Constrained Optimization-Based Neuro-Adaptive Control (CONAC) for Synchronous Machine Drives Under Voltage Constraints

1st Myeongseok Ryu and 4th Kyunghwan Choi
Department of Mechanical and Robotics Engineering
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
dding_98@gm.gist.ac.kr and khchoi@gist.ac.kr

2nd Niklas Monzen, 3rd Pascal Seitter and 5th Christoph M. Hackl
Laboratory for Mechatronic and Renewable Energy Systems (LMRES)
Hochschule München (HM) University of Applied Sciences
Munich, Germany
{niklas.monzen, pascal.seitter, christoph.hackl}@hm.edu

***Abstract*—This paper presents a constrained optimization-based neuro-adaptive control (CONAC) for nonlinear synchronous machines (SMs) under voltage constraints, which allows to control the completely unknown electrical drive system, after a brief learning phase with very satisfactory control performance. The artificial neural network (ANN) in the proposed neuro-adaptive controller (NAC) learns online and empowers the controller to handle parameter uncertainties. Moreover, it solves a constrained optimization problem which allows to consider the nonlinear voltage constraints as well, by deriving the adaptation laws of the ANN's weights from the Lagrangian function. Boundedness of tracking error, convergence of the ANN weights, and satisfaction of the constraints are guaranteed by Lyapunov theory. Numerical simulations in combination with a realistic (nonlinear) synchronous machine drive demonstrate the effectiveness and robustness against parameter and modeling uncertainties of the proposed NAC and its very acceptable constraints handling.**

***Index Terms*—Synchronous Machine Drives, Constrained Optimization, Neuro-Adaptive Control**

NOTATION

In this paper, the following notation is used:

- $\otimes$ denotes the Kronecker product [1, Definition 7.1.2].
- $\boldsymbol{x} = [x_i]_{i\in\{1,\cdots,n\}} \in \mathbb{R}^n$ denotes a vector.
- $\text{vec}(\boldsymbol{A}) := [\text{row}_1(\boldsymbol{A}^\top), \cdots, \text{row}_m(\boldsymbol{A}^\top)]^\top$ for $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ where $\text{row}_i(\boldsymbol{A})$ denotes the i^{th} row of matrix $\boldsymbol{A}$.
- $\boldsymbol{I}_{n\times n}$ denotes the identity matrix of size $n\times n$ and $\boldsymbol{O}_{n\times m}$ denotes the zero matrix of size $n \times m$.

I. INTRODUCTION

A. Motivation

Synchronous machines (SMs) are widely used in various industrial applications, such as electric vehicles and robotics, due to their high efficiency and robustness [2]. For control of SMs, different nonlinear control strategies are developed, such as a nonlinear proportional-integral controller [3, 4], funnel control [5] or model predictive control [6]. However, these control designs typically require exact system knowledge of e.g., resistance(s) and flux linkages, which usually depend nonlinearly on temperature or current and rotor position, respectively. Moreover, the required data is usually not available, which makes system identification a crucial but not trivial and (very) time-consuming task. In addition, if the controllers are not properly tuned and/or the physical constraints such as current and voltage limits are not considered, the control performance may deteriorate significantly [7].

In summary, due to the inevitable limitations above, advanced nonlinear control designs that learn on-line are desirable to cope with unknown systems while guaranteeing operation within physical constraints.

B. Related Works

To overcome the aforementioned limitations, several literatures introduced neuro-adaptive control (NAC), which provides a promising solution for controlling SMs, as the incorporated artificial neural networks (ANNs) can learn the system dynamics and uncertainties online. The ANNs are generally used as estimators in the indirect adaptive control scheme, where they estimate the unknown system parameters (e.g., inductance, flux linkages, etc.) [8] or dynamics [9, 10]. It is notable that as the NACs are based on the adaptive control theory, they are able to guarantee stability and convergence of the tracking error in the sense of Lyapunov, even during the learning phase. However, there is still room for improvement in the existing NACs for SMs.

First, since most of the existing NAC approaches integrate the ANNs in the conventional controller, they have to deal with the large number of tuning parameters, which makes the design and tuning process cumbersome. For the examples of the integrations, ANNs have been integrated into the backstepping scheme in [9, 11]. In [8], the ANNs are used to estimate and cancel the effect of the unknown dynamics coupling in the Lyapunov stability framework. Especially, in [10], the ANN is utilized to compensate for the unknown dead-zone and nonlinear functions in the funnel control scheme for servo mechanisms. This allows to maintain the specific properties of

This work was supported in part by GIST-IREF from Gwangju Institute of Science and Technology (GIST), was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00554087).

the conventional control design, such as optimality, robustness, and stability, even though the system is unknown. However, these integrations are not straightforward and often lead to complex designs with many tuning parameters.

Furthermore, most of the existing NACs assume that the control input saturation is not present or is neglected in the design. In other words, to the best of the authors' knowledge, there is no existing NAC in SMs that considers the voltage saturation in the adaptation law derivation, while several NAC literatures for general systems introduced auxiliary systems to address the control input saturation in the adaptation law derivation directly [12, 13]. Only a few works consider the voltage saturation with additional control design. For instance, in [14], the saturated robust optimal control is used to avoid the voltage saturation, using the estimated position and speed of ANN-based observers. Despite the promising results, the complexity of the control design and the number of tuning parameters are significantly high.

C. Contributions

To the best of the authors' knowledge, the online adaptation law derivation by constrained optimization theory with stability guarantees in a unified framework has not yet been developed. Motivated by the discussion above, we propose a novel constrained optimization-based neuro-adaptive control (CONAC) for the SMs. Main contributions of this paper are:

- No prior system information is required for controller design, tuning and implementation, since the ANN in the CONAC approximates the unknown system dynamics and unknown system parameters online;
- As the entire desired control law is approximated by the ANN, the CONAC has less tuning parameters than existing NAC designs;
- The adaptation laws are derived based on a constrained optimization problem, which is reformulated from the original control problem, by considering the boundedness of ANN weight and input (voltage) saturation as corresponding inequality constraints; and
- Using Lyapunov theory, the boundedness of control (tracking) error and ANN weights is guaranteed. The physical constraints are handled properly as well.

D. Synchronous Machine Dynamics

We consider nonlinear SMs, modeled by [15, Chapter 14],

$$\operatorname{sat}(\boldsymbol{u}_{\mathrm{s,ref}}^{dq}) = \boldsymbol{R}_{\mathrm{s}}^{dq}\boldsymbol{i}_{\mathrm{s}}^{dq} + \omega_{\mathrm{p}}\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{s}}^{dq} + \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq} \tag{1}$$

