

Imposing a Weight Norm Constraint for Neuro-Adaptive Control

Myeongseok Ryu¹, Jiyun Kim², and Kyunghwan Choi¹

Abstract—In this paper, a neuro-adaptive controller with weight norm constraints is proposed for uncertain Euler–Lagrange systems. The boundedness of the weights in the neuro-adaptive controller design is important to prevent excessively large control inputs and system instability. To ensure the boundedness of the weights, the weight norm constraints are imposed as inequality constraints in the weight adaptation. The adaptation law is derived based on the constrained optimization method. The stability of the proposed controller is analyzed in the sense of Lyapunov, ensuring the boundedness of the tracking error and weight estimation. For the comparative study, two benchmark controllers and the proposed controller were evaluated through a numerical simulation of a two-link manipulator system and compared in terms of tracking performance and parameter dependency. The comparative study verified that the proposed controller has better tracking performance and lower parameter dependency.

NOTATION

In this study, the following notation is used:

- \otimes denotes the Kronecker product [1, Definition 7.1.2].
- $\mathbf{x} = [x_i]_{i \in \{1, \dots, n\}} \in \mathbb{R}^n$ denotes a vector.
- $\text{row}_i(\mathbf{A})$ denotes the i^{th} row of the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$.
- $\text{vec}(\mathbf{A}) := [\text{row}_1(\mathbf{A}^\top), \dots, \text{row}_m(\mathbf{A}^\top)]^\top$ for $\mathbf{A} \in \mathbb{R}^{n \times m}$.
- $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.
- \mathbf{I}_n denotes the $n \times n$ identity matrix and $\mathbf{0}_{n \times m}$ denotes the $n \times m$ zero matrix.

I. INTRODUCTION

In various practical applications, Euler–Lagrange systems generally have uncertainties due to their unknown and unmodeled dynamics. These uncertainties degrade the control performance index. Furthermore, if uncertainties dominate the system, they may lead to instability. By compensating for these uncertainties, adaptive control methods have been widely used to attenuate the effects of system uncertainties [2], [3]. Most conventional adaptive control methods focus on estimating unknown parameters in systems or controllers.

More recently, neuro-adaptive control approaches have been developed using neural networks (NNs) to approximate

unknown system dynamics [4]. It uses the approximation capability of NNs with any arbitrary sigmoidal activation function [5], which can approximate sufficiently smooth functions with minimum approximation error through ideal weights in a compact set. This property allows the NNs to be synthesized in the controllers to approximate the unknown dynamics. Various architectures of NNs have been proposed for neuro-adaptive control, such as single-hidden-layer NN (SHLNN) [6], [7], radial basis function NN (RBFNN) [8], [9] and deep NN (DNN) [10]. The literature has shown that NNs can improve the system's performance index and stability by approximating the unknown dynamics.

One of the common issues in using NNs is that the outputs of NNs are not predictable. This is because the input–output relationships of the NNs are not interpretable (i.e., the NNs are usually called black boxes [11], [12]). This issue should be addressed for safety because the unpredictable control input may reach the actuator's physical limitations and result in system collapse.

Most studies modify their adaptation laws to ensure the boundedness of the weights, noting that the boundedness of the NNs' output can be ensured by the boundedness of the weights. In [9], [10], the projection operator is utilized to prevent weight divergence by projecting the adaptation direction on the convex set of weights. However, in the literature, the convex set is usually selected to be as large as possible since there is no information on the ideal weights' norm. Hence, the projection operator guarantees only the boundedness of the weights without theoretical optimality. In addition, the σ -modification [8] and the ϵ -modification [6], [7] are widely used to regulate the magnitude of the weights by adding a stabilizing function in the adaptation law, thereby making the invariance set of the estimation error of the weights. The existing methods have shown their effectiveness in ensuring the boundedness of the weights via numerical simulations. However, they lack theoretical analysis regarding the optimality of the adapted weights.

Similarly, approaches that regulate the magnitude of the weights have also been introduced in the deep learning literature. One of the approaches is L_2 -regularization, which adds the squared magnitudes of the weights to the objective function [13], [14]. Then, the adaptation process attempts to reduce the magnitude of the weights, not only the original objective function. By regulating the magnitude of the weights, the stability of the adaptation process can be enhanced, and overfitting can be prevented. However, L_2 -regularization also involves a trade-off between adaptation stability and the optimality of the weights.

For the theoretical analysis of weight optimality with

*This work was supported by a Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0020535, The Competency Development Program for Industry Specialist) (Corresponding author: Kyunghwan Choi)

¹Myeongseok Ryu and Kyunghwan Choi are with the School of Mechanical and Robotics Engineering, Gwangju Institute of Science and Technology, 61005 Gwangju, Republic of Korea dding_98@gm.gist.ac.kr, khchoi@gist.ac.kr

²Jiyun Kim is with the AI Graduate School, Gwangju Institute of Science and Technology, 61005 Gwangju, Republic of Korea jiyun6606@gm.gist.ac.kr

boundedness, the constrained optimization method [15] can be utilized. It provides the theoretical definition of optimality and the numerical methodology to solve the constrained optimization problem. To the best of the authors' knowledge, no prior work has utilized the constrained optimization method for the real-time weight adaptation of neuro-adaptive control satisfying weight boundedness. Therefore, the constrained optimization method could play a pivotal role in the neuro-adaptive control design addressing the weight boundedness.

