Event-Triggered Neural Network Control for LTI Systems

C. de Souza, Sophie Tarbouriech, A. Girard

Event-triggered Neural Network Control for LTI systems

C. de Souza, S. Tarbouriech, and A. Girard

***Abstract*— This paper addresses the problem of event-triggered control (ETC) of discrete-time linear time-invariant (LTI) systems stabilized by neural network controllers. The event-triggering mechanisms (ETMs) is proposed to update only a portion of the layers required to maintain stability and satisfactory performance of the feedback system, thus reducing the computational cost associated with the evaluation of the neural network. Sufficient convex conditions in the form of linear matrix inequalities (LMIs) are provided to compute the triggering parameters and characterize an estimate of the domain of attraction for the feedback system. The formulation is based on the use of Lyapunov theory and a set of generalized sector constraints that deal with the nonlinear activation functions, represented in this case by the saturation function. Optimization procedures are also formulated to effectively reduce the amount of computation in the neural network. An example borrowed from the literature is used to illustrate the effectiveness of the proposal.**

***Index Terms*— Event-triggered control; Neural network control; Quadratic constraints; Linear Matrix inequalities; Lyapunov theory.**

I. Introduction

NEURAL networks (NNs) have become increasingly effective at many difficult machine-learning tasks. However, they may suffer from a lack of guarantees due to their complex structure. In particular, NNs have various types of nonlinear activation functions, potentially numerous layers, and a large number of hidden neurons, making it difficult to apply classical analysis methods, such as Lyapunov theory. This drawback limits their use in safety-critical applications such as self-driving vehicles, aircraft collision avoidance procedures, and robots for surgical procedures (see e.g. [1] for a survey). In response, a major research effort has been undertaken to develop tools that provide useful certificates of stability, safety, and robustness for NNs. Several approaches have proposed the use of the quadratic constraints (QCs) formalism to bound the nonlinear activation functions. [2] uses QCs to upper-bound the Lipschitz constant of deep feed-forward neural networks, which is a useful indicator of robustness. [3] verifies the safety of NNs by using QCs to outer-bound their outputs given a set of inputs. [4] combines Lyapunov theory with local sector QCs to analyze the stability of feedback systems with neural network controllers. Such an approach also allows to include perturbations described by integral quadratic constraints (IQCs) in the system. As an extension, [5] analyzes stability in offset-free setpoint tracking with a piecewise constant reference. Both works provide ellipsoidal inner approximations of the corresponding regions of attraction. [6] formulates QCs based on partial gradients to certify the input-output stability of reinforcement-learning (RL) controlled systems.

Unlike traditional control systems, network control systems not only need to provide satisfactory control requirements but also need to consider the usage of the communication network. The event-triggered control (ETC) has emerged as an alternative control paradigm to reduce the computational overhead and communication traffic, thus saving the limited network resources. Instead of executing control tasks at periodic intervals, the ETC approach provides a mechanism to determine the sampling instants without compromising the desired system performance. In recent years, several event-triggering mechanisms (ETMs) have been proposed in the literature (see, for instance, [7]–[10]). Regarding the neural networks, the ETC technique has been mostly used to update states during the learning process. [11] investigates an event-triggered state-feedback control for multi-input-multi-output (MIMO) uncertain nonlinear continuous-time systems in affine form, where the controller is approximated by a linear parameterized NN (i.e. a neural network where all nonlinearites are confined to a single layer). [12] proposes an optimal adaptive event-triggered state-feedback control algorithm based on an actor/critic NN structure for nonlinear continuous-time systems. [13] designs a four-layer fully connected feed-forward NN controller based on event-triggering data updating (state) for the stability of continuous-time nonlinear systems. A genetic algorithm (GA) is also used to optimize the initial weights and thresholds of the NN to reduce the error of the controller. From a different perspective, in which the neural network has already been trained, an early work [14] proposes an ETM (based on a local sector) to transmit the output of the layers, thus allowing to reduce the computational cost associated with the control law evaluation.

Following the same idea, but under different considerations, the present work proposes: *i)* the design of ETMs (based on standard error functions) that allow us to compute the control law by updating only a portion of the layers instead of providing the periodic sampling of the entire network; *ii)* the development of sufficient conditions to compute the triggering parameters and characterize an estimate of the domain of

C. de Souza and S. Tarbouriech (carla.souza93@hotmail.com and sophie.tarbouriech@laas.fr) are with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France.

A. Girard (antoine.girard@l2s.centralesupelec.fr) is with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes, 91190, Gif-sur-Yvette, France.

attraction for the feedback system. The approach applies to discrete-time linear time-invariant (LTI) plants stabilized by known neural network controllers. In contrast to [14] that employs the hyperbolic tangent (tanh) as the activation function, the activation function considered in the paper is a saturation function. This choice is motivated by noting that the saturation can be seen as an approximation of smooth functions as tanh and also sigmoid. In addition, existing tools for handling saturations lead to less conservative conditions for stability and performance purposes [15], [16]. Thus, by using Lyapunov theory and a set of generalized sector constraints to deal with the saturation, we propose convex conditions in the form of linear matrix inequalities (LMIs) that guarantee the regional (local) asymptotic stability of the feedback system. Optimization procedures incorporating these conditions are also established to effectively reduce the update of the neural network layers. An example borrowed from the literature is used to illustrate the effectiveness of the proposal.

