CVL-based Constrained Optimization Neuro-Adaptive Control (CVL-CONAC) for Uncertain Control-Affine Systems

Naol Samuel Erega, Myeongseok Ryu, and Kyunghwan Choi

***Abstract*—This study proposes a Convolutional Layer (CVL)-based Constrained Optimization Neuro-Adaptive Controller (CVL-CONAC) for uncertain control-affine nonlinear systems. The controller approximates the ideal stabilizing law using a hybrid CVL-FCL architecture that processes historical state data, ensuring a robust representation of system dynamics. The adaptation law is derived by formulating the control problem as a constrained minimization problem of the tracking error, subject to weight norm constraints and control input norm constraints. Stability analysis, based on the Lyapunov approach, rigorously proves that the tracking error and all network weight vectors remain Uniformly Ultimately Bounded(UUB), thereby ensuring stable and constrained online adaptation.**

***Index Terms*—Neuro-adaptive control, Constrained optimization, Convolutional Layers (CVL), Lyapunov approach, Asymptotic convergence.**

Notation

In this study, the following notation is used:

- $\mathbb{R}^n$ denotes the n-dimensional Euclidean space.
- $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices.
- A vector and a matrix are denoted by $\boldsymbol{x} = [x_i] \in \mathbb{R}^n$ and $\boldsymbol{A} := [a_{ij}] \in \mathbb{R}^{n\times m}$, respectively.
- $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, and $\mathbf{0}_{n\times m}$ denotes the $n \times m$ zero matrix.
- $\mathrm{vec}(\boldsymbol{A})$ denotes the vectorization of the matrix $\boldsymbol{A}$.
- A matrix $\boldsymbol{P} \succ 0$ ($\boldsymbol{P} \succeq 0$) is positive definite (positive semidefinite).
- $\lambda_{\min}(\boldsymbol{A})$ denotes the minimum eigenvalue of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.
- $\otimes$ denotes the Kronecker product [1].

I. Introduction

A. Motivation and Background

Neuro-Adaptive Control (NAC) has emerged as a crucial tool for controlling systems with unknown dynamics, leveraging the universal approximation capabilities of Neural Networks (NNs) [??]. While traditional Lyapunov-based NACs primarily focus on ensuring the boundedness of the tracking error and weight estimation error [??], two key challenges persist in practical implementation: managing the potential divergence of NN weights and satisfying physical input constraints imposed by actuators[??].

The Constrained Optimization-Based Neuro-Adaptive Control (CONAC) framework addresses these limitations by embedding both weight and input constraints directly into the adaptation process via a unified constrained optimization problem[??]. This framework solves the dual optimization problem by updating both the weights and the Lagrange multipliers, guaranteeing satisfaction of the Karush-Kuhn-Tucker (KKT) conditions at steady state[??].

This paper extends the CONAC methodology to incorporate the feature-capturing strength of a Convolutional Layer (CVL) architecture. CVLs are particularly well-suited for processing time-stacked historical sensor data, represented as a 2D input matrix Figure 1, allowing the controller to capture spatio-temporal features essential for approximating complex, time-dependent dynamics[??].

B. Main Contributions

The main contributions of this study are threefold:

- Adapting the CONAC formulation specifically for a CVL-FCL architecture;
- Applying the method to a generic control-affine system;
- Rigorously proving the UUB of the tracking error and all network weights under the derived adaptation law.

II. Problem Formulation and Control Law

A. System Dynamics

We consider an uncertain generic control-affine nonlinear system, where the control gain matrix $g(x)$ is assumed to be square and positive definite ($n = m$ and $g(x) \succ 0$):

$$\dot{x} = f(x) + g(x)\,\mathbf{sat}(u) \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^n$ is the control input.

The system is reformulated using a Hurwitz designer matrix $A_c \in \mathbb{R}^{n\times n}$:

$$\dot{x} = A_c x + f_c(x) + g(x)\,\mathbf{sat}(u) \tag{2}$$

where $f_c(x) = f(x) - A_c x$.