The main contributions of this study are as follows:

- The neuro-adaptive control problem is reformulated into a constrained optimization problem by treating the satisfaction of the weight boundedness as inequality constraints.
- The adaptation law is derived based on the constrained optimization method to minimize the objective function while satisfying the weight norm constraints.
- The stability of the adaptation laws is analyzed via Lyapunov stability analysis, which ensures the boundedness of the tracking error and weight estimation.

The remainder of this paper is organized as follows: Section II presents the target-constrained system and control objective. Section III introduces the proposed controller and the adaptation law. Section IV examines the stability of the proposed controller. A comparative study of the three selected controllers, including the proposed controller through numerical simulation, is reported in Section V. Finally, Section VI concludes the paper by presenting future work.

II. PROBLEM FORMULATION

A. Model Dynamics and Control Objective

Consider an uncertain Euler-Lagrange system modeled as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{V}_m\dot{\mathbf{q}} + \mathbf{F} + \mathbf{G} + \boldsymbol{\tau}_d = \boldsymbol{\tau} \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^n$ and $\boldsymbol{\tau} \in \mathbb{R}^n$ denote the generalized coordinate and the control input, respectively; $\mathbf{M} := \mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$, $\mathbf{V}_m := \mathbf{V}_m(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$, and $\mathbf{G} := \mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ denote the unknown system function matrices; and $\mathbf{F} := \mathbf{F}(\dot{\mathbf{q}}) \in \mathbb{R}^n$ denotes the external force. Using the user-designed matrices $\widehat{\mathbf{M}} > 0$, $\widehat{\mathbf{V}}_m$ and $\widehat{\mathbf{G}}$, (1) can be represented as

$$\widehat{\mathbf{M}}\ddot{\mathbf{q}} + \widehat{\mathbf{V}}_m\dot{\mathbf{q}} + \widehat{\mathbf{G}} = \boldsymbol{\tau} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad (2)$$

where $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) := -(\mathbf{M} - \widehat{\mathbf{M}})\ddot{\mathbf{q}} - (\mathbf{V}_m - \widehat{\mathbf{V}}_m)\dot{\mathbf{q}} - (\mathbf{G} - \widehat{\mathbf{G}}) - \mathbf{F}$ denotes the residual unknown term.

Hence, the objective of the control design is to make \mathbf{q} track the continuously differentiable desired trajectory $\mathbf{q}_d := \mathbf{q}_d(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ under the unknown terms $\mathbf{f} := \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$.

III. CONTROL LAW DEVELOPMENT

In this section, the neuro-adaptive controller is developed. Section III-A presents the details of the neuro-adaptive controller and the NN model. The adaptation law based on the constrained optimization method is derived in Section III-B by formulating a constrained optimization problem.

A. Neuro-adaptive Control Design

The backstepping control-based approach is utilized to generate a reference signal $\mathbf{z}^* := -k_q\tilde{\mathbf{q}} + \dot{\mathbf{q}}_d$ for $\mathbf{z} := \dot{\mathbf{q}}$, where $\tilde{\mathbf{q}} := \mathbf{q} - \mathbf{q}_d$ and $k_q \in \mathbb{R}_{>0}$. The desired stabilizing controller can be designed as follows:

$$\boldsymbol{\tau}^* = -\widehat{\mathbf{M}} \cdot (k_z\tilde{\mathbf{z}}) + (-\widehat{\mathbf{M}}\tilde{\mathbf{q}} + \widehat{\mathbf{V}}_m\mathbf{z} + \widehat{\mathbf{G}} - \mathbf{f} + \widehat{\mathbf{M}}\dot{\mathbf{z}}^*) \quad (3)$$

where $\tilde{\mathbf{z}} := \mathbf{z} - \mathbf{z}^*$ and $k_z \in \mathbb{R}_{>0}$. Note that the desired controller cannot be realized because of \mathbf{f} .

To approximate the desired controller, an NN is utilized. [MS: Even though the DNNs have a higher approximation capability than the SHLNNs [16], here, the SHLNN is utilized for simplicity and low computational complexity.] The NN with a single hidden layer is represented as

$$\Phi(\mathbf{q}_{NN}; \boldsymbol{\theta}) := \mathbf{W}_1^\top \phi(\mathbf{W}_0^\top \mathbf{q}_{NN})$$

where $\mathbf{q}_{NN} \in \mathbb{R}^{l_0+1}$ denotes the NN input vector, $\mathbf{W}_i \in \mathbb{R}^{(l_i+1) \times l_{i+1}}$, $i \in \{0, 1\}$ denotes the weight matrix of the i^{th} layer and $\phi : \mathbb{R}^{l_1} \rightarrow \mathbb{R}^{l_1+1}$ denotes the activation function layer. The activation function layer consists of an elementwise nonlinear function $\sigma(\cdot)$ and an augmented 1 to combine the bias terms in the weight matrix (i.e., $\phi(\mathbf{x}) = (\sigma(\mathbf{x}_{(1)}), \dots, \sigma(\mathbf{x}_{(l_1)}), 1)^\top$). For further simplicity, let $\boldsymbol{\theta}_i := \text{vec}(\mathbf{W}_i) \in \mathbb{R}^{\Xi_i}$ denotes the vectorized weights and $\boldsymbol{\theta} := (\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_0^\top)^\top \in \mathbb{R}^\Xi$ denote the total weight vector, where $\Xi_i = (l_i + 1) \cdot l_{i+1}$ and $\Xi = \Xi_0 + \Xi_1$ denote the number of each layer and total weights, respectively.

