conservatism, it provides a quantitative method to calculate the prediction performance of the GPR, which is later employed to numerically analyze the proposed GP-based controller’s control performance.

### III. Stability Analysis of the PMSM with Adaptive Data-driven Controller

#### A. Tracking Performance with Perfect Prediction

Within this subsection, we initially investigate the stability conditions on \( \lambda_p \) and \( \lambda_i \) refers to (10), (11) when we have access to the precise dynamics \( f(\cdot) \) of the PMSM. The result is shown as follows.

**Theorem 1:** Consider the system satisfying Assumption 1 and controlled by an adaptive PI controller to track a desired trajectory. The PI controller’s control gains are tuned according to (10) and (11), where the gradient \( h(\cdot) \) is known. Choose positive learning rates \( \lambda_p \) and \( \lambda_i \) satisfying

\[
\lambda_p + 4\lambda_i < \frac{1}{2\bar{h}^2\bar{\omega}^2}
\]
(17)
with \( \bar{\omega} = \max_{\omega \in \mathbb{W}} |\omega| \) and \( \bar{h} > \max_{x \in \mathbb{X}} |h(x)| \). Then the controlled system is exponentially stable.

**Proof:** Consider the dynamics of the tracking error \( e \) as

\[
e_{k+1} = e_k + \Delta e_k,
\]
(18)
where the error increment \( \Delta e_k \) is related to the change of control parameters and written as

\[
\Delta e_k = \frac{\partial e_k}{\partial K_{p,k-1}}(K_{p,k} - K_{p,k-1}) + \frac{\partial e_k}{\partial K_{i,k-1}}(K_{i,k} - K_{i,k-1})
\]
\[
= \frac{\partial e_k}{\partial K_{p,k-1}}\Delta K_{p,k} + \frac{\partial e_k}{\partial K_{i,k-1}}\Delta K_{i,k}.
\]
(19)
Applying (10) and (11) in (19), the error increment becomes

\[
\Delta e_k = -\lambda_pe_k e_k e_k h_{k-1} - \lambda_i(e_k e_k + e_k e_k) h_{k-1} h_{k-1} h_{k-1} + \lambda_i (e_k e_k + e_k e_k)
\]
\[
= -e_k h_{k-1}^2 (\lambda_p \epsilon^2 + \lambda_i (e_k e_k + e_k e_k)^2)
\]
\[
= -e_k h_{k-1}^2 (\lambda_p \epsilon^2 + \lambda_i \epsilon^2).
\]
(20)
where \( \epsilon_k = (\epsilon_k^2 + \epsilon_k^2)^2 \). The result of control parameters and written as

\[
\Delta V_k = V_{k+1} - V_k = \Delta e_k (2e_k + \Delta e_k)
\]
\[
= -e_k h_{k-1}^2 (\lambda_p \epsilon^2 + \lambda_i \epsilon^2).
\]
(22)
Considering the facts that

\[
|\epsilon_{k-1} - \epsilon_{k-1}| \leq |\epsilon_{k-1} + \epsilon_{k-1}| \leq 2\bar{\omega},
\]
\[
|\epsilon_{k-1} + \epsilon_{k-1}| \leq |\epsilon_{k-1} + \epsilon_{k-1}| \leq 4\bar{\omega},
\]
(23)
(24)
the value of \( \epsilon_k^2 \) is bounded as

\[
\epsilon_k^2 = \epsilon^2 (\lambda_p (\epsilon_k^2 + \epsilon_k^2)^2 + \lambda_i (\epsilon_k^2 + \epsilon_k^2)^2)
\]
\[
\leq \epsilon^2 (\lambda_p + 4\lambda_i).
\]
(25)
due to $\lambda_p$ and $\lambda_i$ satisfied (17), where the inequality comes from the definition of the compact set $\mathbb{E}$ of $e_k$, indicating $e_k^2 \leq \bar{e}^2 = 4\omega^2$. Combining the condition (17), we have

$$0 < h_{k-1} \leq \frac{e_k^2}{2h_{k-1} \omega^2} < 2.$$  

(26)