with $\boldsymbol{J} := \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, stator voltages reference to voltage source inverter (VSI) $\boldsymbol{u}_{\mathrm{s,ref}}^{dq} := (u_{\mathrm{s,ref}}^{d}, u_{\mathrm{s,ref}}^{q})^{\top} \in \mathbb{R}^2$, stator currents $\boldsymbol{i}_{\mathrm{s}}^{dq} := (i_{\mathrm{s}}^{d}, i_{\mathrm{s}}^{q})^{\top} \in \mathbb{R}^2$, stator resistance matrix $\boldsymbol{R}_{\mathrm{s}}^{dq} := \boldsymbol{R}_{\mathrm{s}}^{dq}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \phi_{\mathrm{p}}, \omega_{\mathrm{p}}) \in \mathbb{R}^{2\times 2}$, and electrical angle $\phi_{\mathrm{p}} \in \mathbb{R}$ and angular velocity $\omega_{\mathrm{p}} \in \mathbb{R}$. The stator voltages are saturated due to physical limitations of the utilized VSI by the saturation function $\operatorname{sat}(\cdot)$ such that $\|\operatorname{sat}(\boldsymbol{u}_{\mathrm{s,ref}}^{dq})\| \leq u_{\max} \in \mathbb{R}_{>0}$, where $u_{\max}$ denotes the maximum voltage amplitude. In general, the stator flux linkages $\boldsymbol{\psi}_{\mathrm{s}}^{dq} := \boldsymbol{\psi}_{\mathrm{s}}^{dq}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \phi_{\mathrm{m}}) \in \mathbb{R}^2$, depend on $\boldsymbol{i}_{\mathrm{s}}^{dq}$ and $\phi_{\mathrm{m}} \in \mathbb{R}$ which is ϕ_{p} divided by pole pair number n_{p}. Therefore, the last term of (1) can be written as $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq} = [\frac{\partial}{\partial \boldsymbol{i}_{\mathrm{s}}^{dq}}\boldsymbol{\psi}_{\mathrm{s}}^{dq}]\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i}_{\mathrm{s}}^{dq} + [\frac{\partial}{\partial \phi_{\mathrm{m}}}\boldsymbol{\psi}_{\mathrm{s}}^{dq}]\frac{\mathrm{d}}{\mathrm{d}t}\phi_{\mathrm{m}}$, where $\frac{\partial}{\partial \boldsymbol{i}_{\mathrm{s}}^{dq}}\boldsymbol{\psi}_{\mathrm{s}}^{dq}$ is typically referred to differential inductance matrix $\boldsymbol{L}_{\mathrm{s}}^{dq} := \boldsymbol{L}_{\mathrm{s}}^{dq}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \phi_{\mathrm{m}}) \in \mathbb{R}^{2\times 2}$. This allows to rewrite (1) as current dynamics

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i}_{\mathrm{s}}^{dq} = (\boldsymbol{L}_{\mathrm{s}}^{dq})^{-1}\big(\operatorname{sat}(\boldsymbol{u}_{\mathrm{s,ref}}^{dq}) - \boldsymbol{R}_{\mathrm{s}}^{dq}\boldsymbol{i}_{\mathrm{s}}^{dq} - \omega_{\mathrm{p}}\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{s}}^{dq} - \boldsymbol{\delta}_{\mathrm{s}}^{dq}\big), \tag{2}$$

where $\boldsymbol{\delta}_{\mathrm{s}}^{dq} := [\frac{\partial}{\partial \boldsymbol{i}_{\mathrm{s}}^{dq}}\boldsymbol{\psi}_{\mathrm{s}}^{dq}]\frac{\mathrm{d}}{\mathrm{d}t}\phi_{\mathrm{m}} \in \mathbb{R}^2$ is the induced voltage due to harmonics in the flux linkages.

For convenience, (2) is rewritten in standard form

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, \operatorname{sat}(\boldsymbol{u})) \tag{3}$$

with state variable $\boldsymbol{x} := \boldsymbol{i}_{\mathrm{s}}^{dq} \in \mathbb{R}^2$, control input $\boldsymbol{u} := \boldsymbol{u}_{\mathrm{s,ref}}^{dq} \in \mathbb{R}^2$ and system function $\boldsymbol{f}(\boldsymbol{x}, \operatorname{sat}(\boldsymbol{u})) := (\boldsymbol{L}_{\mathrm{s}}^{dq})^{-1}(\operatorname{sat}(\boldsymbol{u}_{\mathrm{s,ref}}^{dq}) - \boldsymbol{R}_{\mathrm{s}}^{dq}\boldsymbol{i}_{\mathrm{s}}^{dq} - \omega_{\mathrm{p}}\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{s}}^{dq} - \boldsymbol{\delta}_{\mathrm{s}}^{dq}) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$. In addition, let $\bar{u}$ denotes the maximum voltage amplitude $u_{\max}$. For the following, let $\boldsymbol{B}(\boldsymbol{x}, \operatorname{sat}(\boldsymbol{u}))) := \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \operatorname{sat}(\boldsymbol{u}))}{\partial \operatorname{sat}(\boldsymbol{u})}$ which denotes the Jacobian matrix (also known as the input matrix) of the system function $\boldsymbol{f}(\boldsymbol{x}, \operatorname{sat}(\boldsymbol{u}))$ with respect to $\operatorname{sat}(\boldsymbol{u})$.

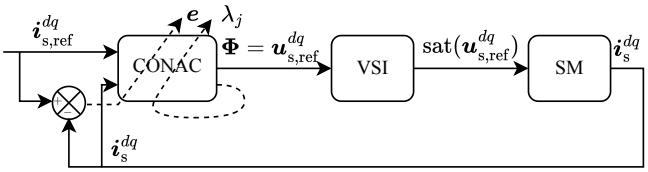


Figure 1: Schematic of the CONAC.

E. Control Objective

The objective of this paper is to design a CONAC that guarantees reference tracking of a continuously differentiable reference signal $\boldsymbol{x}_{\mathrm{ref}} := \boldsymbol{i}_{\mathrm{s,ref}}^{dq}(t) : \mathbb{R} \to \mathbb{R}^2$ under the voltage saturation introduced above. The following assumption is imposed:

Assumption 1. *The reference signal $\boldsymbol{x}_{\mathrm{ref}}$ is (essentially) bounded and available to design a feasible control input in the presence of the control input saturation.*

II. Controller Design

This section details the design of the novel CONAC, which is illustrated in Fig. 1. First, a few mathematical preliminaries are restated. Then, the NAC design is introduced. Later, the adaptation laws are derived based on a constrained optimization, resulting in CONAC.

A. Mathematical preliminaries

Later, the following proposition will be used:

Proposition 1 (see [1, Proposition 7.1.9])**.** *For matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and vector $\boldsymbol{b} \in \mathbb{R}^n$, we have the following property*

$$\boldsymbol{A}^{\top}\boldsymbol{b} = \operatorname{vec}(\boldsymbol{A}^{\top}\boldsymbol{b}) = \operatorname{vec}(\boldsymbol{b}^{\top}\boldsymbol{A}) = (\boldsymbol{I}_m \otimes \boldsymbol{b}^{\top})\operatorname{vec}(\boldsymbol{A}).$$

Moreover, exploiting physical SM properties, the following fact and assumption are considered:

Fact 1. *The input matrix $\boldsymbol{B}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u})))$ is the inverse of $\boldsymbol{L}_{\mathrm{s}}^{dq}$ (recall (2)). Therefore, $\boldsymbol{B}(\boldsymbol{x}) := \boldsymbol{B}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u})))$ is a positive definite nonsingular matrix which only depends on $\boldsymbol{x}$, since $\boldsymbol{L}_{\mathrm{s}}^{dq}$, itself is a positive definite matrix [15, Chapter 14].*