Using this NN, the desired controller $\boldsymbol{\tau}^*$ can be approximated by the ideal weight vector $\boldsymbol{\theta}^*$ for a compact subset $\Omega_{NN} \in \mathbb{R}^{l_0+1}$ to ϵ accuracy such that $\sup_{\mathbf{q}_{NN} \in \Omega_{NN}} \|\Phi(\mathbf{q}_{NN}; \boldsymbol{\theta}^*) - \boldsymbol{\tau}^*\| = \epsilon < \infty$ [5]. The ideal weight vector $\boldsymbol{\theta}^*$ is typically assumed to be bounded. Then, using the estimated weight vector $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1^\top, \hat{\boldsymbol{\theta}}_0^\top)^\top$ of $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_1^{*\top}, \boldsymbol{\theta}_0^{*\top})^\top$, the desired controller $\boldsymbol{\tau}^* \approx -\Phi(\mathbf{q}_{NN}; \boldsymbol{\theta}^*) - \epsilon$ with bounded approximation error $\epsilon \in \mathbb{R}^2$, can be approximated as follows:

$$\boldsymbol{\tau} := -\Phi(\mathbf{q}_{NN}; \hat{\boldsymbol{\theta}}). \quad (4)$$

For further sections, let $\Phi^* := \Phi(\mathbf{q}_{NN}; \boldsymbol{\theta}^*)$ and $\phi^* := \phi(\mathbf{W}_0^{*\top} \mathbf{q}_{NN})$, and let $\hat{\Phi} := \Phi(\mathbf{q}_{NN}; \hat{\boldsymbol{\theta}})$, $\hat{\phi} := \phi(\hat{\mathbf{W}}_0^\top \mathbf{q}_{NN})$ and $\hat{\phi}' := \frac{\partial \hat{\phi}}{\partial \hat{\mathbf{W}}_0^\top \mathbf{q}_{NN}}$.

Using (2), (3), and (4), the error dynamics can be obtained as the first-order system of augmented error $\boldsymbol{\xi} := (\tilde{\mathbf{q}}^\top, \tilde{\mathbf{z}}^\top)^\top \in \mathbb{R}^{2n}$ as follows:

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_\xi \boldsymbol{\xi} + \mathbf{B}_\xi (\Phi^* - \hat{\Phi} + \epsilon)$$

and

$$\mathbf{A}_\xi := \begin{bmatrix} -k_q \mathbf{I}_n & \mathbf{I}_n \\ -\mathbf{I}_n & -k_z \mathbf{I}_n \end{bmatrix}, \quad \mathbf{B}_\xi := \begin{bmatrix} \mathbf{0}_{n \times n} \\ \widehat{\mathbf{M}}^{-1} \end{bmatrix}.$$

B. Adaptation Law Derivation

[MS: As discussed in Section I, the boundedness of the weights should be addressed to prevent excessively large control inputs and system instability.] For the boundedness of the weights, weight norm constraints are imposed on the adaptation process such that $c_{\theta_i} = \|\hat{\boldsymbol{\theta}}_i\|^2 - \bar{\theta}_i^2$, $i \in \{0, 1\}$.

The control problem can be reformulated into a constrained optimization problem as follows:

$$\min_{\hat{\theta}} J(\xi; \hat{\theta}) := \frac{1}{2} \xi^\top \Lambda \xi$$

$$\text{subject to } c_j \leq 0, \quad j \in \mathcal{I} = \{\theta_0, \theta_1\},$$

where $\Lambda = \Lambda^\top > 0$ denotes the weighting matrix. In this optimization problem, ξ is considered a predefined parameter. The corresponding Lagrangian function is defined as

$$L(\xi, \hat{\theta}, [\lambda_j]_{j \in \mathcal{I}}) := J(\xi; \hat{\theta}) + \sum_{j \in \mathcal{I}} \lambda_j c_j(\hat{\theta}).$$

To solve the dual problem $\min_{\hat{\theta}} \max_{[\lambda_j]_{j \in \mathcal{I}}} L$, the adaptation law is derived as follows:

$$\frac{d}{dt} \hat{\theta} = -\alpha \frac{\partial L}{\partial \hat{\theta}} = -\alpha \left(\frac{\partial J}{\partial \hat{\theta}} + \sum_{j \in \mathcal{I}} \lambda_j \frac{\partial c_j}{\partial \hat{\theta}} \right), \quad (5a)$$

$$\frac{d}{dt} \lambda_j = \beta_j \frac{\partial L}{\partial \lambda_j} = \beta_j c_j, \quad \forall j \in \mathcal{I}, \quad (5b)$$

$$\lambda_j = \max(\lambda_j, 0), \quad (5c)$$

where arguments of L are suppressed for brevity, and $\alpha \in \mathbb{R}_{>0}$ and $\beta_j \in \mathbb{R}_{>0}$ denote the adaptation gain and the update rate for each Lagrange multiplier, respectively.

