Imposing a Weight Norm Constraint for Neuro-Adaptive Control

IEEE European Control Conference (ECC) 2025

Myeongseok Ryu[1], Jiyun Kim[2], and Kyunghwan Choi[1]

[1]Department of Mechanical and Robotics Engineering
Gwangju Institute of Science and Technology

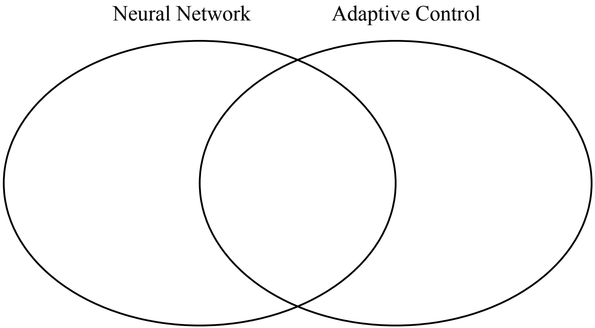
[2]AI Graduate School
Gwangju Institute of Science and Technology

2025-06-25

Outline

Neuro-Adaptive Control

- **Neuro-adaptive control** (NAC) is a control strategy that combines **neural networks (NNs)** with **adaptive control** [1].
- Features of both **NNs** and **adaptive control** can be found in NAC.



Advantages of Neuro-Adaptive Control

- **Adaptability**: NAC adapts to changing environments and system dynamics.
- **Stability Guarantee**: The closed-loop stability is ensured using *Lyapunov stability theory*.
- **Online Learning Capability**: NAC adapts in *real-time* to new data with stability guarantees.
- **Robustness**: NAC handles *uncertainties and disturbances* effectively with adaptive control techniques.

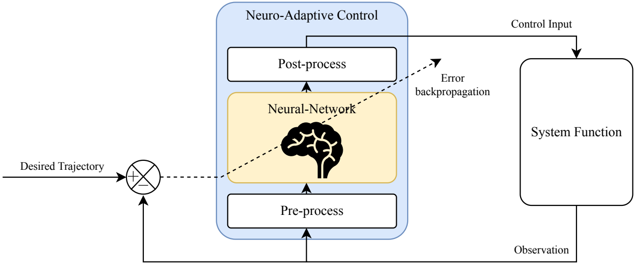


Figure: General framework of neuro-adaptive control (NAC).

Challenges

- **Optimality**: In general, the adaptation laws are driven with respect to the tracking error, which may not guarantee optimal performance.
- **Unpredictable Amplitude of Control input**:
  - The maximum amplitude of the NN weights is not predictable.
  - This can result in unpredictable amplitude of the control input, which may lead to control saturation.
- **Parameter Dependency**:
  - The performance of the NAC is highly dependent on the choice of parameters, such as learning rate and NN architecture.

In general, the NN outputs are bounded by limiting the maximum amplitude of the NN weights.

1. **Projection Operator**
    - Cats
    - Dogs
    - Birds

2. **ϵ-modification, and σ-modification**
    - Cats
    - Dogs
    - Birds

The input saturation is generally handled using an auxiliary system.

1. **Auxiliary Systems**
    - Cats
    - Dogs
    - Birds

2. **Second?**
    - Cats
    - Dogs
    - Birds

Objective 1: Optimality

- Formulate a constrained optimization problem to minimize the tracking error.
- Guarantee the stability of the system and the NN weights.

Objective 2: Stability

- Derive an adaptation law that guarantees the stability of the system.
- Ensure that the NN weights remain bounded during operation.

Objective 3: Boundedness of

- Ensure the controller is robust to uncertainties and disturbances.
- Validate the proposed method through numerical simulations.

Target 2-link Robotic Manipulator System:

- Control input saturation function $\mathrm{sat}(\cdot)$.
- Desired trajectory $\boldsymbol{q}_d$ is given.

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

Control Input:

- NN's output $\boldsymbol{\Phi}$ is used as the control input.
- Consists of the estimated NN weights $\widehat{\boldsymbol{\theta}}$.

$$\boldsymbol{\tau} := \boldsymbol{\Phi}(\boldsymbol{q}_n; \widehat{\boldsymbol{\theta}}) \tag{2}$$

Deep Neural Network (DNN):

- k layers with $\widehat{\boldsymbol{\theta}}_i := \mathrm{vec}(\widehat{\boldsymbol{W}}_i)$.
- Activation function: $\phi(\cdot) := tanh(\cdot)$.

$$\boldsymbol{\Phi}(\boldsymbol{q}_n; \widehat{\boldsymbol{\theta}}) := \begin{cases} \widehat{\boldsymbol{W}}_i^\top \phi_i(\widehat{\boldsymbol{\Phi}}_{i-1}), & i \in \{1, \dots, k\}, \\ \widehat{\boldsymbol{W}}_0^\top \boldsymbol{q}_n, & i = 0, \end{cases} \tag{3}$$

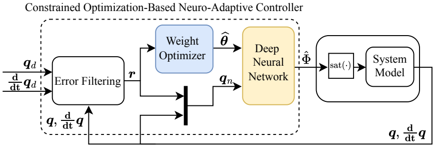


Figure: Architecture of the proposed method.