Assumption 2. *[16, Assumption 1(b)] There exists an unknown positive function $\beta(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}_{>0}$ such that $\frac{1}{2}\boldsymbol{v}^\top[\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{B}(\boldsymbol{x})^{-1}]\boldsymbol{v} \le \beta(\boldsymbol{x})\|\boldsymbol{v}\|^2$ for all $\boldsymbol{v} \in \mathbb{R}^2$.*

Remark 1. *[16, Remark 1 (b)] Assumption 2 is not restrictive, since only the existence of $\beta(\boldsymbol{x})$ is assumed, regardless of its sign.*

B. Neuro-Adaptive Control Design

Before NAC design, we first define a desired control input $\boldsymbol{u}^* \in \mathbb{R}^2$ which will be approximated by the ANN. Motivated by the control design introduced in [16], consider the Lyapunov function candidate $V_1 := \frac{1}{2}\boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\boldsymbol{e} \ge 0$, where $\boldsymbol{e} := \boldsymbol{x} - \boldsymbol{x}_{\mathrm{ref}}$ denotes the tracking error of the state. The time derivative of V_1 is obtained as follows

$$\begin{aligned}\tfrac{\mathrm{d}}{\mathrm{d}t}V_1 =& \boldsymbol{e}^\top\boldsymbol{B}^{-1}(\boldsymbol{x})\big(\boldsymbol{f}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u})) - \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}}\big) + \tfrac{1}{2}\boldsymbol{e}^\top\tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{B}^{-1}(\boldsymbol{x})\boldsymbol{e}\\ \le& \boldsymbol{e}^\top\boldsymbol{B}^{-1}(\boldsymbol{x})(\boldsymbol{f}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u})) - \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}}) + \beta(\boldsymbol{x})\|\boldsymbol{e}\|^2\\ =& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\boldsymbol{B}^{-1}(\boldsymbol{x})\cdot\\ &\times\underbrace{\big(k\boldsymbol{B}(\boldsymbol{x})\boldsymbol{e} + \boldsymbol{f}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u})) - \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}} + \beta(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{e}\big)}_{=:\bar{\boldsymbol{f}}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u}),\boldsymbol{x}_{\mathrm{ref}},\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}})},\end{aligned} \tag{4}$$

where $k > 0$ is an arbitrary (unknown) positive constant gain. According to (4), if we can realize a control input $\boldsymbol{u}^*$ such that $\bar{\boldsymbol{f}}(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u}^*),\boldsymbol{x}_{\mathrm{ref}},\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}}) = \boldsymbol{0}$, then the tracking error $\boldsymbol{e}$ exponentially converges to zero, i.e., $\frac{\mathrm{d}}{\mathrm{d}t}V_1 \le -k\|\boldsymbol{e}\|^2$.

To define a desired stabilizing control input $\boldsymbol{u}^*$, one can use the implicit function theorem [17, see, p. 651]. Using the implicit function theorem, one can conclude that there exists $\boldsymbol{u}^*$ such that $\bar{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{u}^*,\boldsymbol{x}_{\mathrm{ref}},\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}}) = \boldsymbol{0}$ by invoking $\frac{\partial\bar{\boldsymbol{f}}(\cdot)}{\partial\,\mathrm{sat}(\boldsymbol{u})} = \frac{\partial\boldsymbol{f}(\cdot)}{\partial\,\mathrm{sat}(\boldsymbol{u})} = \boldsymbol{B}(\boldsymbol{x})$ and using Fact 1 (i.e., $\boldsymbol{B}(\boldsymbol{x})$ is nonsingular $\forall\boldsymbol{u} \in \mathbb{R}^2$). Note that, according to Assumption 1, $\boldsymbol{u}^*$ can be supposed to be not saturated such that $\|\boldsymbol{u}^*\| \le \bar{u}$ and $\mathrm{sat}(\boldsymbol{u}^*) = \boldsymbol{u}^*$.

Next, the mean value theorem [17, see, p. 651] is applied to define $\boldsymbol{u}_l := c\,\mathrm{sat}(\boldsymbol{u}) + (1-c)\boldsymbol{u}^*$ for $c \in [0,1]$, such that $\bar{\boldsymbol{f}}(\mathrm{sat}(\boldsymbol{u})) - \bar{\boldsymbol{f}}(\boldsymbol{u}^*) = \frac{\partial\bar{\boldsymbol{f}}(\cdot)}{\partial\,\mathrm{sat}(\boldsymbol{u})}\big|_{\boldsymbol{u}=\boldsymbol{u}_l}(\mathrm{sat}(\boldsymbol{u}) - \boldsymbol{u}^*)$, where the arguments $(\boldsymbol{x},\mathrm{sat}(\boldsymbol{u}),\boldsymbol{x}_{\mathrm{ref}},\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{ref}})$ of $\bar{\boldsymbol{f}}(\cdot)$ are suppressed as $(\mathrm{sat}(\boldsymbol{u}))$ for brevity. Adding $-\boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\bar{\boldsymbol{f}}(\boldsymbol{u}^*) = \boldsymbol{0}$ in the right side of (4) yields

$$\begin{aligned}\tfrac{\mathrm{d}}{\mathrm{d}t}V_1 \le& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\bar{\boldsymbol{f}}(\mathrm{sat}(\boldsymbol{u}))\\ =& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\big(\bar{\boldsymbol{f}}(\mathrm{sat}(\boldsymbol{u})) - \bar{\boldsymbol{f}}(\boldsymbol{u}^*)\big)\\ =& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\big(\tfrac{\partial\bar{f}(\cdot)}{\partial\,\mathrm{sat}(\boldsymbol{u})}\big|_{\boldsymbol{u}=\boldsymbol{u}_l}(\mathrm{sat}(\boldsymbol{u}) - \boldsymbol{u}^*)\big)\\ =& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\boldsymbol{B}(\boldsymbol{x})^{-1}\big(\boldsymbol{B}(\boldsymbol{x})(\mathrm{sat}(\boldsymbol{u}) - \boldsymbol{u}^*)\big)\\ =& -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\big(\mathrm{sat}(\boldsymbol{u}) - \boldsymbol{u}^*\big)\\ \le& -k\|\boldsymbol{e}\|^2 + 2\bar{u}\|\boldsymbol{e}\|\\ \le& -k(\|\boldsymbol{e}\| - \tfrac{\bar{u}}{k})^2 + \tfrac{\bar{u}^2}{k}.\end{aligned} \tag{5}$$

According to (5), $\frac{\mathrm{d}}{\mathrm{d}t}V_1$ is negative definite when $\boldsymbol{e}$ leaves a compact set $\{\boldsymbol{e} \in \mathbb{R}^2 \mid \|\boldsymbol{e}\| \le \frac{2\bar{u}}{k}\}$.