Using the chain rule, the gradient of the objective function with respect to the weights (i.e., $\frac{\partial J}{\partial \hat{\theta}}$) in (5a) can be represented as

$$\frac{\partial J}{\partial \hat{\theta}} = \frac{\partial \xi}{\partial \hat{\theta}}^\top \Lambda \xi.$$

The calculation of $\frac{\partial J}{\partial \hat{\theta}}$ is not straightforward because of the dynamics of ξ . Using the forward sensitivity method presented in [17], the sensitivity equation can be obtained as follows:

$$\frac{d}{dt} \eta = A_\xi \eta - B_\xi \frac{\partial \hat{\Phi}}{\partial \hat{\theta}}, \quad (6)$$

where $\eta := \frac{\partial \xi}{\partial \hat{\theta}} \in \mathbb{R}^{2n \times \Xi}$ denotes the sensitivity of the weights to the augmented error. The initial value of η is zero since the initial ξ is independent of the weights. By decomposing for each layer, the dynamics of $\eta_i := \frac{\partial \xi}{\partial \hat{\theta}_i} \in \mathbb{R}^{2n \times \Xi_i}$ can be represented as

$$\begin{aligned} \frac{d}{dt} \eta &= [\eta_1 \quad \eta_0]^\top \\ &= A_\xi [\eta_1 \quad \eta_0] - B_\xi [(I_{l_2} \otimes \hat{\Phi}^\top) \widehat{W}_1^\top \hat{\Phi}' (I_{l_1} \otimes q_{NN}^\top)]. \end{aligned}$$

The calculation of $\frac{\partial \hat{\Phi}}{\partial \hat{\theta}}$ is introduced in [10]. In conclusion, the gradient of the objective with respect to the weights can be obtained as $\frac{\partial J}{\partial \hat{\theta}} = \eta^\top \Lambda \xi$ by simulating the sensitivity equation (6).

On the other hand, the gradient of the constraints with respect to the weights can be represented as follows:

$$\frac{\partial c_{\theta_0}}{\partial \hat{\theta}} = \begin{bmatrix} \mathbf{0}_{\Xi_1 \times 1} \\ 2\hat{\theta}_0 \end{bmatrix}, \quad \frac{\partial c_{\theta_1}}{\partial \hat{\theta}} = \begin{bmatrix} 2\hat{\theta}_1 \\ \mathbf{0}_{\Xi_0 \times 1} \end{bmatrix}.$$

IV. STABILITY ANALYSIS

The following theorem proves the boundedness of the tracking error and the weight estimation of the weights.

Theorem 1. For the dynamical system in (1), the proposed controller (4) and the adaptation law (5) ensure the boundedness of the tracking error ξ and the weight estimation $\hat{\theta}$, provided that the control gains k_q and k_z satisfy (8).

Proof. The boundedness is proven from the last layer to the first layer.

Step 1: Boundedness of $\hat{\theta}_1, \eta_1, \xi$

Without loss of generality, assume that all the constraints are violated. Then, according to (5b) and (5c), all Lagrange multipliers are nonzero.

The dynamics of ξ can be represented as

$$\frac{d}{dt} \xi = A_\xi \xi + B_\xi (-\widehat{W}_1^\top \hat{\Phi} + w(t))$$

where $w(t) := W_1^{*\top} \phi^* + \epsilon$ is a lumped residual term, which is bounded as $\|w(t)\| \leq \bar{w} < 0$ since $\|\theta_1^*\|, \|\phi^*\|$ and $\|\epsilon\|$ are bounded. The dynamics of η_1 and $\hat{\theta}_1$ are represented as

$$\begin{aligned} \frac{d}{dt} \eta_1 &= A_\xi \eta_1 - B_\xi (I_{l_2} \otimes \hat{\Phi}^\top), \\ \frac{d}{dt} \hat{\theta}_1 &= -\alpha (\eta_1^\top \Lambda \xi + 2\lambda_{\theta_1} \hat{\theta}_1). \end{aligned}$$

According to [18, Chap. 4 T. 1.9], the boundedness of η_1 can be obtained since A_ξ is stable and the residual term $-B_\xi (I_{l_2} \otimes \hat{\Phi}^\top)$ is bounded.

Define the Lyapunov function $V_1 := \frac{1}{2} \xi^\top P \xi + \frac{1}{2\alpha} \hat{\theta}_1^\top \hat{\theta}_1$, with the Lyapunov equation $A_\xi^\top P + P A_\xi = -Q$, where $A_\xi < 0, P = P^\top > 0$, and $Q > 0$. Using a proposition $\widehat{W}_1^\top \hat{\Phi} = \text{vec}(\widehat{W}_1^\top \hat{\Phi}) = \text{vec}(\hat{\Phi}^\top \widehat{W}_1) = (I_{l_2} \otimes \hat{\Phi}^\top) \text{vec}(\widehat{W}_1) = (I_{l_2} \otimes \hat{\Phi}^\top) \hat{\theta}_1$ [1, Proposition (7.1.9)], the time derivative of V_1 is