Therefore, the $\Delta V_k$ in (22) satisfies

$$\Delta V_k \leq -\alpha_k e_k^2 = -\alpha_k V_k \leq 0,$$  

(27)

and $\alpha_k = \frac{4h_{k-1}}{\omega^2} \left(1 - \frac{h_{k-1}}{h_k}\right) \in (0, 1)$  

(28)

and if only if $|h_{k-1}| = |\hat{h}|$ according to the condition as shown in Assumption 1 and Eq. (3). It is intuitive that the avoidance of $\Delta V_k = 0$ is directly derived by choosing $\hat{h}$ slightly larger than $\max_{x \in \mathbb{X}} \|h(x)\|$, which inducing the strictly negative of $\Delta V_k$, i.e., $\Delta V_k < 0$, leading to asymptotically stable control system.

Note that the set $S = \{e_k \in \mathbb{E} | \Delta V_k = 0\}$ only contains one element, namely $e_k = 0$. According to the design in (11) and (10), it is straightforwardly to obtain $\Delta K_h = 0$ and $\Delta K_i = 0$, resulting in $\Delta V_k < 0$ from (19). This indicates $S$ is positive invariant, i.e., $e_k \equiv 0 \Rightarrow e_i = 0, \forall i \in \mathbb{N}, i \geq k$. Based on the LaSalle’s invariance principle, $e_k = 0$ is asymptotically stable.

Moreover, considering $V_{k+1} = V_k + \Delta V_k$ and $V_k = |e_k|^2$, then the tracking error is bounded by

$$|e_k| = \sqrt{V_k} \leq \left( \prod_{i=1}^{k-k_0} (1 - \alpha_i) \right)^{1/2} \leq \prod_{i=1}^{k-k_0} (1 - \alpha_i)^{1/2} |e_k|_0 \leq \left( 1 - \min_{i=1, \ldots, k-k_0} \alpha_i \right) (k-k_0)^{1/2} |e_k|_0.$$  

(29)

where $k_0 \in \mathbb{R}$ denotes the start step of the running process of PMSMs. Define a positive value $\alpha = \sqrt{1 - \min_{i=1, \ldots, k-k_0} \alpha_i}$, which is smaller than 1 if there exists $|h_k| \neq \hat{h}$, and then $|e_k| \leq \alpha (k-k_0)^{1/2} |e_k|_0$ leads to exponential stability.

Theorem 1 shows the exponential stability of the controlled system is attained when the learning rates $\lambda_p$ and $\lambda_i$ satisfy (17). Due to the positivity of $\lambda_p$ and $\lambda_i$, the maximal values of the learning rates are bounded by $\hat{h} \leq \omega^2/2$, inversely related to the upper bounds for the unknown function $h(\cdot)$ and angular velocity $\omega$. Note that large $\hat{h}$ and $\tilde{\omega}$ indicate a potentially fast change of states, and fast adaption with large $\lambda_p$ and $\lambda_i$ at the current state may deteriorate the performance, e.g., inducing instability, during the interval between two control actions.

Remark 1: The condition (17) reveals the trade-off between the convergence speed determined by the parameter $\lambda_p$, and the recovering speed for error offset caused by the parameter $\lambda_i$. To achieve rapid adaptation of the system to external disturbances, characterized by a high value of $\lambda_p$, it becomes imperative to employ a small value of $\lambda_i$ to preserve stability, making the adaptation for static errors slow. Note that the proven exponential stability in Theorem 1 eventually achieves the zero offset, but only requires more time. A smart way is to employ an error-related update rate, i.e., choosing $\lambda_p$ larger when the tracking error $e$ is larger for fast convergence, and larger $\lambda_i$ when $e$ is smaller for fast static error elimination. However, this error-based choice of $\lambda_p$ and $\lambda_i$ induces more computations, and in this paper, only constant learning rates will be selected offline for real-time implementation.

Remark 2: In comparison to the proportion term regulated by $\lambda_p$, the integral term controlled by $\lambda_i$ exerts a greater influence on stability. Because it is indicated by a higher coefficient preceding $\lambda_i$ in (17), specifically, a value of 4 for the truncated form. This phenomenon can be intuitively understood by considering that the integral term encompasses the accumulation of past tracking errors, which can potentially be larger than the proportion term during the transitional phase.