N. S. Erega is with the Dept. of Aerospace Engineering, KAIST, Daejeon, Republic of Korea (email: samuelnaol7@kaist.ac.kr).

M. Ryu and K. Choi are with the Cho Chun Shik Graduate School of Mobility, KAIST, Daejeon, Republic of Korea (email: {dding_98, kh.choi}@kaist.ac.kr).

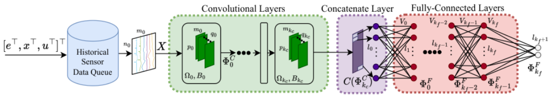

Fig. 1: Proposed CVL-CONAC architecture.

B. Ideal Control Law and Error Dynamics

The objective in defining the Ideal Control Law u^* is to find a control input that forces the tracking error $\mathbf{e} = x - x_{ref}$ dynamics into the desired stable linear form $\dot{e} = A_c e$. This condition is satisfied if:

$$u^* = -g^{-1}(x)\big(A_c x_{ref} + f_c(x) - \dot{x}_{ref}\big)$$

However, since $f_c(x)$ and $g(x)$ are unknown, the ideal controller (i.e., u^*) can't be realized. Hence, u^* is approximated by the CVL-CONAC output $\hat{\Phi}(X, \hat{\theta})$. We assume the existence of an ideal network output Φ^* such that $u^* = -\Phi^* - \epsilon$, where ϵ is the bounded approximation error.

The CVL-CONAC control input is defined as an end-to-end policy:

$$u = -\hat{\Phi}(X, \hat{\theta}) \tag{3}$$

Substituting this into the system dynamics, the tracking error dynamics become:

$$\dot{e} = A_c e + g(x)\big(\Phi^* + \epsilon + \mathbf{sat}(-\hat{\Phi})\big) \tag{4}$$

This formulation is used for stability analysis.

III. System Details

A. CVL-CONAC Architecture

The network $\hat{\Phi}$ is designed as a hybrid architecture suitable for capturing spatio-temporal dynamics from historical data[??].

1) Convolutional Layers (CLs): The CVLs are represented recursively as:

$$\Phi_{j_c}^C = \begin{cases} O(\phi_{j_c}^C(\Phi_{j_c-1}^C), \Omega_{j_c}, B_{j_c}), & j_c \in [1, \dots, k_c] \\ O(\phi_0^C(X), \Omega_0, B_0), & j_c = 0 \end{cases} \tag{5}$$

where $X \in \mathbb{R}^{n_0 \times m_0}$ denotes the network input matrix, $O : \mathbb{R}^{n_{j_c} \times m_{j_c}} \times \mathbb{R}^{p_{j_c} \times m_{j_c} \times q_{j_c}} \times \mathbb{R}^{q_{j_c}} \to \mathbb{R}^{n_{j_c+1} \times m_{j_c+1}}$ denotes the CNN operator (see Appendix A), and $\phi_{j_c}^C : \mathbb{R}^{n_{j_c} \times m_{j_c}} \to \mathbb{R}^{n_{j_c} \times m_{j_c}}$ denotes the matrix activation function (i.e., $\phi^C(\Phi_{j_c-1}^C)_{(i,j)} = \sigma_{j_c}(\Phi_{j_c-1}^C)_{(i,j)})$ for some activation function $\sigma_{j_c} : \mathbb{R} \to \mathbb{R}$. The first activation function ϕ_0^C should be a bounded nonlinear function to ensure that the input to the first CNN operator is bounded. In this study, $\alpha_1 \tanh(\cdot)$ is selected with $\alpha_1 \in \mathbb{R}_{>0}$. The filter set Ω_{j_c} contains q_{j_c} filters $W_{j_c}^{(i)} \in \mathbb{R}^{p_{j_c} \times m_{j_c}}, \forall i \in [1, \dots, q_{j_c}]$, and is represented as $\Omega_{j_c} = \{W_{j_c}^{(1)}, \dots, W_{j_c}^{(q_{j_c})}\}$, where superscript i denotes the filter index. The bias vector B_{j_c} consists of q_{j_c} biases.