Notation. $\mathbb{N}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote respectively the sets of non-negative integers, n-dimensional vectors and $n\times m$ matrices. For any matrix A, $A^\top$ denotes its transpose. For any square matrix A, $\mathtt{trace}(A)$ denotes its trace and $He\{A\} = A + A^\top$. $\mathtt{diag}(A_1, A_2)$ is a block-diagonal matrix with block diagonal matrices A_1 and A_2. For two symmetric matrices of same dimensions, A and B, $A > B$ means that $A - B$ is symmetric positive definite. $\mathbf{I}$ and $\mathbf{0}$ stand respectively for the identity and the null matrix of appropriate dimensions. For a partitioned matrix, the symbol $\star$ stands for symmetric blocks. For any vector $x \in \mathbb{R}^n$ and any symmetric positive definite (or semi-positive definite) matrix, $\|x\|_Q^2$ denotes the quadratic form $x^\top Q x$.

II. Problem Statement

A. Neural Network Model

Consider the feedback-system shown in Figure 1, which consists of a nominal plant F and an event-triggered neural network controller π_{ETM}.

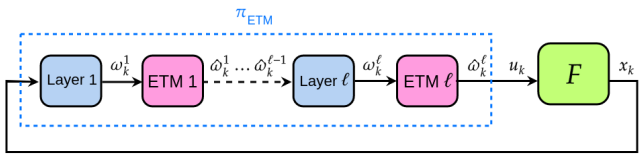

Fig. 1: Feedback system subject to ETMs.

The nominal plant F is a discrete-time linear time-invariant (LTI) system according to the model:

$$x(k+1) = A_F x(k) + B_F u(k), \tag{1}$$

where $x(k) \in \mathbb{R}^{n_F}$ is the state vector and $u(k) \in \mathbb{R}^{n_u}$ is the control input. The controller π_{ETM} is an ℓ-layer, event-triggered feed-forward neural network (NN) defined by:

$$\begin{aligned} \hat{\omega}^0(k) &= x(k), \\ \nu^i(k) &= W^i \hat{\omega}^{i-1}(k) + b^i, \ i \in \{1, \dots, \ell\}, \\ \omega^i(k) &= \mathtt{sat}(\nu^i(k)), \\ u(k) &= W^{\ell+1} \hat{\omega}^\ell(k) + b^{\ell+1}, \end{aligned} \tag{2}$$

where $\nu^i \in \mathbb{R}^{n_i}$ is the input to the i^{th} activation function, $\omega^i \in \mathbb{R}^{n_i}$ and $\hat{\omega}^i \in \mathbb{R}^{n_i}$ are the current output and the last updated output from the i^{th} layer, respectively, and $\mathtt{sat}(\cdot)$ is the decentralized vector-valued saturation function defined by

$$\mathtt{sat}(\nu_j^i(k)) = \mathtt{sign}(\nu_j^i(k)) \min(|\nu_j^i(k)|, \bar{\nu}_j^i), \tag{3}$$

where ν_j^i is the j^{th} component of ν^i, and $\bar{\nu}_j^i > 0$, $i \in [1, \ell]$, $j \in [1, n_i]$, is the maximum ($\underline{\nu}_j^i = -\bar{\nu}_j^i$ minimum) limit of the input signals. The operations for each layer are defined by a weight matrix $W^i \in \mathbb{R}^{n_i \times n_{i-1}}$, a bias vector $b^i \in \mathbb{R}^{n_i}$, and the saturation function $\mathtt{sat}(\cdot)$, which is applied element-wise, according to (3).

It is useful to isolate the nonlinear functions from the linear operations in the neural network [3], [4]. In this case, we can gather the inputs and outputs of all saturation functions into augmented vectors as

$$\begin{aligned} &\nu_\phi = \begin{bmatrix} \nu^{1\top} & \dots & \nu^{\ell\top} \end{bmatrix}^\top, \ \omega_\phi = \begin{bmatrix} \omega^{1\top} & \dots & \omega^{\ell\top} \end{bmatrix}^\top, \text{ and} \\ &\hat{\omega}_\phi = \begin{bmatrix} \hat{\omega}^{1\top} & \dots & \hat{\omega}^{\ell\top} \end{bmatrix}^\top \in \mathbb{R}^{n_\phi}, \end{aligned}$$

with $n_\phi = \sum_{i=1}^{\ell} n_i$. The i^{th} element of ν_ϕ, for instance, is denoted by $\nu_{\phi,i}$. A combined nonlinearity $\mathtt{sat}(\cdot) : \mathbb{R}^{n_\phi} \to \mathbb{R}^{n_\phi}$ can also be obtained by stacking all saturation functions, i.e.