$$\begin{aligned} \frac{d}{dt} V_1 &= \frac{1}{2} \xi^\top (A_\xi^\top P + P A_\xi) \xi + \xi^\top P (-B_\xi \widehat{W}_1^\top \hat{\Phi} + B_\xi w(t)) \\ &\quad - \hat{\theta}_1^\top (\eta_1^\top \Lambda \xi + 2\lambda_{\theta_1} \hat{\theta}_1) \\ &= -\frac{1}{2} \xi^\top Q \xi - \xi^\top P B_\xi (I_{l_2} \otimes \hat{\Phi}^\top) \hat{\theta}_1 + \xi^\top \Delta \\ &\quad - \hat{\theta}_1^\top \eta_1^\top \Lambda \xi - 2\lambda_{\theta_1} \hat{\theta}_1^\top \hat{\theta}_1 \\ &\leq -\frac{1}{2} \lambda_{\min}(Q) \|\xi\|^2 + \bar{\Delta} \|\xi\| + \bar{M} \|\xi\| \|\hat{\theta}_1\| \\ &\quad - 2\lambda_{\theta_1} \|\hat{\theta}_1\|^2 \\ &\leq (-\frac{1}{2} \lambda_{\min}(Q) + \frac{\bar{M}}{2}) \|\xi\|^2 + \bar{\Delta} \|\xi\| \\ &\quad + (-2\lambda_{\theta_1} + \frac{\bar{M}}{2}) \|\hat{\theta}_1\|^2, \end{aligned} \quad (7)$$

where $\Delta := P B_\xi w(t)$ and $M := -P B_\xi (I_{l_2} \otimes \hat{\Phi}^\top) - \Lambda \eta_1$ are bounded such that $\|\Delta\| \leq \bar{\Delta} < \infty$ and $\|M\|_F \leq \bar{M} < \infty$, respectively.

By defining $P = I_n$, the eigenvalues of $Q = -A_\xi^\top - A_\xi$ are $2k_q$ and $2k_z$, since A_ξ is a skew-symmetric matrix except for the diagonal entries. According to (7), if k_q and k_z are provided that

$$\min(k_q, k_z) > \frac{\bar{M}}{2}, \quad (8)$$

and if λ_{θ_1} is increased sufficiently large such that $2\lambda_{\theta_1} > \frac{\bar{M}}{2}$, due to the violation of c_{θ_1} , the tracking error is bounded in

$$\Theta_\xi := \left\{ \xi \mid \|\xi\| \leq \frac{2\bar{\Delta}}{\lambda_{\min}(Q) - \bar{M}} \right\},$$

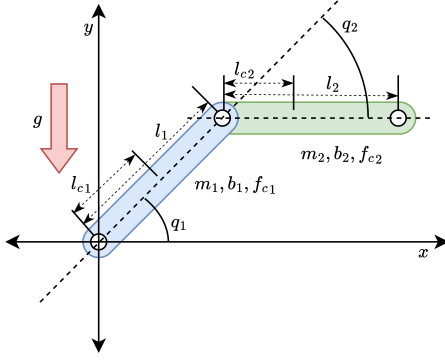


Fig. 1: Two-link manipulator model.

and the weight estimation $\hat{\theta}_1$ is bounded in

$$\Theta_{\theta_1} := \left\{ \hat{\theta}_1 \mid \|\hat{\theta}_1\| \leq \bar{\theta}_1 \right\}.$$

The Lagrange multiplier λ_{θ_1} is also bounded since the λ_{θ_1} update stops once $\hat{\theta}_1$ approaches the compact set Θ_{θ_1} , satisfying the constraint c_{θ_1} .

Step 2: Boundedness of $\hat{\theta}_0, \eta_0$

The dynamics of η_0 and $\hat{\theta}_0$ are represented as

$$\begin{aligned} \frac{d}{dt} \eta_0 &= A_\xi \eta_0 - B_\xi \hat{W}_1^\top \hat{\phi}'(I_{l_1} \otimes q_{NN}^\top) \\ \frac{d}{dt} \hat{\theta}_0 &= -\alpha(\eta_0^\top \Lambda \xi + 2\lambda_{\theta_0} \hat{\theta}_0). \end{aligned}$$

According to [18, Chap. 4 T. 1.9], η_0 is bounded since A_ξ is a stable matrix and $-B_\xi \hat{W}_1^\top \hat{\phi}'(I_{l_1} \otimes q_{NN}^\top)$ is bounded. To obtain the invariance set of $\hat{\theta}_0$, taking the time derivative of the Lyapunov function $V_0 = \frac{1}{2\alpha} \hat{\theta}_0^\top \hat{\theta}_0$ yields:

$$\begin{aligned} \frac{d}{dt} V_0 &= -\hat{\theta}_0^\top (\eta_0 \Lambda \xi + 2\lambda_{\theta_0} \hat{\theta}_0) \\ &\leq \|\hat{\theta}_0\| \|\eta_0 \Lambda \xi\| - 2\lambda_{\theta_0} \hat{\theta}_0^\top \hat{\theta}_0, \\ &\leq -2\lambda_{\theta_0} \|\hat{\theta}_0\|^2 + \|\eta_0 \Lambda \xi\| \|\hat{\theta}_0\|. \end{aligned}$$

Then, the invariance set can be represented as

$$\Theta_{\theta_0} := \left\{ \hat{\theta}_0 \mid \|\hat{\theta}_0\| \leq \frac{\|\eta_0 \Lambda \xi\|}{2\lambda_{\theta_0}} \right\}.$$

If λ_{θ_0} is increased sufficiently large due to the violation of c_{θ_0} , the invariance set Θ_{θ_0} converges to $\{\hat{\theta}_0 \mid \|\hat{\theta}_0\| \leq \bar{\theta}_0\}$ until the constraint c_{θ_0} is satisfied. Therefore, the Lagrange multiplier λ_{θ_0} is also bounded.