The weights for the j_c-th CVL layer are collectively represented as $\hat{\theta}_{Cj_c} \triangleq [\text{vec}(\Omega_{j_c})^T, B_{j_c}^T]^T$.

2) Fully-Connected Layers (FCLs): The output matrix of the CVLs is input to the concatenate layer $C(\Phi_{k_c}^C) = [\text{vec}(\Phi_{k_c}^{C\top})^\top, 1]^\top$ before the input layer of FCLs for compatibility between the CVLs and FCLs. The FCLs are represented recursively as:

$$\Phi_{j_f}^F = \begin{cases} V_{j_f}^\top \phi_{j_f}^F(\Phi_{j_f-1}^F), & j_f \in [1, \dots, k_f] \\ V_0^\top C(\Phi_{k_c}^C), & j_f = 0 \end{cases} \tag{6}$$

where $V_{j_f} \in \mathbb{R}^{l_{j_f}+1 \times l_{j_f+1}}$ denotes the weightmatrix, $\phi_{j_f}^F : \mathbb{R}^{l_{j_f}} \to \mathbb{R}^{l_{j_f}+1}$ denotes the vector activation function defined by $\phi_{j_f}^F(\Phi_{j_f-1}^F) \triangleq [\sigma_{j_f}(\Phi_{j_f-1}^F)^\top, 1]^\top$ for $j_f \in [1, \dots, k_f - 1]$, for some nonlinear activation functions $\sigma_{j_f} : \mathbb{R} \to \mathbb{R}$ such as $\tanh(\cdot)$. Note that $C(\Phi_{k_c}^C)$ and $\Phi_{j_f}^F$ are augmented by 1 to consider the bias of the preceding input layer as a weight in the weight matrix. The output of the FCLs is the final output of the CNN architecture and is represented as $\hat{\Phi} \equiv \Phi_{k_f}^F$.

The FCL weights for the j_f-th layer are $\hat{\theta}_{Fj_f} \triangleq \text{vec}(V_{j_f})$.

IV. Adaptation Laws

A. Optimization Problem

The adaptation law is derived by solving the following constrained minimization problem. We first define the complete set of trainable weights to be constrained:

$$\mathcal{I} = \{C0, C1, \dots, Ck_c\} \cup \{F0, F1, \dots, Fk_f\} \tag{7}$$

The constrained minimization problem is:

$$\begin{aligned} \min_{\hat{\theta}} \quad & \mathcal{J}(e, \hat{\theta}) = \frac{1}{2} e^\top e \\ \text{s.t.} \quad & c_i(\hat{\theta}) = \frac{1}{2}\left(\|\hat{\theta}_i\|^2 - \bar{\theta}_i^2\right) \le 0, \quad \forall i \in \mathcal{I} \\ & c_\Phi(\hat{\theta}) = \frac{1}{2}\left(\|\hat{\Phi}\|^2 - \bar{u}^2\right) \le 0 \end{aligned} \tag{8}$$

where c_i are the weight norm constraints (ensuring boundedness) and c_Φ is the output norm constraint (ensuring input saturation limits $\bar{u}$ are respected). The Lagrangian function is made generic over the index set $\mathcal{I}$:

$$\mathcal{L}(\hat{\theta}, \boldsymbol{\lambda}) = \frac{1}{2} e^\top e + \sum_{i \in \mathcal{I}} \lambda_i c_i(\hat{\theta}) + \lambda_\Phi c_\Phi(\hat{\theta}) \tag{9}$$

where $\lambda_i, \lambda_\Phi \ge 0$ are the Lagrange multipliers.

B. Primal Update

The weight update for each block $\hat{\theta}_i$ is proportional to the negative gradient of the Lagrangian: $\dot{\hat{\theta}}_i = -\alpha \nabla_{\hat{\theta}_i} \mathcal{L}$.