$$\mathtt{sat}(\nu_\phi) = \begin{bmatrix} \mathtt{sat}(\nu^1)^\top & \dots & \mathtt{sat}(\nu^\ell)^\top \end{bmatrix}^\top \in \mathbb{R}^{n_\phi}, \tag{4}$$

which leads to $\omega_\phi(k) = \mathtt{sat}(\nu_\phi)$. From that, the neural network control policy takes the form:

$$\begin{bmatrix} u(k) \\ \nu_\phi(k) \end{bmatrix} = N \begin{bmatrix} x(k) \\ \hat{\omega}_\phi(k) \\ 1 \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} N &= \left[\begin{array}{c|ccc|c|c} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & W^{\ell+1} & b^{\ell+1} \\ \hline W^1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & b^1 \\ \mathbf{0} & W^2 & \dots & \mathbf{0} & \mathbf{0} & b^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & W^\ell & \mathbf{0} & b^\ell \end{array}\right], \\ &= \begin{bmatrix} N_{ux} & N_{u\omega} & N_{ub} \\ N_{\nu x} & N_{\nu\omega} & N_{\nu b} \end{bmatrix}. \end{aligned} \tag{6}$$

Remark 1: Note that in our approach $N_{ux} = \mathbf{0}$. However, this matrix can assume a non-null value, e.g. $K \in \mathbb{R}^{n_u \times n_F}$, in the case where $u(k)$ is given by the sum of the expression in (2) with a linear term $Kx(k)$.

By denoting $(x_*, u_*, \nu_*, \omega_*)$ the equilibrium point of (1) and (5)-(6), the following conditions must be satisfied

$$\begin{aligned} x_* &= A_F x_* + B_F u_*, \ \omega_*^0 = x_*, \\ \begin{bmatrix} u_* \\ \nu_* \end{bmatrix} &= N \begin{bmatrix} x_* \\ \omega_* \\ 1 \end{bmatrix}, \\ \omega_* &= \mathtt{sat}(\nu_*). \end{aligned} \tag{7}$$

B. Event-triggering strategy

To reduce the computational cost associated with the neural network evaluation, we introduce an ETM at the output of each layer, according to Figure 1. They decide whether or not to update the current outputs through the neural network.

Such decisions are made based on the following rule:

$$\hat{\omega}^i(k) = \begin{cases} \omega^i(k), & \text{if } f^i(\omega^i(k), \hat{\omega}^i(k-1), \hat{\omega}^{i-1}(k)) > 0, \\ \hat{\omega}^i(k-1), & \text{otherwise,} \end{cases} \tag{8}$$

where $f^i(\omega^i(k), \hat{\omega}^i(k-1), \hat{\omega}^{i-1}(k))$ is defined by

$$f^i(\omega^i(k), \hat{\omega}^i(k-1), \hat{\omega}^{i-1}(k)) = \|\hat{\omega}^i(k-1) - \omega^i(k)\|^2_{q^i_\Delta} - \|\omega^i(k) - \omega^i_*\|^2_{q^i_\omega} - \|\hat{\omega}^{i-1}(k) - \omega^{i-1}_*\|^2_{q^{i-1}_{\hat{\omega}}} \tag{9}$$

where $q^i_\Delta, q^i_\omega, q^{i-1}_{\hat{\omega}} \in \mathbb{R}^{n_i \times n_i}$ are symmetric positive definite matrices to be designed. Since these matrices act as weights on the terms associated with the triggering condition, their computation directly affects the event-triggering rule and consequently how much the data update can be reduced.

It is important to point out that whenever a layer is not updated, the following ones are not updated either. The reason for this is that $\omega^i(k)$ equals to $\hat{\omega}^i(k-1)$, consequently f^i becomes negative, preventing the event from being triggered. Thus, in this case, we do not need to evaluate the subsequent triggering rules anymore, but keep the outputs with their previous values. The proposed ETM requires to store the values of $\hat{\omega}^i$, hence the memory requirements are proportional to n_ϕ the size of the neural network.

Let us define the error vector $e^i(k) \in \mathbb{R}^{n_i}$ on the i^{th} layer as:

$$e^i(k) = \hat{\omega}^i(k) - \omega^i(k). \tag{10}$$

If $\omega^i(k)$ is updated at instant k, from (8), it follows that $e^i(k) = \hat{\omega}^i(k) - \omega^i(k) = \omega^i(k) - \omega^i(k) = \mathbf{0}$. On the other hand, if $\omega^i(k)$ is not updated at instant k, from (8), we have that $e^i(k) = \hat{\omega}^i(k) - \omega^i(k) = \hat{\omega}^i(k-1) - \omega^i(k)$. Thus, the following inequality is always satisfied.

$$\|e^i(k)\|^2_{q^i_\Delta} \leq \|\omega^i(k) - \omega^i_*\|^2_{q^i_\omega} + \|\hat{\omega}^{i-1}(k) - \omega^{i-1}_*\|^2_{q^{i-1}_{\hat{\omega}}}. \tag{11}$$

Based on the above, the problem we intend to solve can be summarized as:

Problem 1: Consider that the NN controller π_{ETM} (5) stabilizes the LTI system (1). Design the triggering parameters q^i_Δ, q^i_ω and $q^{i-1}_{\hat{\omega}}$, $i \in \{1, \ldots, \ell\}$, in (9), to reduce the computational cost associated with the control law evaluation, while maintaining the stability of the closed-loop system.