V. SIMULATIONS

A. Setup

The two-link manipulator model in [19] is employed for the simulation demonstration as described in Fig. 1. In the system, the parameters $q_p, q_{d_p}, \tau_p, m_p, l_p, l_{c_p}, b_p$ and f_{c_p} denote the joint angle, desired joint angle, torque, mass, length, center of mass, viscous coefficient, and friction coefficient, respectively, for link $p \in [1, 2]$. The values of the system

TABLE I: System model parameters.

Symbol	Description	Link 1	Link 2
m_1, m_2	Mass of link	23.902 (kg)	3.88 (kg)
l_1, l_2	Length of link	0.45 (m)	0.45 (m)
l_{c1}, l_{c2}	COM of link	0.091 (m)	0.048 (m)
θ_1, b_2	Viscous coefficient	2.288 (Nms)	0.172 (Nms)
f_{c1}, f_{c2}	Friction coefficient	7.17 (Nm)	1.734 (Nm)

parameters are given in Table I. The reference signal of $q = [q_1, q_2]^\top$ is defined as follows:

$$q_d = \begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} +\cos(\frac{\pi}{2}t) + 1 \\ -\cos(\frac{\pi}{2}t) - 1 \end{bmatrix}.$$

For the comparative study, three controllers were selected: the neuro-adaptive controller with L_2 -regularization (NAC-L2) and with ϵ -modification (NAC-eMod) and the proposed controller with constrained optimization (NAC-CO). The performances of the selected controllers are compared based on the tracking performances and the dependencies of the parameters λ , ρ , and β_j of NAC-L2, NAC-eMod, and NAC-CO, respectively. The square root of the integrated squared error (ISE) (i.e., $\sqrt{\int_0^T \|\xi\|^2 dt}$, where T denotes a simulation termination time) is utilized to evaluate the tracking performance. The parameter dependencies of the controllers were examined via various values of the parameters. The values ranged from 0.001 to 1 across 10 samples.

The control laws of all three controllers were the same as those defined in (4). The adaptation law of NAC-L2 is derived by adding the squared weight term $\frac{1}{2} \lambda \hat{\theta}^\top \hat{\theta}$ to the objective function such that $J_{L_2} = J + \frac{1}{2} \lambda \hat{\theta}^\top \hat{\theta}$, where $\lambda \in \mathbb{R}_{>0}$ denotes the L_2 coefficient. The adaptation law obtained via the gradient descent method is subsequently adjusted by adding a stabilizing term $-\alpha \lambda \hat{\theta}$ as follows:

$$\frac{d}{dt} \hat{\theta} = \frac{\partial J_{L_2}}{\partial \hat{\theta}} = -\alpha \left(\frac{\partial J}{\partial \hat{\theta}} + \lambda \hat{\theta} \right).$$

Note that this adaptation law derived based on L_2 -regularization method in deep learning is inherently the same as the σ -modification in adaptive control theory, which adds the term $-\alpha \sigma \hat{\theta}$, where $\sigma \in \mathbb{R}_{>0}$. For NAC-eMod, similar to the σ -modification, the stabilizing function $-\alpha \rho \|\tilde{z}\| \hat{\theta}$ is added to the adaptation law as follows:

$$\frac{d}{dt} \hat{\theta} = -\alpha \left(\frac{\partial J}{\partial \hat{\theta}} + \rho \|\tilde{z}\| \hat{\theta} \right)$$

□ where $\rho \in \mathbb{R}_{>0}$ denotes the ϵ -modification coefficient. By $\|\tilde{z}\|$, the stabilizing function proportionally increases as the tracking error \tilde{z} increases. Therefore, the adaptation attempts to reduce the tracking error mainly without the effect of the stabilizing function if the tracking error is sufficiently regulated. The adaptation law of NAC-CO is presented in (5). Owing to the stabilizing functions, the weights of NAC-L2 and NAC-eMod are biased since the stabilizing functions drive the weights toward the origin.

All controllers had the same control parameters except their crucial parameters (i.e., λ , ρ and β_j) as $k_q = 1.1$,

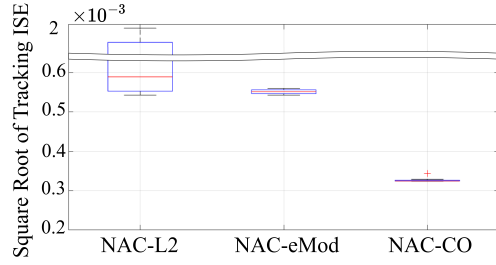


Fig. 2: Box-and-Whisker plot of the square root of the tracking ISEs of NAC-L2, NAC-eMod and NAC-CO across various parameter values.

TABLE II: Quantitative comparison of square root of tracking ISE.