1) Objective Gradient ($\nabla_{\hat{\theta}_i} \mathcal{J}$): The objective gradient is approximated using the chain rule and the sensitivity approximation $\frac{\partial x}{\partial u} \approx \mathbf{I_n}$:

$$\nabla_{\hat{\theta}_i} \mathcal{J} = \left(\frac{\partial e}{\partial \hat{\theta}_i}\right)^T e \approx \left(\frac{\partial x}{\partial u}\frac{\partial u}{\partial \hat{\theta}_i}\right)^T e = -\left(\frac{\partial \hat{\Phi}}{\partial \hat{\theta}_i}\right)^T e$$

2) Constraint Gradients ($\nabla_{\hat{\theta}_i} c_j$): The gradients of the constraints are:

- Weight Constraint: $\nabla_{\hat{\theta}_i} c_i = \hat{\theta}_i$
- Output Constraint : $\nabla_{\hat{\theta}_i} c_\Phi = \left(\frac{\partial \hat{\Phi}}{\partial \hat{\theta}_i}\right)^T \hat{\Phi}$

3) Final Primal Update Rule: The resulting weight update rule for block $\hat{\theta}_i$ is:

$$\dot{\hat{\theta}}_i = \alpha \left[\left(\frac{\partial \hat{\Phi}}{\partial \hat{\theta}_i} \right)^T e - \lambda_i \hat{\theta}_i - \lambda_\Phi \left(\frac{\partial \hat{\Phi}}{\partial \hat{\theta}_i} \right)^T \hat{\Phi} \right] \tag{10}$$

where $\alpha \in \mathbb{R}_{>0}$ is the learning rate.

C. Dual Update

The Lagrange multipliers are updated using projected gradient ascent:

$$\dot{\lambda}_j = \beta_j c_j(\hat{\theta}), \quad \forall j \in \{i, \Phi\} \tag{11}$$

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

where $\beta_j \in \mathbb{R}_{>0}$ is the update rate for the corresponding multiplier[??].

D. Network Jacobians

The adaptation law requires the block Jacobians $\hat{\Phi}'_i \triangleq \frac{\partial \hat{\Phi}}{\partial \hat{\theta}_i}$, where $\hat{\Phi} \equiv \Phi^F_{k_f}$.

1) FCL Jacobians ($\hat{\theta}_{Fj_f}$, $j_f \in [0, \dots, k_f]$): The Jacobians of $\hat{\Phi}$ with respect to the weights of FCLs are derived recursively.

For the input layer weights ($j_f = 0$):

$$\frac{\partial \hat{\Phi}}{\partial \text{vec}(V_0)} = \Big(\prod_{l=1}^{k_f} V_l^\top \phi_l^{F'} \Big) (I_{l_0} \otimes C(\Phi^C_{k_c})) \tag{13}$$

For the inner and output layer weights ($j_f \in [1, \dots, k_f]$):

$$\frac{\partial \hat{\Phi}}{\partial \text{vec}(V_{j_f})} = \Big(\prod_{l=j_f+1}^{k_f} V_l^\top \phi_l^{F'} \Big) (I_{l_{j_f}} \otimes \Phi^{F^\top}_{j_f - 1}) \tag{14}$$

where $\phi^{F'}_{j_f} \triangleq \frac{\partial \phi^F_{j_f}(x)}{\partial x}$ is the Jacobian of the activation function with respect to some vector x.

2) CVL Jacobians ($\hat{\theta}_{Cj_c}$, $j_c \in [0, \dots, k_c]$): The Jacobians of $\hat{\Phi}$ with respect to the weights of CVLs (filters Ω_{j_c} and biases B_{j_c}) are derived using the backpropagation method, linking the output sensitivity $\frac{\partial \hat{\Phi}}{\partial \Phi^C_{j_c}}$ backward through the convolutional layers.