Note that an implicit goal in solving Problem 1 is to characterize an estimate of the region of attraction for the feedback system. Such a region can be defined as the set of all points x for which $x(0) = x$ leads to trajectories $x(k)$ that converge asymptotically toward the equilibrium point x_*, i.e. $\lim_{k\to\infty} x(k) = x_*$. Since its geometric characterization is not a simple task [15], we compute subsets with well-fitted representation, such as sublevel sets associated to Lyapunov function.

III. Auxiliary results

To deal with the combined saturation $\mathtt{sat}(\nu_\phi)$, we have the following property directly derived from [15, Lemma 1.6, page 43].

Lemma 1: Consider a matrix $G = \begin{bmatrix} G^{1\top} & \ldots & G^{\ell\top} \end{bmatrix}^\top$ with $G^i \in \mathbb{R}^{n_i \times n_F}$, $i \in \{1, \ldots, \ell\}$. For given vectors ν_* and $\bar{\nu} = \begin{bmatrix} \bar{\nu}^{1\top} & \ldots & \bar{\nu}^{\ell\top} \end{bmatrix}^\top$ satisfying $-\bar{\nu} \leq \nu_* \leq \bar{\nu}$ and, consequently, $\omega_* = \mathtt{sat}(\nu_*) = \nu_*$, if $x(k)$ belongs to the set $\mathcal{S}$ defined by

$$\mathcal{S} = \{x(k), x_* \in \mathbb{R}^{n_F} : -\bar{\nu} - \nu_* \leq G(x(k) - x_*) \leq \bar{\nu} - \nu_*\}$$

then, the descentralized nonlinearity $\omega_\phi(k)$ satisfies the following inequality:

$$\Psi^\top T[G(x(k) - x_*) - (\omega_\phi(k) - \omega_*)] \leq 0 \tag{12}$$

with $\Psi = \nu_\phi(k) - \omega_\phi(k)$, for any diagonal positive definite matrix $T \in \mathbb{R}^{n_\phi \times n_\phi}$.

Proof: For $i \in \{1, \ldots, n_\phi\}$, assume that $-\bar{\nu}_i \leq \nu_{*,i} \leq \bar{\nu}_i$, which implies that $\omega_{*,i} = \mathtt{sat}(\nu_{*,i}) = \nu_{*,i}$. If $(x(k), x_*)$ are elements of $\mathcal{S}$, it follows that $G_i(x(k) - x_*) - (\bar{\nu}_i - \nu_{*,i}) \leq 0$ and $G_i(x(k) - x_*) - (-\bar{\nu}_i - \nu_{*,i}) \geq 0$. Consider now the three cases below.

- Case 1: $\nu_{\phi,i}(k) > \bar{\nu}_i$. It follows that $\Psi_i = \nu_{\phi,i}(k) - \bar{\nu}_i > 0$ and one gets $\Psi_i^\top T_{i,i}[G_i(x(k) - x_*) - (\bar{\nu}_i - \nu_{*,i})] \leq 0$.
- Case 2: $\nu_{\phi,i}(k) < -\bar{\nu}_i$. It follows that $\Psi_i = \nu_{\phi,i} - (-\bar{\nu}_i) < 0$ and one gets $\Psi_i^\top T_{i,i}[G_i(x(k) - x_*) - (-\bar{\nu}_i - \nu_{*,i})] \leq 0$.
- Case 3: $-\bar{\nu}_i < \nu_{\phi,i} < \bar{\nu}_i$. It follows that $\Psi_i = 0$ and $\Psi_i^\top T_{i,i}[G_i(x(k) - x_*) - (\omega_{\phi,i} - \omega_{*,i})] \leq 0$.

■

Lemma 1 is based on the assumption that: $-\bar{\nu} \leq \nu_* \leq \bar{\nu}$, which implies that $\omega_* = \mathtt{sat}(\nu_*) = \nu_*$. By using this information in (7), we have that:

$$\begin{aligned} x_* &= [\mathbf{I}_{n_F} - A_F - B_F R_\omega]^{-1} B_F R_b, \\ u_* &= R_w x_* + R_b, \\ \nu_* &= R N_{\nu x} x_* + R N_{\nu b}, \\ \omega_* &= \nu_*, \end{aligned} \tag{13}$$

where $R = (\mathbf{I}_{n_F} - N_{\nu\omega})^{-1}$, $R_b = N_{u\omega} R N_{\nu b} + N_{ub}$ and $R_w = N_{ux} + N_{u\omega} R N_{\nu x}$. Therefore, $(x_*, u_*, \omega_*, \nu_*)$ satisfying (13) is an equilibrium point of (1) and (5)-(6) if

$$-\bar{\nu} \leq R N_{\nu x} x_* + R N_{\nu b} \leq \bar{\nu}. \tag{14}$$

IV. Lyapunov Condition

In this section, a Lyapunov function and Lemma 1 are used to compute ETMs aiming at reducing the computational burden in the neural network while maintaining the stability of the feedback system. An estimate of its domain of attraction is also characterized.