Furthermore, the inclusion of the limited-case scenario, which encompasses the transitional phase, in the determination of the maximal learning rate amplifies the significance of $\lambda_i$.

Note that Theorem 1 only considers the case with perfect information of prediction, but in reality, $h(\cdot)$ is unknown. Instead, the prediction $\mu(\cdot)$ is used, and the control performance under the predicted system dynamics is discussed next.

Notably, Theorem 1 exclusively pertains to scenarios in the function $h(\cdot)$ is accessible. However, in the context of practical experiments, the function $h(\cdot)$ is uncertain. Instead, the prediction for $h(\cdot)$ is employed by using GPR, and the subsequent section investigates the control performance with the predicted system dynamics.

B. Gaussian Process-based Tracking Performance Guarantee

In this subsection, the control performance by using the proposed adaptive PI controller with GPR is analyzed. Note that the predictions $\mu(\cdot)$ are only applied on the determination of $\Delta K_p,k$ and $\Delta K_i,k$, while the partial derivatives $\partial e_k/\partial K_h,k$ and $\partial e_k/\partial K_i,k$ are related to the true system with $h(\cdot)$. Before the discussion of the control performance for the proposed GPR-PI controller, the satisfaction of the property for the positivity of $h(\cdot)$ is shown as follows.

Lemma 2: Consider a GPR model with the kernel function satisfying Assumption 3 and a modified data set $\mathbb{D} = \{x_k, \rho(y_k)\}_{k=1}^N$, where the original data pairs $\{x_k, y_k\}$ satisfy Assumption 2 and the modification function $\rho(\cdot)$ is a ReLU function defined as $\rho(y) = \max(0, y)$.

Then, a differentiable prior mean function $m(\cdot)$ exists such that the posterior mean satisfies $\mu(x) > 0, \forall x \in \mathbb{X}$.

Proof: Since $\kappa_X^T (K + \sigma_n^2 I_n)^{-1} (y - m_X) \leq \|\kappa_X\|\|K + \sigma_n^2 I_n\|^{-1} (y - m_X)$ holds bounded by $\kappa_X^T (K + \sigma_n^2 I_n)^{-1} (y - m_X) \leq \sqrt{N} \kappa(0) \|K + \sigma_n^2 I_n\|^{-1} (y - m_X) $ with the right-hand side as a constant by given data set $\mathbb{D}$, there exists a continuous prior mean function $m(\cdot)$, which satisfies $m(x) > \sqrt{N} \kappa(0) \|K + \sigma_n^2 I_n\|^{-1} (y - m_X)$ for all $x \in \mathbb{X} \setminus \{x_1, \ldots, x_N\}$. For all the samples $k = 1, \ldots, N$ in data set $\mathbb{D}$, choose $m(x_k) = \rho(y_k) > 0$, then the positivity of the posterior mean for $x \in \mathbb{X}$ is derived.

Lemma 2 shows the positivity of the posterior prediction can always be obtained by using the modified data set $\mathbb{D}$ and choosing a proper prior mean $m(\cdot)$. In practice, $m(\cdot)$ is always chosen as a large sufficient constant.

Remark 3: Due to the assumption of $h(\cdot)$ in Assumption 1, the data pair modified by the ReLU function has a smaller measurement noise $|v|$ by considering

$$|h_k - \rho(y_k)| = |h_k - h_k - y_k| = |h_k - y_k| = |g_k|$$  

(31)
for $y_k < 0$ and $|\Delta h_k| = |h_k - y_k| = |\hat{y}_k|$ for $y_k \geq 0$, indicating that the modified data set also satisfies Assumption 2. Therefore, the prediction error bound in Lemma 1 still holds.

With the guaranteed positivity of the prediction $\mu_k$, the control performance of GPR-PI is shown as follows.