Let Φ_i denote the i-th output of $\hat{\Phi}$. The Jacobians are:

$$\frac{\partial \Phi_i}{\partial W^{l_k}_{j_c}} = \sum_{l_i=1}^{n_{j_c+1}} \Big(\frac{\partial \Phi_i}{\partial \Phi^C_{j_c}(l_i, l_k)} \, \text{row}(l_b : l_e)(\phi^C_{j_c}) \Big), \tag{15}$$

$$\frac{\partial \Phi_i}{\partial B_{j_c}(l_k)} = \sum_{l_i=1}^{n_{j_c+1}} \Big(\frac{\partial \Phi_i}{\partial \Phi^C_{j_c}(l_i, l_k)} \cdot 1 \Big), \tag{16}$$

with $l_b \triangleq l_i$, $l_e \triangleq l_i + p_{j_c} - 1$, $j_c \in [0, \dots, k_c]$

for all $l_k \in [1, \dots, q_{j_c}]$, where $\Phi^C_{j_c} \triangleq \phi^C_{j_c}(\Phi^C_{j_c - 1})$ for $j_c \in [1, \dots, k_c]$, $\Phi^C_0 \triangleq \phi^C_0(X)$, and $\partial \Phi_i / \partial \Phi^C_{j_c}$ denotes the backpropagated gradient of Φ_i with respect to $\Phi^C_{j_c}$ (obtained using the back-propagation method, detailed in Appendix B).

V. Stability Analysis

The stability analysis aims to prove that the filtered tracking error e and all estimated weight vectors $\hat{\theta}$ remain Uniformly Ultimately Bounded (UUB), provided the weight norm constraints are imposed.

A. Boundedness of e

Let the Lyapunov function for the error be $V_e = \frac{1}{2} e^T P e$, with P satisfying $A_c^T P + P A_c = -Q$ ($Q \succ 0$).

The time derivative $\dot{V}_e$ is derived from the error dynamics (4):

$$\dot{V}_e = e^T P \left[A_c e + g(x) \big(\Phi^* + \epsilon + \textbf{sat}(-\Phi) \big) \right]$$

Using the properties of P and A_c, the derivative is bounded by:

$$\dot{V}_e \leq -\frac{1}{2} \lambda_{\min}(Q) \|e\|^2 + \|e\| \|P g(x)\| \left(\|\Phi^*\| + \|\epsilon\| + \|\textbf{sat}(-\Phi)\| \right)$$

Assuming the perturbations and control bounds are finite ($\Psi = \sup\{\|\Phi^*\|, \|\epsilon\|, \|\textbf{sat}(-\Phi)\|\}$) and $\|P g(x)\| \leq \gamma_e$, the ultimate bounding set Ω_e for the tracking error e is:

$$e \in \Omega_e = \left\{ e \mid \|e\| \leq \frac{2 \gamma_e \Psi}{\lambda_{\min}(Q)} \right\}$$

Thus, the tracking error e is UUB.

B. Boundedness of Output Layer Weights ($\hat{\theta}_{Fj_{kf}}$)

1) Case 1: Control Input Saturated ($\lambda_\Phi > 0$): In this case, the input constraint is active, and $\lambda_\Phi > 0$. We analyze the time derivative of the Lyapunov candidate for the estimated weights $V_{\hat{\theta}_{Fj_{kf}}} = \frac{1}{2\alpha} \hat{\theta}^T_{Fj_{kf}} \hat{\theta}_{Fj_{kf}}$.