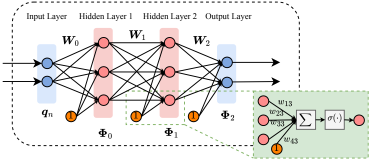


Figure: Architecture of the DNN.

Notations: $\boldsymbol{q} \in \mathbb{R}^n$: Joint position, $\boldsymbol{M}$: Inertia matrix, $\boldsymbol{C}$: Coriolis matrix, $\boldsymbol{G}$: Gravity vector, $\boldsymbol{\tau}$: Control input, $\widehat{\boldsymbol{\theta}}$: Estimated NN weights. $\widehat{\boldsymbol{\theta}}$: Estimated NN weights, $\boldsymbol{e}$: Tracking error, $\boldsymbol{\tau}_d$: Disturbance, $\mathrm{sat}(\cdot)$: Saturation function.

Optimization Problem Statement:

- **Find** NN weights $\widehat{\boldsymbol{\theta}}$,
- That **minimize** objective function $J(\cdot)$,

$$J(\boldsymbol{r}) := \frac{1}{2}\boldsymbol{r}^\top \boldsymbol{r}. \tag{4}$$

  - where $\boldsymbol{r} := \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{e} + \Lambda\boldsymbol{e}$ is the filtered tracking error,
- while satisfying the following **constraints**:
  - Boundedness of the NN weights $\widehat{\boldsymbol{\theta}}$.
  - Saturation of the control input $\boldsymbol{\tau}$.

Considered Constraints

- **Weight Boundedness**:

$$c_{\theta_i} := \|\widehat{\boldsymbol{\theta}}_i\|^2 - \bar{\theta}_i^{\,2} \leq 0, \forall i \in \{0, \dots, k\} \tag{5}$$

- **Control Input Saturation**:
  - Input bound constraint:

$$c_{\overline{\tau}_i} := \tau_i - \overline{\tau_i} \leq 0, \; c_{\underline{\tau}_i} := \underline{\tau_i} - \tau_i \leq 0 \tag{6}$$

  - Input norm constraint:

$$c_{\boldsymbol{\tau}} := \|\boldsymbol{\tau}\|^2 - \overline{\tau}^2 \leq 0 \tag{7}$$

Notations: $\Lambda \in \mathbb{R}^{n\times n}_{>0}$: filtering matrix

Optimization Problem

$$\begin{aligned} &\min_{\widehat{\boldsymbol{\theta}}} J(\boldsymbol{r}; \widehat{\boldsymbol{\theta}}) \\ &\text{s.t. } c_j(\widehat{\boldsymbol{\theta}}) \leq 0, \forall j \in \mathcal{J} \end{aligned} \tag{8}$$

MinMax Problem

$$\begin{aligned} &\min_{\widehat{\boldsymbol{\theta}}} \max_{[\lambda_j]_{j \in \mathcal{J}}} L(\cdot) \\ &\text{where } L(\boldsymbol{r}, \widehat{\boldsymbol{\theta}}, [\lambda_j]_{j \in \mathcal{J}}) := J(\boldsymbol{r}; \widehat{\boldsymbol{\theta}}) + \textstyle\sum_{j \in \mathcal{J}} \lambda_j c_j(\widehat{\boldsymbol{\theta}}) \end{aligned} \tag{9}$$

For the min/max problem, the steepest descent/ascent method is used to derive the adaptation law.

Steepest Descent Method for $\widehat{\boldsymbol{\theta}}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\boldsymbol{\theta}} = -\alpha \frac{\partial L}{\partial \widehat{\boldsymbol{\theta}}} \tag{10}$$

Steepest Ascent Method for λ_j:

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_j = \tag{11}$$

Adaptation Law

$$\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\boldsymbol{\theta}} = -\alpha \frac{\partial L}{\partial \widehat{\boldsymbol{\theta}}} = -\alpha \left(\frac{\partial J}{\partial \widehat{\boldsymbol{\theta}}} + \sum_{j \in \mathcal{I}} \lambda_j \frac{\partial c_j}{\partial \widehat{\boldsymbol{\theta}}} \right), \tag{12a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_j = \beta_j \frac{\partial L}{\partial \lambda_j} = \beta_j c_j, \qquad \forall j \in \mathcal{I}, \tag{12b}$$

$$\lambda_j = \max(\lambda_j, 0), \tag{12c}$$

Theorem 1

For the dynamical system described in (1), the neuro-adaptive controller in (2) with the weight adaptation laws in (12) ensure the boundedness of the filtered error $\boldsymbol{r}$ and the weight estimate $\widehat{\boldsymbol{\theta}}$, under the control input constraints satisfying Assumption 1 and 2. This holds under the weight norm constraint (5).