Inspired by [12] an ANN is employed to approximate the entire desired control input $\boldsymbol{u}^*$. Since the SM controller must run with high sampling rates, only a simple ANN $\boldsymbol{\Phi} : \mathbb{R}^{l_0} \times \mathbb{R}^{\Xi} \to \mathbb{R}^2$ with one hidden layer is selected. It is represented as

$$\boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}) = \boldsymbol{W}_1^\top\boldsymbol{\phi}(\boldsymbol{W}_0^\top\boldsymbol{x}_n)$$

where $\boldsymbol{W}_i \in \mathbb{R}^{(l_i+1)\times l_{i+1}}$ for $i \in \{0,1\}$ is the weighting matrix of i^{th} layer, $\boldsymbol{\phi} : \mathbb{R}^{l_1} \to \mathbb{R}^{l_1+1}$ is the activation function, and $\boldsymbol{x}_n \in \mathbb{R}^{l_0+1}$ is the input vector. The input, hidden, and output layers have l_0, l_1, and l_2 nodes, respectively. For simplicity, the weighting matrices are vectorized as $\boldsymbol{\theta}_i := \mathrm{vec}(\boldsymbol{W}_i) \in \mathbb{R}^{\Xi_i}$ for $i \in \{0,1\}$ where $\Xi_i := (l_i+1)l_{i+1}$, and augmented as $\boldsymbol{\theta} := \big(\boldsymbol{\theta}_0^\top,\boldsymbol{\theta}_1^\top\big)^\top \in \mathbb{R}^{\Xi}$, where $\Xi = \Xi_0+\Xi_1$. The activation function is defined as element-wise nonlinear function $\sigma(\cdot) : \mathbb{R} \to \mathbb{R}$ and augmentation 1 to account for bias term in the weighting matrices such that $\boldsymbol{\phi}(\boldsymbol{x}) = \big(\sigma(x_1),\cdots,\sigma(x_{l_1}),1\big)^\top$ for some vector $\boldsymbol{x} = (x_1,\cdots,x_{l_1})^\top \in \mathbb{R}^{l_1}$. The nonlinear function $\sigma(\cdot)$ is chosen as $\tanh(\cdot)$ in this paper for the boundedness of its output and derivative.

According to the universal approximation theorem [18], ANNs whose activation functions are sigmoidal functions (e.g., sigmoid or $\tanh(\cdot)$) can approximate any continuous nonlinear function $\boldsymbol{g}(\boldsymbol{x}_n)$ with a small bounded approximation error $\bar{\epsilon} \in \mathbb{R}_{>0}$ in a compact set $\boldsymbol{x}_n \in \Omega$ such that $\sup_{\boldsymbol{x}_n\in\Omega}\|\boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}^*) - \boldsymbol{g}(\boldsymbol{x}_n)\| = \bar{\epsilon} < \infty$, where $\boldsymbol{\theta}^* := (\boldsymbol{\theta}_0^{*\top},\boldsymbol{\theta}_1^{*\top})^\top \in \mathbb{R}^{\Xi}$ denotes the ideal weighting vector. Since the ideal weighting vector is unknown, the estimate of weighting vector $\widehat{\boldsymbol{\theta}} := (\widehat{\boldsymbol{\theta}}_0^\top,\widehat{\boldsymbol{\theta}}_1^\top)^\top \in \mathbb{R}^{\Xi}$ is used in the controller. The ideal and estimated control inputs are represented as follows

$$\boldsymbol{u}^* = \boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}^*) + \boldsymbol{\epsilon}, \tag{6a}$$

$$\boldsymbol{u} = \boldsymbol{\Phi}(\boldsymbol{x}_n;\widehat{\boldsymbol{\theta}}), \tag{6b}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^2$ is the bounded approximation error vector such that $\|\boldsymbol{\epsilon}\| \le \bar{\epsilon} < \infty$.

Substituting the control inputs (6a) and (6b) in (5), the time derivative of V_1 is obtained as follows

$$\tfrac{\mathrm{d}}{\mathrm{d}t}V_1 \le -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top(\mathrm{sat}(\widehat{\boldsymbol{\Phi}}) - \boldsymbol{\Phi}^* - \boldsymbol{\epsilon}) \tag{7}$$

where $\widehat{\boldsymbol{\Phi}} := \boldsymbol{\Phi}(\boldsymbol{x}_n;\widehat{\boldsymbol{\theta}})$ and $\boldsymbol{\Phi}^* := \boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}^*)$. This con-

cludes that $\boldsymbol{e}$ can be driven to zero by adapting $\widehat{\boldsymbol{\theta}}$ such that $\boldsymbol{\Phi}(\boldsymbol{x}_n;\widehat{\boldsymbol{\theta}}) \to \boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}^*)$ if $\|\boldsymbol{\epsilon}\|$ is sufficiently small. It is notable that, since the equality $\boldsymbol{\Phi}(\boldsymbol{x}_n;\widehat{\boldsymbol{\theta}}) = \boldsymbol{\Phi}(\boldsymbol{x}_n;\boldsymbol{\theta}^*)$ does not hold in learning phase, only the boundedness of $\boldsymbol{e}$ will be ensured in Section III. Importantly, input (voltage) saturation is not yet considered; this will be done next.

C. Constrained Optimization-Based Adaptation Law

The adaptation law is derived from the constrained optimization theory. First, the control problem is reformulated into the following constrained optimization problem:

$$\min_{\widehat{\boldsymbol{\theta}}} J(\boldsymbol{e};\widehat{\boldsymbol{\theta}}) = \tfrac{1}{2}\boldsymbol{e}^\top \boldsymbol{e},$$

$$\text{subject to} \quad \begin{cases} c_{\theta_i}(\widehat{\boldsymbol{\theta}}) = \frac{1}{2}(\widehat{\boldsymbol{\theta}}_i^\top \widehat{\boldsymbol{\theta}}_i - \bar{\theta}_i^2) \le 0, & \forall i \in \{0,1\} \\ c_u(\widehat{\boldsymbol{\theta}}) = \frac{1}{2}(\widehat{\boldsymbol{\Phi}}^\top \widehat{\boldsymbol{\Phi}} - \bar{u}^2) \le 0 \end{cases} \tag{8}$$

where $J : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is the objective function, and $c_{\theta_i} : \mathbb{R}^{\Xi_i} \times \mathbb{R}^{\Xi_i} \to \mathbb{R}$ for $i \in \{0,1\}$ and $c_u : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ are the inequality constraints for the weights and the control input, and $\bar{\theta}_i \in \mathbb{R}_{>0}$ and $\bar{u}$ are the positive maximum values of $\|\widehat{\boldsymbol{\theta}}_i\|$ and $\|\boldsymbol{u}\|$, respectively. The estimated weighting vector $\widehat{\boldsymbol{\theta}}$ is the optimization variable of (8). The corresponding Lagrangian function of (8) is defined by

$$L(\boldsymbol{e},\widehat{\boldsymbol{\theta}},[\lambda_j]_{j\in\mathcal{I}}) := J(\boldsymbol{e};\widehat{\boldsymbol{\theta}}) + \textstyle\sum_{j\in\mathcal{I}} \lambda_j c_j(\widehat{\boldsymbol{\theta}}) \tag{9}$$

where $\mathcal{I} := \{\theta_0, \theta_1, u\}$ is a set of imposed inequality constraint functions and $\lambda_j \in \mathbb{R}_{\ge 0}$, $\forall j \in \mathcal{I}$ are the corresponding Lagrangian multipliers.