**Theorem 2:** Consider the system satisfying Assumption 1 and controlled by a PI controller to track a desired trajectory. The PI controller’s control gains are tuned according to Eq. (2), Eqs. (9) to (11), where the unknown part $h(x_k)$ is predicted using GPR with Assumption 3 from the data set $\mathcal{D}$ containing $N \in \mathbb{N}$ samples satisfying Assumption 2. Moreover, a prior mean function $m(\cdot)$ is chosen such that $\mu_k(x) \geq 0, \forall x \in \mathcal{X}$. Pick $\xi \in (0, 1)$, $\chi \in (0, 1)$ and choose the learning rates satisfying

$$\lambda_p + 4\lambda_i < \omega^{-2}\hat{h}^{-1}(\hat{h} + \eta)^{-1}/2$$

with $\dot{\eta} = \max_{x \in \mathcal{X}} \eta(x)$. Then, the controlled system is exponentially stable.

**Proof:** By using the prediction $\mu_k$ to replace $h_k$ in the computation of the coefficient increment $\Delta K_P, h_k$ and $\Delta K_I$ as in (10) and (11) respectively, the increment of the tracking error $\Delta e_k$ in (19) is reformulated as

$$\Delta e_k = -\epsilon_k h_{k-1} \mu_{k-1} \epsilon_k \lambda,$$

where $\mu_{k-1} = \mu(x_{k-1})$ for notational simplicity. Choose the Lyapunov candidate as in (21), and then $\Delta V_k$ is written as

$$\Delta V_k = \epsilon_k^T h_{k-1} \mu_{k-1} \epsilon_k \lambda (2 - h_{k-1} \mu_{k-1} \epsilon_k^T \lambda),$$

where $h_{k-1}\mu_{k-1}$ is bounded by

$$\mu_k h_k = h_k^2 + h_k (\mu_k - h_k) \leq h_k^2 + h_k |\mu_k - h_k| \leq h_k^2 + h_k \eta_k$$

with the probability of at least $1 - \delta$, where $\eta_k = \eta(x_k)$.

Note that the first inequality in (35) holds due to the positive $h(\cdot)$ in Assumption 1, and the second from prediction error bound in Lemma 1. Moreover, considering the upper bounds of $h(\cdot)$, $\eta(\cdot)$ and (25), the positive $\epsilon_k^T \lambda \mu_k h_k$ is upper bounded by

$$\epsilon_k^T \lambda \mu_k h_k \leq (\lambda_p + 4\lambda_i) \epsilon_k^2 \mu_k h_k < 2\frac{h_k(h_k + \eta_k)}{h(h + \eta)}.$$

Then, the increment of the Lyapunov function is bounded by

$$\Delta V_k = -\alpha \epsilon_k^2 = -\alpha V_k,$$

$$\alpha_k = \frac{4}{h_k(h_k + \eta_k)} \left(1 - \frac{h_k(h_k + \eta_k)}{h(h + \eta)}\right) \in (0, 1).$$

Similarly as the proof for Theorem 1, the exponential stability is achieved considering $|\epsilon_k| \leq \alpha^{k-1} |\epsilon_0|$ for $\alpha \in (0, 1)$. Theorem 2 shows the exponential stability is achieved by choosing lower learning rates $\lambda_p$ and $\lambda_i$ than in (17). Slower adaptation is naturally expected, as making significant changes to both $K_P$ and $K_I$ in large steps could lead to instability, particularly with inaccurate predictions, i.e., large $\eta$.

**Remark 4:** Since the Jacobian function $h(\cdot)$ is unknown, its upper bound may not be available from the first principle but can be approximated by using GPR [27]. Choose any $\delta \in (0, 1)$, and then the upper bound of $h(\cdot)$ is approximated as

$$\hat{h} \leq \sigma_f \sqrt{2 \log(\delta^{-1}) + 24 \sqrt{\log(5\sigma_f^{-1} \sqrt{vL_k})}}$$

with probability of at least $1 - \delta$. The positive $L_k$ is the Lipschitz constant of the kernel function $\kappa(\cdot)$, whose detailed expression depends on its specific choice. The domain related terms are defined as $\theta = \max_{x, x' \in \mathcal{X}} \|x - x'\|$. Note that this probabilistic upper bound for $h(\cdot)$ is based on reasonable hyperparameters of GPR, which are obtained through proper hyperparameter optimization methods.