	NAC-L2	NAC-eMod	NAC-CO (proposed)
Maximum	11.1753e-3	0.5603e-3	0.3439e-3
Median	0.5898e-3	0.5519e-3	0.3240e-3
Minimum	0.5434e-3	0.5434e-3	0.3235e-3

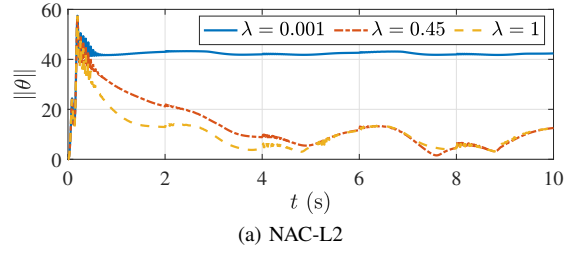
$k_z = 10$, $\widehat{\mathbf{M}} = \mathbf{I}_2$ and $\mathbf{\Lambda} = \text{diag}([5, 1, 15, 15])$. The parameters of the NNs were set to $l_0 = 2$, $l_1 = 16$, $l_2 = 2$, and $\alpha = 10^3$, and the same random seed was applied for weight initialization. [MS: The validity of the selected number of nodes were demonstrated through experiments] The NN input vector was set to the desired trajectory \mathbf{q}_d , with the augmented 1 to incorporate the bias term in the weight matrix such that $\mathbf{q}_{NN} = [\mathbf{q}_d^\top, 1]^\top$. For NAC-CO, the parameters of the weight norm constraints were set as $\bar{\theta}_0 = 10$ and $\bar{\theta}_1 = 20$. The sampling time of the simulation and the simulation termination time were set to $T_s = 100 \mu\text{s}$ and $T = 10 \text{ s}$, respectively.

B. Results

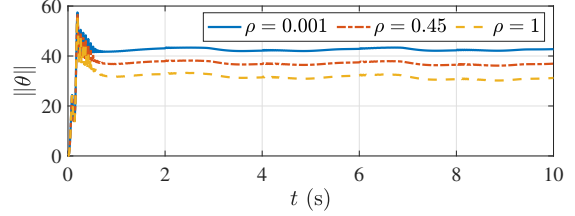
As shown in Fig. 2, the maximum square root of the tracking ISE of NAC-CO is smaller than the minimum square root of the tracking ISEs of NAC-L2 and NAC-eMod for all variations in the parameters. This is because NAC-L2 and NAC-eMod bias the weights to the origin due to the presence of stabilizing functions. A quantitative comparison of the square root of the tracking ISE is provided in Table II.

For the detailed analysis, three values of the parameters (i.e., $\lambda, \rho, \beta_j \in [0.001, 0.45, 1]$) were selected as described in Fig. 3 and Fig. 4. As shown in Fig. 3a, increasing λ reduces the weight norm of NAC-L2 via the stabilizing function $-\alpha\lambda\widehat{\boldsymbol{\theta}}$. Moreover, the high dependency of NAC-L2 on the L_2 -regularization coefficient λ can also be observed. Since the weight norm is decreased, NAC-L2 cannot generate sufficient control inputs, resulting in a larger square root of tracking ISE, as shown in Fig. 4a.

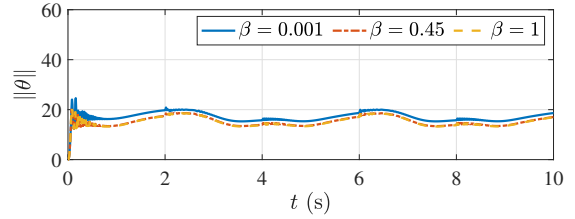
On the other hand, NAC-eMod has a lower dependency on the ϵ -modification coefficient ρ , as shown in Fig. 3b and Fig. 4b. This is because the stabilizing function $-\alpha\rho\|\tilde{\mathbf{z}}\|\widehat{\boldsymbol{\theta}}$ can be decreased once the tracking error $\tilde{\mathbf{z}}$ is sufficiently regulated. However, the bias of the weights to the origin still



(a) NAC-L2

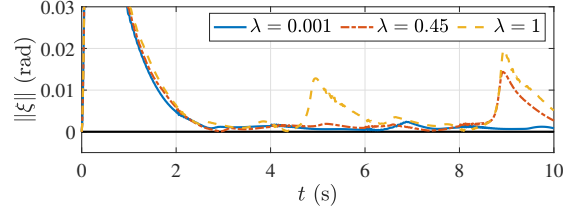


(b) NAC-eMod

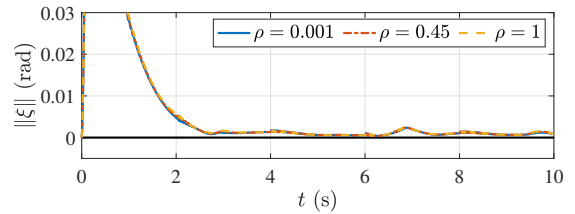


(c) NAC-CO

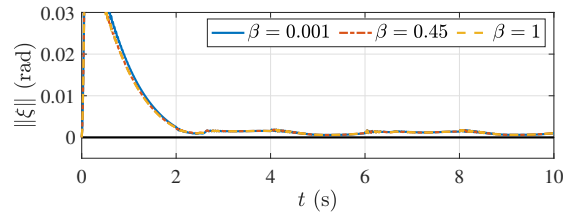
Fig. 3: Weight norms of NAC-L2, NAC-eMod, and NAC-CO.



(a) NAC-L2



(b) NAC-eMod



(c) NAC-CO

Fig. 4: Tracking errors of NAC-L2, NAC-eMod, and NAC-CO.