To solve the dual problem of (9) (i.e., $\min_{\widehat{\boldsymbol{\theta}}} \max_{[\lambda_j]_{j\in\mathcal{I}}} L(\boldsymbol{e},\widehat{\boldsymbol{\theta}},[\lambda_j]_{j\in\mathcal{I}})$), the adaptation law is derived as follows:

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\widehat{\boldsymbol{\theta}} = -\alpha\tfrac{\partial L}{\partial \widehat{\boldsymbol{\theta}}} = -\alpha\big(\tfrac{\partial J}{\partial \widehat{\boldsymbol{\theta}}} + \textstyle\sum_{j\in\mathcal{I}} \lambda_j \tfrac{\partial c_j}{\partial \widehat{\boldsymbol{\theta}}}\big) \tag{10a}$$

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\lambda_j = \beta_j \tfrac{\partial L}{\partial \lambda_j} = \beta_j c_j, \quad \forall j \in \mathcal{I} \tag{10b}$$

$$\lambda_j = \max(\lambda_j, 0), \tag{10c}$$

where $\alpha \in \mathbb{R}_{>0}$ and $\beta_j \in \mathbb{R}_{>0}$ are the adaptation gain and update rate of the corresponding Lagrange multipliers. Note that the Lagrange multipliers are always semi-positive due to (10c).

In the adaptation law (10a), one needs to calculate the partial derivative of the state vector with respect to the control input $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} \in \mathbb{R}^{2\times 2}$ (also called the sensitivity function [17, see, Sec. 3.3]) for $\frac{\partial J}{\partial \widehat{\boldsymbol{\theta}}} = \Big((\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{u}})(\frac{\partial \boldsymbol{u}}{\partial \widehat{\boldsymbol{\theta}}})\Big)^\top \boldsymbol{e} = \Big((\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}})(\frac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}})\Big)^\top \boldsymbol{e}$. The exact method to obtain the sensitivity function is the forward sensitivity method [19] which simulates the sensitivity equation $\frac{\mathrm{d}}{\mathrm{d}t}(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}) = \frac{\partial}{\partial \boldsymbol{u}}(\boldsymbol{f}(\boldsymbol{x}, \mathrm{sat}(\boldsymbol{u})))$. However, the sensitivity equation cannot be simulated, since the system functions are unknown and in view of the present saturation. Furthermore, simulating the sensitivity equation requires large memory allocation and computational cost, since the number of weights is generally large for ANNs.

To overcome this issue, the sensitivity matrix should be approximated. The simplest approximation is to use the sign of each entry (i.e., control direction) of the Jacobian $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}$ as $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} \approx [\mathrm{sign}(\frac{\partial x_i}{\partial u_j})]_{i,j\in\{1,2\}}$ [20]. However, for the given system, the control direction is unknown, but the positive or negative definiteness of the control input matrix is known. Therefore, in view of Fact 1 the sensitivity matrix is approximated as $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} \approx \boldsymbol{I}_2$ and the adaptation law can be simplified as follows:

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\widehat{\boldsymbol{\theta}} = -\alpha\big(\tfrac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}}^\top \boldsymbol{e} + \textstyle\sum_{j\in\mathcal{I}} \lambda_j \tfrac{\partial c_j}{\partial \widehat{\boldsymbol{\theta}}}\big). \tag{11}$$

The gradients of the ANN with respect to weights can be calculated using Proposition 1 and the chain rule as follows

$$\tfrac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}} = \begin{bmatrix} \frac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}_0} & \frac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}_1} \end{bmatrix} \in \mathbb{R}^{2\times\Xi},$$

where $\frac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}_0} = (\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)$, $\frac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}_1} = \widehat{\boldsymbol{W}}1^\top \widehat{\boldsymbol{\phi}}'(\boldsymbol{I}_{l_1} \otimes \boldsymbol{x}_n^\top)$, and $\widehat{\boldsymbol{\phi}} := \boldsymbol{\phi}(\widehat{\boldsymbol{W}}_0^\top \boldsymbol{x}_n)$ and $\widehat{\boldsymbol{\phi}}' := \frac{\mathrm{d}\widehat{\boldsymbol{\phi}}}{\mathrm{d}(\widehat{\boldsymbol{W}}_0^\top \boldsymbol{x}_n)}$. [MS: Some gradient should be replaced with total derivative.] The remaining partial derivative terms in (11) are calculated as follows:

$$\tfrac{\partial c_{\theta_0}}{\partial \widehat{\boldsymbol{\theta}}} = \begin{pmatrix} \widehat{\boldsymbol{\theta}}_0 \\ \boldsymbol{0}_{\Xi_1\times 1} \end{pmatrix}, \quad \tfrac{\partial c_{\theta_1}}{\partial \widehat{\boldsymbol{\theta}}} = \begin{pmatrix} \boldsymbol{0}_{\Xi_0\times 1} \\ \widehat{\boldsymbol{\theta}}_1 \end{pmatrix}, \quad \tfrac{\partial c_u}{\partial \widehat{\boldsymbol{\theta}}} = \tfrac{\partial \widehat{\boldsymbol{\Phi}}}{\partial \widehat{\boldsymbol{\theta}}}^\top \widehat{\boldsymbol{\Phi}}.$$

III. Stability Analysis

The following theorem shows the boundedness of $\boldsymbol{e}$ and $\widehat{\boldsymbol{\theta}}$.

Theorem 1. *For dynamical system (3), the neuro-adaptive controller (6b), and weight adaptation law (10) ensures boundedness of tracking error $\boldsymbol{e}(\cdot)$ and weights $\widehat{\boldsymbol{\theta}}(\cdot)$ while the imposed constraints $c_{\theta_i}(\cdot)$ $i \in \{0,1\}$ and $c_u(\cdot)$ are considered.*

Proof. The stability analysis is conducted by the Lyapunov stability theory. Without loss of generality, weight constraint $c_{\theta_i}, \forall i \in \{0,1\}$ is assumed to be violated. This is because, even if $c_{\theta_i}, \forall i \in \{0,1\}$ is satisfied, $\widehat{\boldsymbol{\theta}}_i$, $\forall i \in \{0,1\}$ is adapted to reduce $J(\cdot)$ without considering the constraint, unless c_{θ_i} is violated. Therefore, this does not affect the stability analysis. We consider two cases: control input saturation is active and control input saturation is not active.

Case 1: Control input saturation is active.

According to the result of (5), the time derivative of the Lyapunov function V_1 is negative when the tracking error $\boldsymbol{e}$ leaves the invariant set Θ_e^1 defined as $\Theta_e^1 := \{\boldsymbol{e} \in \mathbb{R}^2 \mid \|\boldsymbol{e}\| \le \frac{2\bar{u}}{k}\}$.