**Remark 5:** To obtain a time-independent bound for the learning rates $\lambda_p$ and $\lambda_i$, (35) is employed to approximate $h_k \mu_k$, requiring the information of $h$. This approximation is very conservative, and an alternative exists by considering

$$\mu_k h_k = \mu_k^2 - \mu_k (\mu_k - h_k) \leq \mu_k^2 + \mu_k \eta_k.$$  

Since $\mu_k$ and $\eta_k$ are available at $t_k$, the time-varying upper bound for the learning rates is formulated as

$$\lambda_p + 4\lambda_i \mu_k h_k < \omega^{-2} \mu_k^{-1}(\mu_k + \eta_k)^{-1}/2.$$  

However, the time-invariant condition for the learning rates derived from (41) is more conservative than (32) by considering $\eta_k < \hat{h} + \eta$ for any time instant $k$ and

$$\sup_k \mu_k (\mu_k + \eta_k) = (\hat{h} + \eta)(\hat{h} + 2\eta) \geq \hat{h}(\hat{h} + \eta) = \sup_k h_k(h_k + \eta_k).$$

Therefore, the time-invariant bound in (32) is tighter, allowing larger adaption speed.

According to (32) in Theorem 2, the maximal stabilizable learning rates $\lambda_p$ and $\lambda_i$ are inverse related to the maximal uniform prediction error bound $\eta$. Intuitively, smaller $\eta$ allows larger $\lambda_p$ and $\lambda_i$, inducing faster adaption. The next subsection delves into the utilization of online learning techniques aimed at reducing prediction errors.

**C. Performance for Online Learning-based Control**

To lower the maximal prediction error bound $\eta$, online learning is employed, which for GPR can be realized by adding new data samples into the training data set. In this work, each training pair $\{x_k, y_k\}$ newly generated at any time instance $t_k$ is added into the training data set. The performance of this online learning-based adaptive PI controller is shown in the following corollary.

**Corollary 1:** Let all the assumptions in Theorem 2 hold. The Gaussian process regression model is updated by adding $\{x_k, y_k\}$ into $\mathcal{D}$ at $t_k$, and then infer the unknown Jacobian $h(\cdot)$ at $x_k$. Choose the learning rates $\lambda_p$ and $\lambda_i$ satisfying

$$\lambda_p + 4\lambda_i \leq \omega^{-2} \hat{h}^{-1}(\hat{h} + \eta)^{-1}/2$$

and then the controlled system achieves exponential stability.

**Proof:** We will prove that the posterior variance after adding new samples is bounded by $\sigma_n^2$. According to the definition of the kernel function and the assumption that
x_k \neq x_{k-1}$, the result is derived by considering a single sample data set with $\mathcal{D}_k = \{x_k, y_k\}$, then the posterior variance at $x_k$ is bounded as

$$
\sigma^2_k(x_k) < \kappa(x_k, x_k) - \frac{\kappa^2(x_k, x_k)}{\kappa(x_k, x_k) + \sigma_n^2} < \sigma_n^2.
$$

(44)

Then, the real data set at $t_k$ after adding new training sample, i.e., $\mathcal{D}_k = \{x_i, y_i\}_{i=0}^k$, is considered. Note that adding any training sample decreases the posterior variance by considering the result from sequential Gaussian process [45], where the update for $\sigma(x_k)$ is written as

$$
\sigma^2_{new}(x_k) = \sigma^2_{old}(x_k)
$$

- \left( \kappa(x_k, x') - \kappa'_{X_{old}}^T (K_{old} + \sigma_n^2 I_{|\mathcal{D}_{old}|})^{-1} \kappa'_{X_{old}} \right)^2
$$

$$
\leq \sigma^2_{old}(x_k),
$$

(45)
in which $\sigma_{new}(x_k)$ and $\sigma_{old}(x_k)$ are the posterior variances at $x_k$ from GPR models with data sets $\mathcal{D}_{new}$ and $\mathcal{D}_{old}$, respectively. The new data set includes one more training sample $\{x', y\}'$ than $\mathcal{D}_{old}$, i.e., $\mathcal{D}_{new} = \{\mathcal{D}_{old}, \{x', y\}'\}$. The corresponding kernel vectors $\kappa'_{X_{old}}$ and $\kappa'_{X_{old}}$ and Gramm matrix $K_{old}$ are defined under $\mathcal{D}_{old}$, where $\kappa'_{X_{old}}$ and $\kappa'_{X_{old}}$ are evaluated at $x'$ and $x_k$ respectively. Therefore, the upper bound of $\sigma(\cdot)$ by $\sigma_n$ is derived leading to $\eta(x_k)$ bounded as

$$
\eta(x_k) < \sqrt{\beta(X, \delta, \tau)} \sigma_n + \gamma(X, \delta, \tau)
$$