The derivative $\dot{V}_{\hat{\theta}_{Fj_{kf}}}$ is obtained by substituting the primal update law for $\dot{\hat{\theta}}_{Fj_{kf}}$:

$$\dot{V}_{\hat{\theta}_{Fj_{kf}}} = \hat{\theta}^T_{Fj_{kf}} (\Phi'_{Fj_{kf}})^T e - \lambda_{Fj_{kf}} \|\hat{\theta}_{Fj_{kf}}\|^2 - \lambda_\Phi \hat{\theta}^T_{Fj_{kf}} (\Phi'_{Fj_{kf}})^T \Phi$$

Since the constraint term is non-positive ($-\lambda_\Phi \hat{\theta}^T_{Fj_{kf}} (\Phi'_{Fj_{kf}})^T \Phi \leq 0$), we drop it for worst-case analysis. We use the fact that the Jacobian norm is bounded: $\left\| (\Phi'_{Fj_{kf}})^T \right\| \leq \gamma_{Fj_{kf}}$. Applying Cauchy-Schwarz:

$$\dot{V}_{\hat{\theta}_{Fj_{kf}}} \leq \gamma_{Fj_{kf}} \|e\| \|\hat{\theta}_{Fj_{kf}}\| - \lambda_{Fj_{kf}} \|\hat{\theta}_{Fj_{kf}}\|^2$$

Since $\lambda_{Fj_{kf}} > 0$, $\dot{V}_{\hat{\theta}_{Fj_{kf}}}$ is negative outside the compact set where:

$$\hat{\theta}_{Fj_{kf}} \in \Omega_{\hat{\theta}_{Fj_{kf}}} \triangleq \left\{ \hat{\theta}_{Fj_{kf}} \mid \|\hat{\theta}_{Fj_{kf}}\| \leq \frac{\gamma_{Fj_{kf}}}{\lambda_{Fj_{kf}}} \|e\| \right\}$$

Since $\|e\|$ is UUB, $\hat{\theta}_{Fj_{kf}}$ is also UUB.

2) Case 2: Control Input Not Saturated ($\lambda_\Phi = 0$): With $\lambda_\Phi = 0$ we set $V_3 = V_e + V_{\hat{\theta}_{Fj_{kf}}}$ and treat the two parts separately.

a) Error Lyapunov derivative.: Using the error dynamics,

$$\dot{V}_e = -\tfrac{1}{2}e^\top Qe + e^\top Pg(x)\big(\Phi^* + \epsilon - \hat{\Phi}\big). \tag{17}$$

Apply the first-order expansion:

$$\Phi^* - \hat{\Phi} = \hat{\Phi}'_{F_{j_{kf}}}\big(\theta^*_{F_{j_{kf}}} - \hat{\theta}_{F_{j_{kf}}}\big) + \eta,$$

where η collects higher-order terms, to obtain

$$\begin{aligned}\dot{V}_e = -\tfrac{1}{2}e^\top Qe + e^\top Pg(x)\hat{\Phi}'_{F_{j_{kf}}}\theta^*_{F_{j_{kf}}} - e^\top Pg(x)\hat{\Phi}'_{F_{j_{kf}}}\hat{\theta}_{F_{j_{kf}}}\\ + e^\top Pg(x)(\epsilon + \eta).\end{aligned} \tag{18}$$

b) Parameter Lyapunov derivative.: For the parameter part (adaptive law with $\lambda_{F_{j_{kf}}}$),

$$\dot{V}_{\hat{\theta}_{F_{j_{kf}}}} = \hat{\theta}^\top_{F_{j_{kf}}}(\hat{\Phi}'_{F_{j_{kf}}})^\top e - \lambda_{F_{j_{kf}}}\|\hat{\theta}_{F_{j_{kf}}}\|^2. \tag{19}$$

c) Combine and group coupling terms.: Summing $\dot{V}_e + \dot{V}_{\hat{\theta}_{F_{j_{kf}}}}$ groups the terms with $\hat{\theta}$:

$$\begin{aligned}\dot{V}_3 = -\tfrac{1}{2}e^\top Qe + e^\top Pg(x)\hat{\Phi}'_{F_{j_{kf}}}\theta^*_{F_{j_{kf}}} + e^\top Pg(x)(\epsilon + \eta)\\ + e^\top\big(\hat{\Phi}'_{F_{j_{kf}}} - Pg(x)\hat{\Phi}'_{F_{j_{kf}}}\big)\hat{\theta}_{F_{j_{kf}}} - \lambda_{F_{j_{kf}}}\|\hat{\theta}_{F_{j_{kf}}}\|^2.\end{aligned} \tag{20}$$