Consider the Lyapunov function candidate $V_2 := \frac{1}{2\alpha}\widehat{\boldsymbol{\theta}}_1^\top \widehat{\boldsymbol{\theta}}_1$ to investigate the boundedness of the outer layer's weight $\widehat{\boldsymbol{\theta}}_1$ and the satisfaction of c_{θ_1} and c_u. Using Proposition 1, the time derivative of V_2 is obtained as follows:

$$\begin{aligned} \tfrac{\mathrm{d}}{\mathrm{d}t}V_2 &= -\widehat{\boldsymbol{\theta}}_1^\top ((\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)^\top \boldsymbol{e} + \lambda_{\theta_1}\widehat{\boldsymbol{\theta}}_1 + \lambda_u(\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)^\top \widehat{\boldsymbol{\Phi}}) \\ &\le \|(\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)\|\|\widehat{\boldsymbol{\theta}}_1\|\|\boldsymbol{e}\| - \lambda_{\theta_1}\|\widehat{\boldsymbol{\theta}}_1\|^2 \\ &\quad - \lambda_u \widehat{\boldsymbol{\theta}}_1^\top (\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)^\top \underbrace{(\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)\widehat{\boldsymbol{\theta}}_1}_{=\widehat{\boldsymbol{\Phi}}} \\ &\le -(\lambda_{\theta_1} + \lambda_u c_1)\|\widehat{\boldsymbol{\theta}}_1\|^2 + c_2\|\boldsymbol{e}\|\|\widehat{\boldsymbol{\theta}}_1\| \\ &\le -(\lambda_{\theta_1} + \lambda_u c_1)\|\widehat{\boldsymbol{\theta}}_1\|^2 + \tfrac{2c_2\bar{u}}{k}\|\widehat{\boldsymbol{\theta}}_1\|. \end{aligned}$$

where $c_1 := \lambda_{\min}((\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)^\top (\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)) \in \mathbb{R}_{\geq 0}$ and $c_2 := \|(\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)\| \in \mathbb{R}_{\geq 0}$. In the same manner, the invariant set of $\widehat{\boldsymbol{\theta}}_1$ is defined as $\Theta^1_{\theta_1} := \{\widehat{\boldsymbol{\theta}}_1 \in \mathbb{R}^{\Xi_1} \mid \|\widehat{\boldsymbol{\theta}}_1\| \leq \frac{2c_2\bar{u}}{k(\lambda_{\theta_1}+\lambda_u c_1)}\}$. This implies that if λ_{θ_1} sufficiently increases by the constraint violation, (i.e., see, (10b)) c_{θ_1} can be satisfied. On the other hand, by multiplying c_2 to both sides of $\|\widehat{\boldsymbol{\theta}}_1\| \leq \frac{2c_2\bar{u}}{k(\lambda_{\theta_1}+\lambda_u c_1)}$, the following inequality is obtained:

$$\|\widehat{\boldsymbol{\Phi}}\| \leq \|\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}\|\|\widehat{\boldsymbol{\theta}}_1\| = c_2\|\widehat{\boldsymbol{\theta}}_1\| \leq \frac{2c_2^2\bar{u}}{k(\lambda_{\theta_1}+\lambda_u c_1)}.$$

Therefore, similarly, c_u can be satisfied by sufficiently large λ_u.

The boundedness of the inner layer's weight $\widehat{\boldsymbol{\theta}}_0$ and the satisfaction of c_{θ_0} can be shown in the same way as those of the outer layer.

Case 2: Control input saturation is not *active.*

In this case, the saturation function can be removed in (7). Consider the Lyapunov function candidate $V_3 := V_1 + V_2$. Using Proposition 1, the time derivative of V_3 is given by

$$\begin{aligned}
\tfrac{\mathrm{d}}{\mathrm{d}t}V_3 \leq & -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top(\widehat{\boldsymbol{\Phi}} - \boldsymbol{\Phi}^* - \epsilon) + \widehat{\boldsymbol{\theta}}_1^T \tfrac{\mathrm{d}}{\mathrm{d}t}\widehat{\boldsymbol{\theta}}_1/\alpha \\
= & -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\widehat{\boldsymbol{\Phi}} + \boldsymbol{e}^\top(-\boldsymbol{\Phi}^* - \epsilon) \\
& - \widehat{\boldsymbol{\theta}}_1^\top(\boldsymbol{I}_{l_2} \otimes \widehat{\boldsymbol{\phi}}^\top)^\top\boldsymbol{e} - \lambda_{\theta_1}\widehat{\boldsymbol{\theta}}_1^\top\widehat{\boldsymbol{\theta}}_1 \\
= & -k\|\boldsymbol{e}\|^2 + \boldsymbol{e}^\top\widehat{\boldsymbol{\Phi}} + \boldsymbol{e}^\top(-\boldsymbol{\Phi}^* - \epsilon) \\
& - \widehat{\boldsymbol{\Phi}}^\top\boldsymbol{e} - \lambda_{\theta_1}\widehat{\boldsymbol{\theta}}_1^\top\widehat{\boldsymbol{\theta}}_1 \\
\leq & -k\|\boldsymbol{e}\|^2 + \|\boldsymbol{\Phi}^* + \epsilon\|\|e\| - \lambda_{\theta_1}\|\widehat{\boldsymbol{\theta}}_1\|^2.
\end{aligned}$$

Therefore, one can conclude that $\boldsymbol{e}$ is bounded and $\widehat{\boldsymbol{\theta}}_1$ converges to zero until the weight constraint c_{θ_1} is satisfied. The invariant set of $\boldsymbol{e}$ is defined as $\Theta^2_e := \{\boldsymbol{e} \in \mathbb{R}^2 \mid \|\boldsymbol{e}\| \leq \frac{\bar{u}}{k}\}$ (see, (6a)). The boundedness of $\widehat{\boldsymbol{\theta}}_0$ and the satisfaction of c_{θ_0} can be shown similarly as Case 1.

In conclusion, the tracking error $\boldsymbol{e}$ and the estimated weights $\widehat{\boldsymbol{\theta}}$ are bounded, satisfying the imposed constraints c_{θ_i}, $\forall i \in \{0, 1\}$ and c_u.

□

IV. Implementation and Validation

The proposed CONAC is validated via numerical simulation in MATLAB & Simulink R2024b. The sampling time of the controller is 125 μs, and simulation sampling time is 100 times faster than the controller. The utilized interior permanent magnet synchronous motor's (IPMSM's) flux linkage maps are identified from a real machine in the laboratory and shown in Fig. 2. Further, system and simulation parameters are listed in Table I.

The reference current pattern of the simulation scenario consists of step signals whose amplitudes increase from zero to 4.19 A, and the step's time lengths are 40 ms. For the d-axis, the sign of the reference currents is negative, while for the q-axis, the reference sign alternates repeatedly between positive and negative values. To investigate cross-coupling effects, the

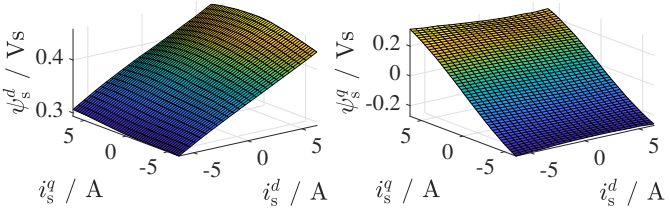


Figure 2: The stator flux linkages of the IPMSM at the rated mechanical speed $\omega_{\mathrm{m,R}} = 240.855$ rad/s.

Table I: IPMSM parameters.

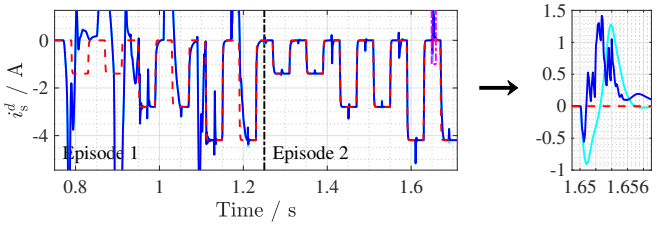
Symbol	Description	Value
$p_{\mathrm{m,R}}$	rated mech. power	2.5 kW
$\omega_{\mathrm{m,R}}$	rated mech. speed	240.855 rad/s
$m_{\mathrm{m,R}}$	rated mech. torque	10.5 Nm
$i^d_{\mathrm{s,max}}$	max d current	4.19 A
$i^q_{\mathrm{s,max}}$	max. q current	4.19 A
$u_{\max}$	max. voltage	340 V
R_{s}	stator resistance	1.475 Ω

q-axis reference current is applied 20 ms earlier than the d-axis reference step. To illustrate how the ANNs adjust their weights over time, the reference pattern is presented twice, labeled as Episode 1 and Episode 2.