Theorem 1: Consider the feedback system consisting of F in (1) and π_{ETM} in (2) with equilibrium point $(x_*, u_*, \nu_*, \omega_*)$ satisfying (13)-(14). Let $\bar{\nu} \in \mathbb{R}^{n_\phi}$ be given and define the matrix

$$R_\phi = \begin{bmatrix} \mathbf{I}_{n_F} & \mathbf{0}_{n_F \times n_\phi} & \mathbf{0}_{n_F \times n_\phi} \\ N_{\nu x} & N_{\nu\omega} & N_{\nu\omega} \\ \mathbf{0}_{n_\phi \times n_F} & \mathbf{I}_{n_\phi} & \mathbf{0}_{n_\phi \times n_F} \end{bmatrix}$$

If there exist symmetric positive definite matrices $P, q^0_{\hat{\omega}} \in \mathbb{R}^{n_F \times n_F}$, $q^i_\Delta, q^i_\omega \in \mathbb{R}^{n_i \times n_i}$, $q^j_{\hat{\omega}} \in \mathbb{R}^{n_j \times n_j}$, a diagonal positive definite matrix $T \in \mathbb{R}^{n_\phi \times n_\phi}$, a matrix $Z = \begin{bmatrix} Z^{1\top} & \dots & Z^{\ell\top} \end{bmatrix}^\top$, $Z^i \in \mathbb{R}^{n_i \times n_F}$, with $i \in \{1, \dots, \ell\}$ and $j \in \{1, \dots, \ell-1\}$, and a scalar $0 < \alpha \in \mathbb{R}$, such that the LMI (15) (at the top of the next page) and

$$\begin{bmatrix} P & Z_i^\top \\ \star & 2\alpha T_{i,i} - \alpha^2 \hat{\nu}_i^{-2} \end{bmatrix} \geq \mathbf{0}, \ \forall i \in \{1, \dots, n_\phi\}, \tag{16}$$

where $Q_{\hat{\omega}} = \texttt{diag}(q^1_{\hat{\omega}}, \dots, q^{\ell-1}_{\hat{\omega}}, \mathbf{0})$, $Q_\Delta = \texttt{diag}(q^1_\Delta - q^1_{\hat{\omega}}, \dots, q^{\ell-1}_\Delta - q^{\ell-1}_{\hat{\omega}}, q^\ell_\Delta), Q_\omega = \texttt{diag}(q^1_\omega + q^1_{\hat{\omega}}, \dots, q^{\ell-1}_\omega + q^{\ell-1}_{\hat{\omega}}, q^\ell_\omega)$, and $\hat{\nu}_i = \min(|-\bar{\nu}_i - \nu_{*,i}|, |\bar{\nu}_i - \nu_{*,i}|)$, hold. Then:

1) the feedback system F and π_{ETM} is locally stable around x_*,
2) the set $\mathcal{E}(P, x_*) = \{x \in \mathbb{R}^{n_F} : (x - x_*)^\top P (x - x_*) \leq 1\}$ is an estimate of the domain of attraction for the feedback system.

Proof: First of all, let us define the candidate Lyapunov function as $V(x(k)) = (x(k) - x_*)^\top P (x(k) - x_*)$ with $\mathbf{0} < P = P^\top \in \mathbb{R}^{n_F \times n_F}$. Suppose the feasibility of (16) and use the fact that $[\alpha \hat{\nu}_i^{-2} - T_{i,i}] \hat{\nu}_i^2 [\alpha \hat{\nu}_i^{-2} - T_{i,i}] \geq 0$ or equivalently $T^2_{i,i} \hat{\nu}_i^2 \geq 2\alpha T_{i,i} - \alpha^2 \hat{\nu}_i^{-2}$. By considering the change of variables $Z_i = T_{i,i} G_i$, we have that (16) results in

$$\begin{bmatrix} P & G_i^\top T_{i,i} \\ \star & T^2_{i,i} \hat{\nu}_i^2 \end{bmatrix} \geq \mathbf{0}, \ \forall i \in \{1, \dots, n_\phi\}.$$

Then, pre- and post-multiply this inequality by $\texttt{diag}(\mathbf{I}_{n_F}, T^{-1}_{i,i})$ and its transpose, respectively, and apply Schur complement. By pre-and post multiplying the resulting inequality, respectively, by $(x - x_*)^\top$ and $(x - x_*)$, we obtain

$$(x - x_*)^\top G_i^\top \hat{\nu}_i^{-2} G_i (x - x_*) \leq (x - x_*)^\top P (x - x_*).$$

Note that the bounds of the set $\mathcal{S}$ are asymmetric. Then, one can consider the minimal bound in absolute value as the bound on the saturation: $\hat{\nu}_i = \min(|-\bar{\nu}_i - \nu_{*,i}|, |\bar{\nu}_i - \nu_{*,i}|)$. The previous inequality ensures that $\mathcal{E}(P, x_*) \subseteq \mathcal{S}$, consequently, Lemma 1 applies.