Assumption 1

The constraint functions $c_j(\widehat{\boldsymbol{\theta}}), \forall j \in \mathcal{I}$, are convex in the τ-space and satisfy $c_j(\widehat{\boldsymbol{\theta}}) \leq 0$ and $c_j(\boldsymbol{\theta}^*) \leq 0$.

Assumption 2

The selected constraints satisfy the Linear Independence Constraint Qualification (LICQ) [2, Chap. 12 Def. 12.1].

Proof of Theorem 1 is omitted due to space limitations. The detailed proof can be found in [?].

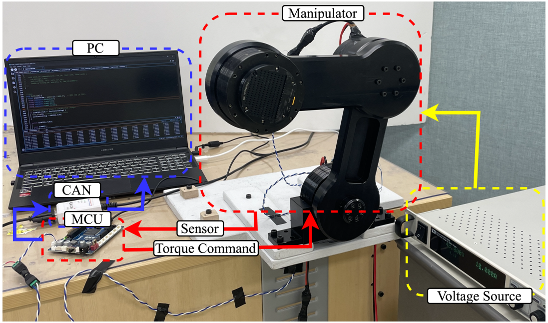


Figure: 2-Link Robotic Manipulator

Notations: $\boldsymbol{q}$: Joint position, $\dot{\boldsymbol{q}}$: Joint velocity, $\boldsymbol{\tau}$: Control input, $\boldsymbol{M}$: Inertia matrix, $\boldsymbol{C}$: Coriolis matrix, $\boldsymbol{G}$: Gravity vector.

Representative Simulation Results

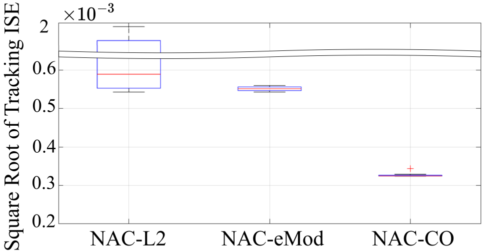


Figure: Parameter dependencies of the proposed method.

Table: Largest cities in the world (source: Wikipedia)

City	Population
Mexico City	20,116,842
Shanghai	19,210,000
Peking	15,796,450
Istanbul	14,160,467

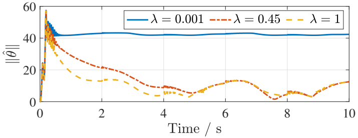


Figure: Weight norms of NAC-L2

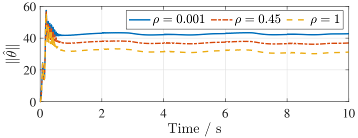


Figure: Weight norms of NAC-eMod

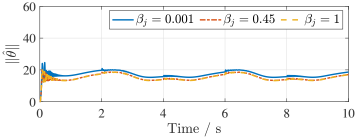


Figure: Weight norms of NAC-CO

The weight norms of the proposed method (NAC-CO) are bounded, while ...

Parameter Dependencies

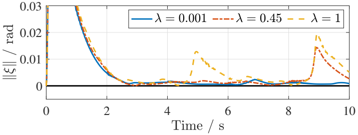


Figure: Tracking error of NAC-L2

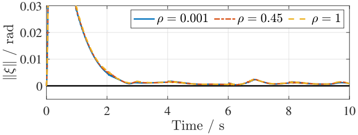


Figure: Tracking error of NAC-eMod

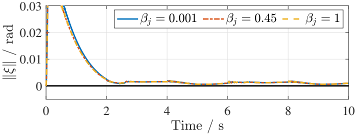


Figure: Tracking error of NAC-CO

The Tracking error ...

Summary of Contributions

- Proposed a novel constrained optimization-based neuro-adaptive control (CONAC) method.
- Adaptation laws are derived using constrained optimization method.
- The proposed method guarantees the stability of the system and the boundedness of the NN weights.
- Feasibility of the proposed method is validated through numerical simulations and real-time experiments.

Future Work

- Extend the proposed method to state constraints.
- Enhance the robustness and flexibility of the proposed method for various systems.

Thank you for your attention!

[1] 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.

[2] J. Nocedal and S. Wright, *Numerical optimization.*
Springer series in operations research and financial engineering, New York, NY: Springer, 2. ed. ed., 2006.