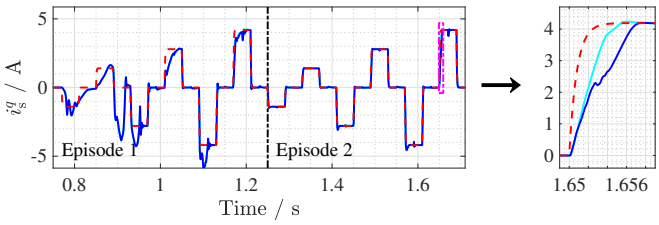
The operating point of the IPMSM is set to the rated mechanical speed of $\omega_{\mathrm{p}} = 240.855$ rad/s. However, due to the high mechanical speed, the system may become unstable at the beginning of implementation with the randomly initialized weights. To avoid this, the mechanical speed is gradually increased from zero to the rated mechanical speed over 0.5 s, while maintaining a zero current reference for warm start. Once the rated mechanical speed is reached, the reference current pattern is applied after a brief idle period of 0.25 s. Additionally, a low-pass filter (LPF) with a cutoff frequency of 200 Hz is used to smooth the transition by attenuating high-frequency components in the reference signal.

For the comparative study, the CONAC with control input constraint c_u (C_1) and without control input constraint (C_2) were simulated. The shared parameters of (C_1) and (C_2) are as follows: $l_0 = 4, l_1 = 32, l_2 = 2, \alpha = 30$ and $\beta_{\theta_0} = \beta_{\theta_1} = 10$. The update rate β_u is set as 5×10^{-3} for (C_1) and $\beta_u = 0$ for (C_2) so that the control input constraint is not imposed in (C_2). The input vector of the ANNs is defined as $\boldsymbol{x}_n = \left((\boldsymbol{i}^{dq}_{\mathrm{s}})^\top, (\boldsymbol{i}^{dq}_{\mathrm{s,ref}})^\top, 1\right)^\top \in \mathbb{R}^5$. The weights of the ANNs are uniformly initialized in the range of $[-10^{-5}, 10^{-5}]$. The parameters of constraints are set as $\bar{\theta}_0 = 12.649, \bar{\theta}_1 = 80$, and $\bar{u} = u_{\max}$.

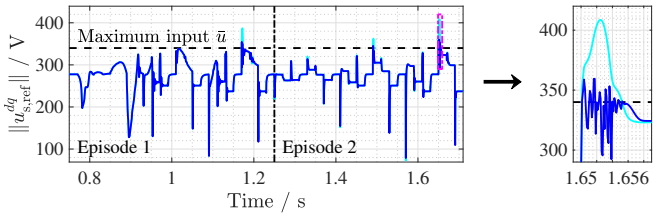
The quantitative evaluation of the tracking performance and the satisfaction of c_u is conducted by L_2 norm (i.e., $\sqrt{\int_0^T \|\zeta\|^2\,\mathrm{d}t}$, where T is the time duration and $\zeta \in \{i^d_{\mathrm{s}} -$



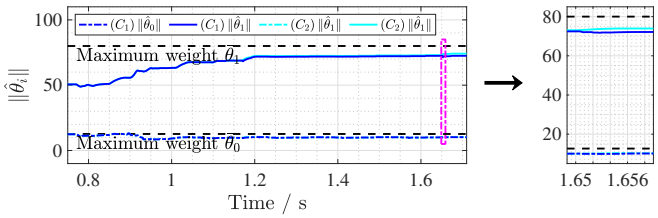
(a) Tracking result of d-axis stator current i_s^d.



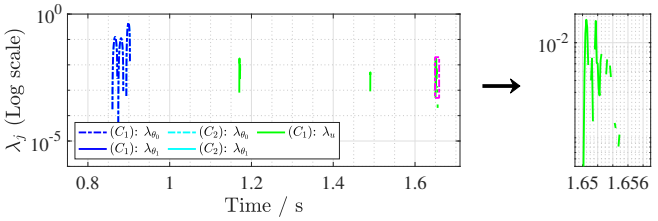
(b) Tracking result of q-axis stator current i_s^q.



(c) Norm of control input (stator voltages) $\boldsymbol{u}_{s,ref}^{dq}$.



(d) Norm of weights $\widehat{\boldsymbol{\theta}}_i$.



(e) Lagrange multipliers λ_j in logarithmic scale.

Figure 3: Simulation results of the CONAC with control input constraint (C_1) (blue) and without control input constraint (C_2) (cyan), and reference signal of $\boldsymbol{i}_s^{dq}$ (red dashed line) at the rated mechanical speed 240.855 rad/s.

$i_{s,ref}^d, i_s^q - i_{s,ref}^q, \max(c_u, 0)\}$).

A. Simulation Results

1) Effectiveness of Neural Network: The control results of (C_1) and (C_2) are shown in Fig. 3, and the evaluation of the tracking performance by L_2 norm is provided in Table II. For both cases (C_1) and (C_2), the ANNs were well-trained to track

Table II: Quantitative comparison of performances' L_2 norm.

	Episode 1			**Episode 2**		
	(C_1)	(C_2)	%	(C_1)	(C_2)	%
$i_s^d - i_{s,ref}^d$	1.835	1.835	+0.0	0.119	0.117	+1.3
$i_s^q - i_{s,ref}^q$	0.597	0.591	+1.0	0.157	0.126	+19.6
$\max(c_u, 0)$	0.297	2.031	−85.9	0.488	3.251	−85.0

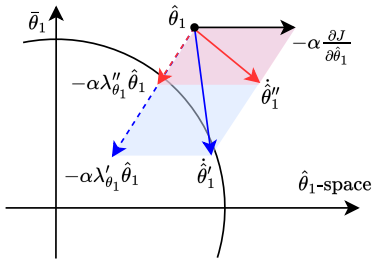


Figure 4: The effect of Lagrange multiplier λ_{θ_i} on the adaptation direction of $\widehat{\boldsymbol{\theta}}_i$. The notation $(\cdot')$ and $(\cdot'')$ represent two different cases of bigger and smaller Lagrange multiplier, respectively (i.e., $\lambda'_{\theta_1} > \lambda''_{\theta_1}$).

the reference signal of i_s^d (see Fig. 3a) and i_s^q (see Fig. 3b) over 2 episodes. At the beginning of Episode 1, the training results in a non-desired tracking performance. However, the ANNs quickly adapt their weights to track the reference signal by approximating the ideal control input. The L_2 norm of i_s^d and i_s^q are reduced by 93.5% and 73.7% for (C_1) and 93.6% and 78.7% for (C_2), respectively, during Episode 2 (see Fig. 3a and 3b). Furthermore, the settling time of i_s^d is about 5 ms and 6 ms for i_s^q for both controllers during Episode 2. Note that the slightly slow settling time of i_s^q in Episode 2 (see right-hand side of Fig. 3b) is due to the control input constraint c_u. Moreover, the overshoots of the i_s^d and i_s^q and the cross-coupling effects were significantly mitigated in Episode 2 after training during Episode 1.