Moreover, since the LMI in (15) is strict, there exists a scalar $\epsilon > 0$, such that pre- and post multiplying (15) by $\begin{bmatrix} (x(k) - x_*)^\top & (\omega_\phi(k) - \omega_*)^\top & e_\phi(k)^\top \end{bmatrix}$ and its transpose, respectively, replacing Z by TG, and rearranging terms, yields (17) (at the top of the next page). By denoting $e_\phi(k) = \hat{\omega}_\phi(k) - \omega_\phi(k)$ and considering $u(k) = N_{ux} x(k) + N_{u\omega} \hat{\omega}_\phi(k)$, we can replace $A_F x(k) + B_F u(k)$ by $x(k+1)$, according to (1), to obtain

$$\begin{aligned} &(x(k+1) - x_*)^\top P (x(k+1) - x_*) - (x(k) - x_*)^\top P (x(k) - x_*) \\ &- 2[\nu_\phi(k) - \nu_* - (\omega_\phi(k) - \omega_*)]^\top T [G(x - x_*) - (\omega_\phi(k) - \omega_*)] \\ &- e_\phi(k)^\top Q_\Delta e_\phi(k) + (\omega_\phi(k) - \omega_*)^\top Q_\omega (\omega_\phi(k) - \omega_*) \\ &+ 2 e_\phi(k)^\top Q_{\hat{\omega}} (\omega_\phi(k) - \omega_*) + (x(k) - x_*)^\top q^0_{\hat{\omega}} (x(k) - x_*) \\ &< -\epsilon \|x(k) - x_*\|^2. \end{aligned}$$

Note that the last four terms on the left-hand side of the previous expression can be rewritten as $\sum_{i=1}^{\ell} \left(\|e_i(k)\|^2_{q^i_\Delta} - \|\omega^i(k) - \omega^i_*\|^2_{q^i_\omega} - \|\hat{\omega}^{i-1}(k) - \omega^{i-1}_*\|^2_{q^{i-1}_{\hat{\omega}}} \right)$ considering that $\hat{\omega}^0(k) = x(k)$ and $e^0(k) = 0$. Also, by denoting $\Psi = \nu_\phi(k) - \nu_* - (\omega_\phi(k) - \omega_*)$ and observing that $\Delta V(x(k)) = V(x(k+1)) - V(x(k)) = (x(k+1) - x_*)^\top P (x(k+1) - x_*) - (x(k) - x_*)^\top P (x(k) - x_*)$, we have that

$$\begin{aligned} &\Delta V(x(k)) - 2\Psi^\top T [G(x(k) - x_*) - (\omega_\phi(k) - \omega_*)] \\ &- \sum_{i=1}^{\ell} \left(\|e_i(k)\|^2_{q^i_\Delta} - \|\omega^i(k) - \omega^i_*\|^2_{q^i_\omega} - \|\hat{\omega}^{i-1}(k) - \omega^{i-1}_*\|^2_{q^{i-1}_{\hat{\omega}}} \right) \\ &< -\epsilon \|x(k) - x_*\|^2. \end{aligned}$$

In view of Lemma 1 and the triggering condition (11), we can conclude that $\Delta V(x(k)) \leq -\epsilon \|x(k) - x_*\|^2$, which implies that if $x(k) \in \mathcal{E}(P, x_*)$ then $x(k+1) \in \mathcal{E}(P, x_*)$ ($\mathcal{E}(P, x_*)$ is an invariant set) and and x converges to the equilibrium point x_*. Therefore, $\mathcal{E}(P, x_*)$ constitutes an inner-approximation of the domain of attraction for the closed-loop system. The proof is complete. ■

The complexity of the proposed LMI-conditions is related with its number of scalar variables, $\mathcal{K}$, and its number of rows, $\mathcal{L}$. In such a case, we have $\mathcal{K} = 1 + (n_F + n_\phi)(n_F + 1) + 1.5 \sum_{i=1}^{\ell} n_i(n_i + 1)$ and $\mathcal{L} = n_F(1 + n_\phi) + 3n_\phi$. For MATLAB solvers Lmilab and SeDumi, the following expressions allow respectively to compute the complexity in terms of $\mathcal{K}$ and $\mathcal{L}$: $\mathcal{K}^3\mathcal{L}$ and $\mathcal{K}^2\mathcal{L}^{2.5} + \mathcal{L}^{3.5}$ [17].

Remark 2: Theorem 1 can be simplified to consider a triggering rule that only takes into account information about the layer being assessed, i.e. the last term of (9) disappears. In this case, we have to assume $Q_\Delta = \texttt{diag}(q^1_\Delta, \dots, q^{\ell-1}_\Delta, q^\ell_\Delta)$, $Q_\omega = \texttt{diag}(q^1_\omega, \dots, q^{\ell-1}_\omega, q^\ell_\omega)$, $Q_{\hat{\omega}} = \mathbf{0}$ and $q^0_{\hat{\omega}} = \mathbf{0}$ on the conditions of Theorem 1.