(46)

with the probability of at least $1 - \delta$.

Online learning provides a potentially smaller error bound, denoted as $\eta$ rather than $\eta$, as shown in Corollary 1. However, continuous data collection increases the size of the data set linearly w.r.t time, inducing increasing computation time for GPR with $O(|\mathcal{D}|^3)$, which is essential for real-time control for PMSMs. To bound the number of training samples $|\mathcal{D}|$ such that the computation time for GPR online update and prediction is bounded by the sampling interval $T = t_{k+1} - t_k$, for $k$, deletion strategies are used to remove some old training samples from the data set. Here, the prediction and control performance with deletion strategies is used as in [46], which is reformulated to fit the current scenario as shown below.

**Lemma 3:** Let all the assumptions in Corollary 1 hold. Employ the online learning strategy for GPR by collecting newly generated samples into the training data set. Moreover, execute any deletion strategy before adding new data, then the controlled system achieves exponential stability by using the adaptive PI control under GPR-based coefficient update with learning rates satisfying (43).

**Proof:** The result is intuitive by considering (44) holds for a data set with a single training sample.

Although Lemma 3 shows that any deletion strategy is suitable to guarantee stability under (43), smart deletion strategies, e.g., the age-based data method [47] and the information-theoretic approach [48], benefit the adaption of the system by inducing a tighter (smaller) upper bound of the posterior variance $\sigma(\cdot)$, allowing larger learning rates. From another perspective, setting the learning rates satisfying (43) with $\eta$ allows the usage of a single point data set, which saves the data storage and accelerates the computation speed at GPR.

With this single-point data set, the GPR-based PI controller is applicable for deployment on the microprocessor with few resources, including computational power and data storage.

**IV. SIMULATIONS AND EXPERIMENTS**

**A. Implementation**

This section explains the overall structure of the GPR method used in PMSM controller tuning. The standard process for PMSM control comprises four primary blocks at the top. The target speed is provided to the speed controller to reduce tracking errors by calculating a reference q-current. The discrepancy between the reference current and the feedback current is then routed to the current controller. The current error is mitigated through the calculation of a reference voltage, leading to the output of a target current at each time step. Subsequently, the target voltage is tracked using the space vector pulse width modulation (SVPWM) strategy to generate the appropriate PWM signals.

After elucidating the fundamental blocks of PMSM control, the subsequent step involves the GPR-assisted part. The real-time tracking error of the PMSM at times $k$ and $k-1$, using the collected data, is fed into the GPR model to apply an online point-adding procedure. To distinguish between the GPR-online and GPR-offline methods in PMSM, the key factor is whether the online point-adding procedure is utilized. The prediction of gradient values for controller tuning is inferred from the GPR Model. Following the principles of Lyapunov, the specified learning rate is restricted by a limiting block before being integrated into the designed update scheme. The final step entails applying the controller tuning scheme and adjusting the control parameters of the speed controller.

**B. Simulation Result**

1) The Anti-Load Control Results within Stable Range:

This section demonstrates the simulation-based implementation of our proposed control method using Simulink. A comparison is drawn between three control methods: the conventional PI controller, the GPR-based control method with online point addition (GPR-PI), and the GPR-based control method without point addition (Online-GPR-PI). To assess the performance of the proposed method under various load conditions, loads of 0.3 Nm and 0.2 Nm are applied. The PMSM used in the simulation is characterized by the parameters presented in Table I. This particular PMSM has low
inertia, resulting in larger overshoots compared to a standard PMSM, making it suitable for evaluating the effectiveness of our proposed method in reducing overshoot.