d) Assumptions and operator bounds.: We use

$$e^\top Qe \geq \lambda_{\min}(Q)\|e\|^2, \quad \|Pg(x)\| \leq \gamma_e, \quad \|\hat{\Phi}'_{F_{j_{kf}}}\| \leq \gamma_{F_{j_{kf}}},$$

$$\|\epsilon + \eta\| \leq \rho, \quad \|\theta^*_{F_{j_{kf}}}\| \leq \overline{\theta}_{F_{j_{kf}}}.$$

And by defining:

$$M_{F_{j_{kf}}} \triangleq \hat{\Phi}'_{F_{j_{kf}}} - Pg(x)\hat{\Phi}'_{F_{j_{kf}}}, \qquad \|M_{F_{j_{kf}}}\| \leq \gamma_{\hat{\theta}} \triangleq \gamma_{F_{j_{kf}}}(1+\gamma_e).$$

e) Term-by-term bounds.:

$$\big|e^\top Pg(x)\hat{\Phi}'_{F_{j_{kf}}}\theta^*_{F_{j_{kf}}}\big| \leq \gamma_e\gamma_{F_{j_{kf}}}\overline{\theta}_{F_{j_{kf}}}\|e\|,$$

$$\big|e^\top Pg(x)(\epsilon + \eta)\big| \leq \gamma_e\rho\|e\|,$$

$$\big|e^\top M_{F_{j_{kf}}}\hat{\theta}_{F_{j_{kf}}}\big| \leq \gamma_{\hat{\theta}}\|e\|\,\|\hat{\theta}_{F_{j_{kf}}}\|.$$

Apply Young's inequality to the last coupling term with design constant $\alpha_3 > 0$:

$$\gamma_{\hat{\theta}}\|e\|\,\|\hat{\theta}\| \leq \frac{1}{2\alpha_3}\|e\|^2 + \frac{\alpha_3\gamma_{\hat{\theta}}^2}{2}\|\hat{\theta}\|^2.$$

f) Final Lyapunov bound.: Collecting the bounds into (20) yields

$$\begin{aligned}\dot{V}_3 \leq \Big(-\tfrac{1}{2}\lambda_{\min}(Q) + \tfrac{1}{2\alpha_3}\Big)\|e\|^2 + \big(\gamma_e\gamma_{F_{j_{kf}}}\overline{\theta}_{F_{j_{kf}}} + \gamma_e\rho\big)\|e\|\\ + \Big(\tfrac{\alpha_3\gamma_{\hat{\theta}}^2}{2} - \lambda_{F_{j_{kf}}}\Big)\|\hat{\theta}_{F_{j_{kf}}}\|^2.\end{aligned} \tag{21}$$

If the parameters are chosen such that: $\alpha_3 > \frac{1}{\lambda_{\min}(Q)}$ and $\lambda_{F_{j_{kf}}} > \frac{\alpha_3\gamma_{\hat{\theta}}^2}{2}$. Define

Then:

$$k_e \triangleq \tfrac{1}{2}\lambda_{\min}(Q) - \tfrac{1}{2\alpha_3} > 0, \qquad k_\theta \triangleq \lambda_{F_{j_{kf}}} - \tfrac{\alpha_3\gamma_{\hat{\theta}}^2}{2} > 0,$$

and

$$c \triangleq \gamma_e\big(\gamma_{F_{j_{kf}}}\overline{\theta}_{F_{j_{kf}}} + \rho\big).$$

From (21)

$$\dot{V}_3 \leq -k_e\|e\|^2 - k_\theta\|\hat{\theta}_{F_{j_{kf}}}\|^2 + c\|e\|,$$

The final bound for the combined Lyapunov function derivative in Case 2 is given by:

$$\dot{V}_3 \leq -k_e\|e\|^2 - k_\theta\|\hat{\theta}_{F j_{kf}}\|^2 + c\|e\|.$$

Since the design parameters are chosen such that $k_e > 0$ and $k_\theta > 0$, the derivative $\dot{V}_3$ is guaranteed to be negative outside a compact set, confirming that the combined state $(e, \hat{\theta}_{F j_{kf}})$ is Ultimately Uniformly Bounded (UUB).