V. Optimization procedure

This section presents an optimization procedure to reduce the computational cost on the neural network. According to the inequality (8), a solution consists in reducing the weight on the error measure, i.e. by choosing q^i_Δ to shrink $\|\hat{\omega}^i(k-1) - \omega^i(k)\|^2_{q^i_\Delta}$ compared with the magnitude of the norm-sum of the differences $\omega^i(k) - \omega_*$ and $\hat{\omega}^{i-1}(k) - \omega_*$. Then, an intuitive method is to find q^i_Δ as "small" as possible and q^i_ω and $q^{i-1}_{\hat{\omega}}$ as "large" as possible. However, experience shows that this may not be effective due to the generality of the objective function. Therefore, it seems better to impose constraints on some matrices and therefore to consider the following optimization procedure:

$$\mathcal{O}_1 : \begin{cases} \min & -\rho \\ \text{subject to} & (15), (16), \ \texttt{diag}(q^i_\Delta) \leq I_{n_\phi}, \ \text{and} \\ & \texttt{diag}(q^i_\omega, q^{i-1}_{\hat{\omega}}) \geq \rho I_{2n_\phi - n_\ell + n_F}. \end{cases} \tag{18}$$

with $\rho > 0$. Thus, we are bounding q^i_Δ to unity and looking for the largest scalar ρ that lower bounds the weight matrices q^i_ω and $q^i_{\hat{\omega}}$.

Another objective consists in considering a given region of admissible initial states $\mathcal{X}_0$ for which we can reduce the

$$\begin{bmatrix} (A_F+B_FN_{ux})^\top P(A_F+B_FN_{ux})-P+q^0_{\hat{\omega}} & (A_F+B_FN_{ux})^\top PB_FN_{u\omega} & (A_F+B_FN_{ux})^\top PB_FN_{u\omega} \\ \star & (B_FN_{u\omega})^\top PB_FN_{u\omega}+Q_\omega & (B_FN_{u\omega})^\top PB_FN_{u\omega}+Q_{\hat{\omega}} \\ \star & \star & (B_FN_{u\omega})^\top PB_FN_{u\omega}-Q_\Delta \end{bmatrix} + R_\phi^\top \begin{bmatrix} \mathbf{0}_{n_F\times n_\phi} & -Z^\top & Z^\top \\ \star & \mathbf{0}_{n_\phi} & T \\ \star & \star & -2T \end{bmatrix} R_\phi < \mathbf{0}, \quad (15)$$

$$[\bullet]^\top \left(\begin{bmatrix} A_F+B_FN_{ux} \\ B_FN_{u\omega} \\ B_FN_{u\omega} \end{bmatrix}^\top P \begin{bmatrix} A_F+B_FN_{ux} \\ B_FN_{u\omega} \\ B_FN_{u\omega} \end{bmatrix} + \begin{bmatrix} -P+q^0_{\hat{\omega}} & \mathbf{0} & \mathbf{0} \\ \star & Q_\omega & Q_{\hat{\omega}} \\ \star & \star & -Q_\Delta \end{bmatrix} \right) \begin{bmatrix} x(k)-x_* \\ \omega_\phi(k)-\omega_* \\ e_\phi(k) \end{bmatrix} + [\bullet]^\top \begin{bmatrix} \mathbf{0}_{n_F\times n_\phi} & -G^\top T & G^\top T \\ \star & \mathbf{0}_{n_\phi} & T \\ \star & \star & -2T \end{bmatrix} \begin{bmatrix} x(k)-x_* \\ \nu_\phi(k)-\nu_* \\ \omega_\phi(k)-\omega_* \end{bmatrix} \leq -\epsilon\|x(k)-x_*\|^2. \quad (17)$$

computational cost on the neural network. In this case, we should ensure that $\mathcal{X}_0$ is contained in the region of attraction of the closed-loop system, i.e. $\mathcal{X}_0 \subseteq \mathcal{E}(P, x_*)$. If $\mathcal{X}_0$ is characterized by the ellipsoid $\mathcal{E}(X, x_*) = \{x \in \mathbb{R}^{n_F} : (x-x_*)^\top X(x-x_*) \leq 1\}$ with $\mathbf{0} < X \in \mathbb{R}^{n_F\times n_F}$ given, then the inclusion is obtained by imposing

$$\begin{bmatrix} X & P \\ P & P \end{bmatrix} \geq \mathbf{0}. \quad (19)$$

Thus, a second optimization procedure $\mathcal{O}_2$ can be defined similarly to $\mathcal{O}_1$, but by adding the constraint(19). It is important to stress that, the optimization procedure $\mathcal{O}_2$ deals with the classical trade-off between the size of the estimate of the region of attraction and the update saving. Indeed, the bigger is the region of attraction, the smaller is the update saving.

VI. Simulations

Consider the inverted pendulum system with mass $m = 0.15\ kg$, length $l = 0.5\ m$, and friction coefficient $\mu = 0.5\ Nms/rad$. Its dynamics is described by the following discrete-time model:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ \dfrac{g\delta}{l} & 1-\dfrac{\delta\mu}{ml^2} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{\delta}{ml^2} \end{bmatrix} u(k), \quad (20)$$

where the states $x_1(k)$ and $x_2(k)$ represent respectively the angular position (rad) and velocity (rad/s), $u(k)$ is the control input (Nm) and $\delta = 0.02$ is the sampling time [18].