2) Constraint Handling: The constraint handling is now investigated. In Fig. 3d and Fig. 3e, the norms of $\widehat{\boldsymbol{\theta}}_i, \forall i \in \{0, 1\}$ and the corresponding $\lambda_{\theta_i}, \forall i \in \{0, 1\}$ are shown. According to (10b), λ_{θ_i} was generated from zero when the norm of $\widehat{\boldsymbol{\theta}}_i$ exceeds the constraint $\bar{\theta}_i$ and altered the adaptation direction defined in (10a) to drive $\widehat{\boldsymbol{\theta}}_i$ towards more satisfactory points. The influence of λ_{θ_i} on $\widehat{\boldsymbol{\theta}}_i$ is illustrated in Fig. 4. When the norm of $\widehat{\boldsymbol{\theta}}_i$ was within the constraint (i.e., $c_{\theta_i} < 0$), the Lagrange multiplier λ_{θ_i} decreases to zero, i.e., see, (10b).

Similarly, the constraint handling of control input saturation can be investigated. Without constraints on the control input, the control input of (C_2) significantly exceeds the limit $\bar{u}$, as shown in Fig. 3c. This allowed (C_2) to make assertive control decisions resulting in a slightly faster settling time of i_s^q and 19.6% smaller L_2 norm of i_s^q during Episode 2 (see right-hand side of Fig. 3b). However, since (C_2) does not have any

information about the saturation, it increases the control input excessively large to track the reference signal despite the fact that the control input is saturated and distorted. On the other hand, (C_1) not only attempts to track the reference signal but also considers the control input constraint. Therefore, for case (C_1), the proposed CONAC adapts its weights with respect to the tracking errors in both axes with lower c_u violation. As a result, L_2 norm of the control input violation is reduced by 85.9% and 85.0% during Episode 1 and Episode 2, respectively, compared to (C_2). It is worth noting that there is still a small violation of c_u in (C_1) during Episode 2, which is due to the discrete control of a continuous system.

V. Conclusion

Constrained optimization-based neuro-adaptive control (CONAC) was proposed for synchronous machine (SM) drives with input voltage saturation. The CONAC is designed to track a typical stator current reference signal by approximating the ideal control input by a artificial neural network (ANN) with one hidden layer. The weight adaptation law is derived by constrained optimization theory to ensure boundedness of tracking error and weights. The high-fidelity simulation results of a realistic (nonlinear) SM show that the CONAC can track the reference signal and adapt its weights and control action within the constraints.

As future work, the parameter dependencies of the CONAC will be investigated in more detail to select an optimal ANN topology. Furthermore, the CONAC will be implemented in a real-time control system to validate the effectiveness of the proposed method and its feasibility in the laboratory.

References

[1] D. S. Bernstein, *Matrix mathematics: theory, facts, and formulas*. Princeton university press, 2009.

[2] S. Sakunthala, R. Kiranmayi, and P. N. Mandadi, "A study on industrial motor drives: Comparison and applications of pmsm and bldc motor drives," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 537–540, IEEE, 2017.

[3] N. Monzen and C. M. Hackl, "Optimal reference voltage saturation for nonlinear current control of synchronous machine drives," in *2024 IEEE 21st International Power Electronics and Motion Control Conference (PEMC)*, pp. 1–6, IEEE, 2024.

[4] C. M. Hackl, M. J. Kamper, J. Kullick, and J. Mitchell, "Nonlinear PI current control of reluctance synchronous machines," 2015.

[5] C. M. Hackl, "Current PI-funnel control with anti-windup for synchronous machines," in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015)*, (Osaka, Japan), pp. 1997–2004, Institute of Electrical and Electronics Engineers (IEEE), dec 2015.

[6] I. Hammoud, S. Hentzelt, K. Xu, T. Oehlschlaegel, M. Abdelrahem, C. M. Hackl, and R. Kennel, "On continuous set model predictive control of permanent magnet synchronous machines," *IEEE Transactions on Power Electronics*, pp. 1–1, 2022.

[7] N. Monzen and C. M. Hackl, "Optimal reference voltage saturation for nonlinear current control of synchronous machine drives," *Authorea Preprints*, 2024.

[8] H. Jie, G. Zheng, J. Zou, X. Xin, and L. Guo, "Adaptive decoupling control using radial basis function neural network for permanent magnet synchronous motor considering uncertain and time-varying parameters," *IEEE Access*, vol. 8, pp. 112323–112332, 2020.

[9] F. F. M. El-Sousy, M. F. El-Naggar, M. Amin, A. Abu-Siada, and K. A. Abuhasel, "Robust adaptive neural-network backstepping control design for high-speed permanent-magnet synchronous motor drives: Theory and experiments," *IEEE Access*, vol. 7, pp. 99327–99348, 2019.

[10] Shubo Wang, Haisheng Yu, Jinpeng Yu, Jing Na, and Xuemei Ren, "Neural-network-based adaptive funnel control for servo mechanisms with unknown dead-zone,"

[11] J. Yu, P. Shi, W. Dong, B. Chen, and C. Lin, "Neural network-based adaptive dynamic surface control for permanent magnet synchronous motors," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 640–645, 2015.

[12] K. Esfandiari, F. Abdollahi, and H. Talebi, "A stable nonlinear in parameter neural network controller for a class of saturated nonlinear systems," *International Federation of Automatic Control Proceedings Volumes (IFAC)*, vol. 47, no. 3, pp. 2533–2538, 2014.

[13] G. Peng, C. Yang, W. He, and C. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3138–3148, 2019.

[14] L. N. Tan, T. P. Cong, and D. P. Cong, "Neural network observers and sensorless robust optimal control for partially unknown pmsm with disturbances and saturating voltages," *IEEE Transactions on Power Electronics*, vol. 36, no. 10, pp. 12045–12056, 2021.

[15] C. M. Hackl, *Non-identifier based adaptive control in mechatronics: Theory and Application*. No. 466 in Lecture Notes in Control and Information Sciences, Berlin: Springer International Publishing, 2017.

[16] A. Boulkroune, M. Tadjine, M. M'Saad, and M. Farza, "Fuzzy adaptive controller for mimo nonlinear systems with known and unknown control direction," *Fuzzy Sets and Systems*, vol. 161, no. 6, pp. 797–820, 2010. Theme: "Fuzzy Control".

[17] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, NJ: Prentice-Hall, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel.

[18] A. R. Barron, "Universal approximation bounds for superposition of a sigmoidal function," *IEEE Transactions on Information theory*, vol. 39, no. 3, pp. 930–945, 1993.

[19] B. Sengupta, K. J. Friston, and W. D. Penny, "Efficient gradient computation for dynamical models," *Neuroimage*, vol. 98, pp. 521–7, 2014.

[20] M. Saerens and A. Soquet, "Neural controller based on back-propagation algorithm," *IEE Proceedings F (Radar and Signal Processing)*, vol. 138, pp. 55–62, 1991.