The stable simulation results for the GPR methods are depicted in Fig. 2. Based on the simulation results, it is apparent that the incorporation of the GPR method can enhance the performance of the baseline controller. Additionally, the inclusion of the new point addition procedure can lead to a modest improvement in the performance of the standard GPR method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>d axis Inductance</td>
<td>0.00024 H</td>
</tr>
<tr>
<td>Resistance</td>
<td>0.12 Ohm</td>
</tr>
<tr>
<td>Inertia</td>
<td>0.00008 gm(^2)</td>
</tr>
</tbody>
</table>

Conversely, the term “inferior neighbor” denotes the lower boundary of the box, calculated as \((Q_1 \times 1.5 \times IQR)\), where \(Q_1\) is the first quartile. The count of data points exceeding either the inferior or superior neighbor defines the number of outliers.

To demonstrate the superiority of the GPR method, we also present the performance of the RBFNN method [49]–[51] in both the low-speed and high-speed domains. It is evident that beyond 800 RPM, this method does not exhibit as favorable results as in the low-speed domain. Its performance in terms of recovery time is insufficient to compensate for its deficiencies in the high-speed region, as shown in Fig. 5 and Table III.

2) The Comparison of Stable Range: From a stability standpoint, it is essential to compare the performance of GPR-PI and Online-GPR-PI. As illustrated in Fig. 3, when the same learning rate is employed, GPR-PI exhibits instability, whereas Online-GPR-PI maintains a strong load-disturbance rejection capability, underscoring the superiority of the online approach.

3) Monte Carlo Tests: In order to improve the generalizability of our findings, we employ Monte Carlo testing to evaluate a variety of tracking performance scenarios, as illustrated in the violin plot presented in Fig. 4. These tracking tasks encompass a range from 300 RPM to 2000 RPM, all of which are subjected to a sudden load disturbance of 0.3 Nm. The width of the violin plot indicates the quantity of individual numerical data points, while the height conveys the data distribution. The box plot provides valuable statistical information, with specific values elaborated in Table II. In this table, the term “superior neighbor” refers to the upper boundary of the box, calculated as \((Q_3 + 1.5 \times IQR)\), where \(Q_3\) represents the third quartile and \(IQR\) is the interquartile range.

3. Experiments

To demonstrate the superiority of the GPR method, we also present the performance of the RBFNN method [49]–[51] in both the low-speed and high-speed domains. It is evident that beyond 800 RPM, this method does not exhibit as favorable results as in the low-speed domain. Its performance in terms of recovery time is insufficient to compensate for its deficiencies in the high-speed region, as shown in Fig. 5 and Table III.

Fig. 5. Monte Carlo results of different tasks of speed tracking from 300 RPM to 2000 RPM. The spontaneous load is 0.3 Nm. (a) Violin Plot considering the density of data, the width of the violin means the number of data in this region. (b) Boxplot. Method 1: PI, Method 2: Online-GPR-PI, Method 3: GPR-PI. (The quantitative indices are shown in Table II)

<table>
<thead>
<tr>
<th>PERFORMANCE COMPARISON WITH MONTE CARLO TESTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>Superior neighbor</td>
</tr>
<tr>
<td>Inferior neighbor</td>
</tr>
<tr>
<td>Number of outliers</td>
</tr>
<tr>
<td>Max error</td>
</tr>
</tbody>
</table>

Number of Collected Points: 12600133

C. Experimental Set-Up

Fig. 6 depicts the experimental setup employed in this study. The controlled PMSM is driven by an STM32F405RGT6
control board, operating at a speed loop frequency of 250 Hz. On the opposing side, the traction motor is controlled by an ODrive Control Board featuring an embedded torque control algorithm. The motor control workstation utilized in the experiment was designed by STMicroelectronics.

The electrical angle is derived by monitoring the DAC output on an oscilloscope, while the phase current is measured using an ETA5540A current probe. The parameters of the surface-mounted PMSM under test are consistent with Table I. The control task executed in the experiment closely mirrored that of the simulation, involving a reference speed of 900rpm and external loads of 0.3Nm and 0.2Nm.