The ultimate bounding sets are therefore given by:

1) Ultimate Bounding Set for Tracking Error (Ω_e):

$$e \in \Omega_e \triangleq \left\{ e \mid \|e\| \leq \frac{c}{k_e} \right\}$$

2) Ultimate Bounding Set for Weights ($\Omega_{\hat{\theta}}$):

$$\hat{\theta}_{F j_{kf}} \in \Omega_{\hat{\theta}_{F j_{kf}}} \triangleq \left\{ \hat{\theta}_{F j_{kf}} \mid \|\hat{\theta}_{F j_{kf}}\| \leq \sqrt{\frac{c}{k_\theta}\|e\|} \right\}$$

Thus, the tracking error e and the output layer weights $\hat{\theta}_{F j_{kf}}$ are UUB.

C. Boundedness of Inner Layer Weights ($\hat{\theta}_j$ for j being an Inner Layer)

The boundedness of all inner layers is established recursively, relying fundamentally on the boundedness of the outermost layer weights.

The Lyapunov derivative for a generic inner block $\hat{\theta}_j$ is bounded by:

$$\dot{V}_{\hat{\theta}_j} \leq \gamma_j\|e\|\|\hat{\theta}_j\| + \lambda_\Phi\gamma_j\|\Phi\|\|\hat{\theta}_j\| - \lambda_j\|\hat{\theta}_j\|^2 \tag{22}$$

where $\gamma_j \triangleq \|\Phi'_j\|$ is the norm of the backpropagated Jacobian associated with layer j.

The stability of $\hat{\theta}_j$ relies on the crucial condition that the Jacobian norm $\gamma_j = \|\Phi'_j\|$ is bounded if and only if all subsequent weights $\|\hat{\theta}_k\|$ (where k is closer to the output) are bounded.

1) Base Case (Output Layer $\hat{\theta}_{F_{j_{kf}}}$): The output-layer weights $\hat{\theta}_{F_{j_{kf}}}$ were rigorously proven to be uniformly ultimately bounded (UUB) in the previous section (Case 1 and Case 2 analysis).
2) Recursive Step (Backward Pass): We proceed recursively from the output layer towards the input.
   - Fully Connected Layer (FCL) Weights ($\hat{\theta}_{F_0}, \dots$): Consider the layer immediately preceding the output, $\hat{\theta}_{F_0}$. The associated Jacobian Φ'_{F_0} depends on $\hat{\theta}_{F_{j_{kf}}}$ through the activation derivative $\phi'_{F_{j_{kf}}}$. Since $\|\hat{\theta}_{F_{j_{kf}}}\|$ is UUB, it follows that $\|\Phi'_{F_0}\|$ is bounded by a constant γ_{F_0}. Applying this bound to (22) implies that $\hat{\theta}_{F_0}$ is UUB.
   - Convolutional Layer (CL) Weights ($\hat{\theta}_{C_i}, \dots$): The Jacobians for the convolutional layers, Φ'_{C_i}, depend on the weights of all subsequent layers (both

FCL and CL layers closer to the output). Since all subsequent weights are recursively shown to be bounded, each Jacobian norm γ_{C_i} is bounded. Consequently, applying the bound to (22) shows that all convolutional-layer weights $\hat{\theta}_{C_i}$ are UUB.

Thus, the imposition of the weight norm constraints c_j and the resulting non-zero Lagrange multipliers λ_j ensures, through a recursive argument stemming from the bounded output layer, that all inner layer weight vectors are UUB.

VI. Simulation Results

Appendix

References

[1] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton University Press, 2009. [Online]. Available: http://www.jstor.org/stable/j.ctt7t833