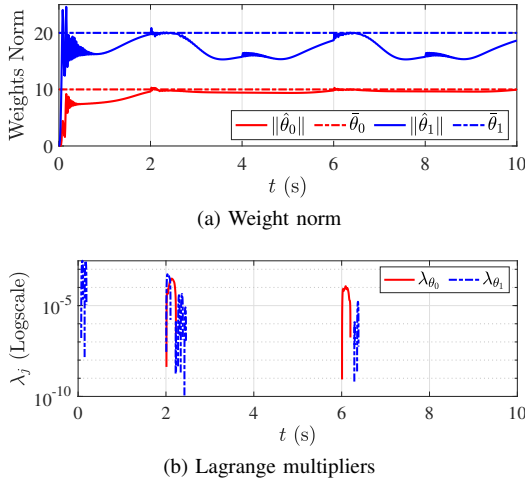


Fig. 5: Weight norms and Lagrange multipliers of NAC-CO ($\beta = 0.001$).

exists, as described in Fig. 3b (i.e., smaller weight norms are observed as ρ increases.). Therefore, similar to NAC-L2, the biased weights produce insufficient control input, resulting in a relatively larger square root of tracking ISE than that of NAC-CO, as described in Table II.

Finally, the weight norm of NAC-CO is smaller than those of NAC-L2 and NAC-eMod, as shown in Fig. 3c, with better tracking performance. Even if a large β_j is provided, NAC-CO can adjust the adaptation direction to satisfy the weight norm constraints faster, according to (5b). Therefore, the lowest dependency on the update rate β_j is observed in NAC-CO, as shown in Fig. 3c and Fig. 4c. Note that β_j of NAC-CO is the update rate for the Lagrange multipliers, whereas λ and ρ are the coefficients of the stabilizing function that generates the biases of the weights. However, considering the implementation using a digital computer, excessively large β_j values should be avoided.

The details of the satisfaction of the weight norm constraints are shown in Fig. 5 for NAC-CO with $\beta_j = 0.001$. As the weight norms of each layer reach the constraint boundary, the corresponding Lagrange multipliers are generated. Using the Lagrange multipliers, the adaptation direction is adjusted toward the constraint satisfactory point. The Lagrange multipliers disappear when the constraints are satisfied, and the weights are adapted to optimize the original objective function without weight bias.

Furthermore, it is important to note that NAC-CO shows enhanced tracking performance with smaller weights than NAC-L2 and NAC-eMod. This implies that the weights in NAC-CO approach the different local optimal solution points from those of NAC-L2 and NAC-eMod. Therefore, if the physical analysis of the system is available to predict the feasible maximum control inputs, NAC-CO can find the local optimal solution without unnecessarily large control inputs by imposing proper weight norm constraints.

VI. CONCLUSION

In this paper, a neuro-adaptive control method is proposed for uncertain Euler-Lagrange systems, ensuring weight boundedness. Adaptation laws are derived by formulating a constrained optimization problem with weight norm constraints. The boundedness of the tracking error and the weight estimation are analyzed via Lyapunov analysis. The simulation results demonstrate that the proposed controller outperforms the existing methods in terms of tracking performance and parameter dependency. As further work, the state and output constraints for safety will be handled, ensuring stability.

REFERENCES

- [1] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton University Press, 2009.
- [2] G. Tao, *Adaptive Control Design and Analysis (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. USA: John Wiley & Sons, Inc., 2003.
- [3] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006.
- [4] J. A. Farrell and M. M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. USA: Wiley-Interscience, 2006.
- [5] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, pp. 303–314, Dec. 1989.
- [6] K. Esfandiari, F. Abdollahi, and H. A. Talebi, "Adaptive control of uncertain nonaffine nonlinear systems with input saturation using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2311–2322, 2015.
- [7] W. Gao and R. Selmic, "Neural network control of a class of nonlinear systems with actuator saturation," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 147–156, 2006.
- [8] S. Ge and C. Wang, "Direct adaptive nn control of a class of nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 214–221, 2002.
- [9] X. Zhou, H. Shen, Z. Wang, H. Ahn, and J. Wang, "Driver-centric lane-keeping assistance system design: A noncertainty-equivalent neuro-adaptive control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 6, pp. 3017–3028, 2023.
- [10] O. S. Patil, D. M. Le, M. L. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Systems Letters*, vol. 6, pp. 1855–1860, 2022.
- [11] Y.-h. Sheu, "Illuminating the black box: Interpreting deep neural network models for psychiatric research," *Frontiers in Psychiatry*, vol. 11, 2020.
- [12] C. Rudin and J. Radin, "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition," *Harvard Data Science Review*, vol. 1, nov 22 2019. <https://hdsr.mitpress.mit.edu/pub/9kuryi8>.
- [13] D. Wu and J. Xu, "On the optimal weighted ℓ_2 regularization in overparameterized linear regression," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), Curran Associates Inc., 2020.
- [14] A. Lewkowycz and G. Gur-Ari, "On the training dynamics of deep networks with ℓ_2 regularization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), Curran Associates Inc., 2020.
- [15] J. Nocedal and S. Wright, *Numerical optimization*. Springer series in operations research and financial engineering, New York, NY: Springer, 2. ed. ed., 2006.
- [16] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," 2017.
- [17] B. Sengupta, K. Friston, and W. Penny, "Efficient gradient computation for dynamical models," *NeuroImage*, vol. 98, pp. 521–527, 2014.
- [18] C. A. Desoer and M. Vidyasagar, *Feedback Systems*. Society for Industrial and Applied Mathematics, 2009.

- [19] E. D. Markus, J. T. Agee, and A. A. Jimoh, "Trajectory control of a two-link robot manipulator in the presence of gravity and friction," in *2013 Africon*, pp. 1–5, 2013.