![Experimental Set-Up of Controlling the PMSM](image)

Fig. 6. Experimental Set-Up of Controlling the PMSM.

D. Experimental Result

1) Anti-disturbance after Torque Load: The stabilized design procedure, which had been verified through simulation, was implemented in the experimental setup. The control tasks involved setting the reference speed at 900 RPM and introducing an impromptu load of 0.3 Nm and 0.1 Nm on the opposite side of the motor. The performance of the various control methods is presented in Fig. 7 through Fig. 8. The compared control methods encompass PI control, GP with online point update (Online-GPR-PI) control, and GP without online point update (GPR-PI) control.

2) Anti-disturbance after Removing Torque Load: Subsequently, the load 0.3 Nm and 0.1 Nm was removed, and the performance of the methods was analyzed, as presented in Fig. 9 - Fig. 10.

3) Comparison: The quantitative comparison of the proposed methods is presented in Table IV. This table offers an encompassing overview of control performance, with each metric representing an average value. Notably, the rise time is significantly shorter, measuring just one-third of the traditional PI method. In terms of overshoot, there is minimal disparity among the three methods. However, when examining the response time, both the Online-GPR-PI and GPR-PI methods demonstrate substantial enhancements, with Online-GPR-PI exhibiting superior performance in this regard. Taking all the metrics into account, it becomes evident that Online-GPR-PI surpasses GPR-PI, which, in turn, surpasses the PI method.

V. DISCUSSION AND CONCLUSION

A. Discussion

In this study, we propose an adaptive control approach building upon the established traditional PI controller. Beyond that, we harness data-driven techniques to amplify its efficacy. Specifically, we integrate a GPR-based controller with the pre-tuned PI controller employed by the original industry standards. The results unequivocally showcase the substantial enhancement achieved by GPR in elevating the motor’s performance throughout continuous online learning sequences. This advancement translates to a significant boost in its ability to withstand disturbances, with response times decreased by over three times. However, there exists untapped potential for refinement, especially in scenarios where the load is minimal or unloaded. In such cases, there is a tendency for larger overshoots. We attribute this to the inherent limitation of the initial dataset provided to GPR, which might not comprehensively capture sudden instances of heavy loads due to its higher proportion of stable data. Furthermore, the computational burden from GPR introduces a time factor, creating a form of inertia in the motor’s learning process, consequently leading to overshoots in certain situations. Nevertheless, the efficacy of GPR remains undeniable, as it exhibits swift learning and self-adaptation alongside conventional controllers, a pivotal aspect for real-world applications.

B. Conclusion

In this paper, we developed a pioneering GPR-based online tuning approach for the PI controller. GPR is harnessed to identify the unknown Jacobian relationship between speed and control input, thereby enhancing the tuner’s precision. Additionally, with the prediction error bound of GPR, the control performance for the proposed learning-based tuning is guaranteed. The effectiveness of this method is subsequently confirmed through experimental validation, both with and without load torque: The rise time experiences a reduction of over 2/3, and the average recovery time demonstrates a reduction exceeding 60%. These results underscore the substantial benefits offered by the proposed method in enhancing the control performance of PMSMs.

### References


Fig. 7. Performance of adding spontaneous load. Overcome (a) 0.3Nm load using the primary PI controller. (b) 0.3Nm load using Online-GPR-PI controller to assist the primary controller. (c) 0.3Nm load using the GPR controller to assist the primary controller.

Fig. 8. Performance of adding spontaneous load. Overcome (a) 0.1Nm load using the primary PI controller. (b) 0.1Nm load using Online-GPR-PI controller to assist the primary controller. (c) 0.1Nm load using the GPR controller to assist the primary controller.

Fig. 9. Performance of removing spontaneous load. Overcome (a) 0.3Nm load using the primary PI controller. (b) 0.3Nm load using Online-GPR-PI controller to assist the primary controller. (c) 0.3Nm load using the GPR controller to assist the primary controller.

Fig. 10. Performance of removing spontaneous load. Overcome (a) 0.1Nm load using the primary PI controller. (b) 0.1Nm load using Online-GPR-PI controller to assist the primary controller. (c) 0.1Nm load using the GPR controller to assist the primary controller.