To stabilize (20), we use a controller π under the form of a 2-layer, feedforward neural network, with $n_1 = n_2 = 32$ and the saturation as the activation function for both layers. For training purpose only, we replaced the saturation by its smooth approximation provided by $\tanh$, making it possible to rely on MATLAB's Reinforcement Learning toolbox. During the training, the agent's decision is characterized by a Gaussian distribution probability with mean $\pi(x(k))$ and standard deviation σ. Also, to illustrate the applicability of our condition to equilibrium points different from 0, we have not set the bias in neural network to zero during training. After training, the policy mean π is used as the deterministic controller $u(k) = \pi(x(k))$ with the saturation as the activation function.

First, we design ETMs to reduce the amount of computation on the neural network. By assuming $\bar{\nu} = -\underline{\nu} = 1 \times 1_{64\times 1}$, we use the optimization procedure $\mathcal{O}_2$ with

$$X = \begin{bmatrix} 0.3024 & 0.0122 \\ \star & 0.0154 \end{bmatrix}, \quad (21)$$

chosen based on the maximum estimate of the region of attraction obtained without trigger, $\alpha = 9\times 10^{-4}$ and the following restrictions: $\mathtt{diag}(q^1_\Delta, q^2_\Delta) \leq \mathbf{I}_{n_\phi}$ and $\mathtt{diag}(q^1_\omega, q^2_\omega, q^0_{\hat{\omega}}) \leq \rho\mathbf{I}_{n_\phi}$. We then simulate the response of the feedback system for 1000 initial conditions belonging to the estimate of the region of attraction. We found an average update rate (the ratio between the number of events and the number of samplings) of about 15% for both layers, thus reducing almost 85% of data update in the neural network.

Figure 2 depicts the estimate of the region of attraction $\mathcal{E}(P, x_*)$ (blue solid line) and the set of admissible initial states $\mathcal{E}(X, x_*)$ (red solid line). Some convergent and divergent trajectories are also shown in cyan dashed lines and green dash-dotted lines, starting from the points marked with cyan circles and green asterisks, respectively. Note that the estimate of the region of attraction does not overstep the bounds $\{-\bar{\nu}-\nu_* \leq G(x-x_*) \leq \bar{\nu}-\nu_*\}$ (orange lines) for both layers, as expected.

Furthermore, for the convergent trajectory (dark solid line) starting from the initial condition $x(0) = \begin{bmatrix} -0.4274 & -7.7563 \end{bmatrix}^\top$, we plot the temporal response of the closed-loop system in Figure 3. We can see that the trajectories of the states converge to the equilibrium point $x_* = \begin{bmatrix} -0.1147 & 0 \end{bmatrix}^\top$, which verifies the relations (13)-(14), with an update rate of about 12% for both layers, thus reducing the computational cost associated with the control law evaluation.

Finally, Figure 4 shows for each layer the output of the first neuron, i.e. ω^1_1 and ω^2_1. As can be seen, unlike ω^1_1, ω^2_1 saturates

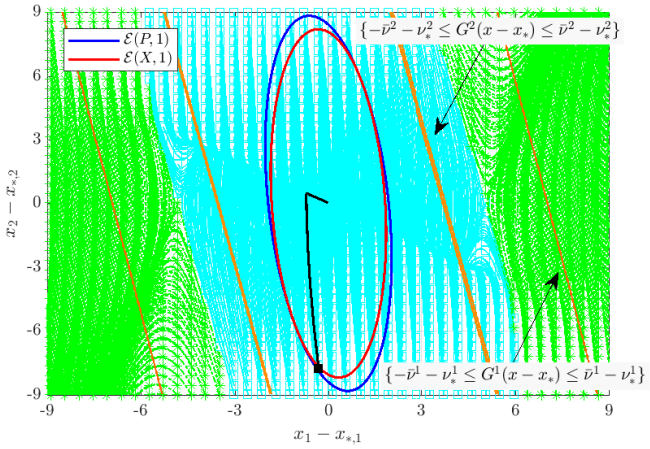

Fig. 2: Estimate of the region of attraction for the feedback system in the plan $(x_1 - x_{*,1} \times x_2 - x_{*,2})$.

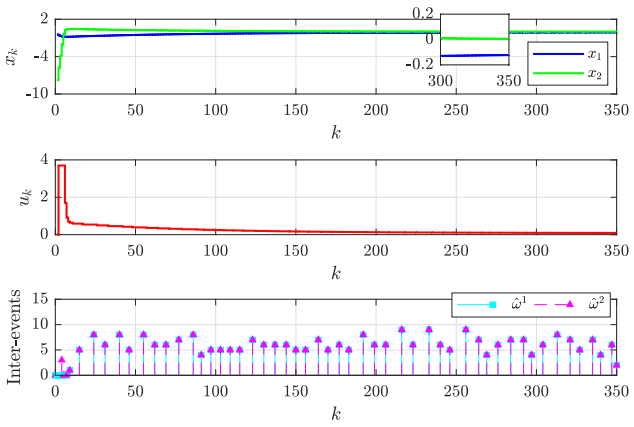

Fig. 3: The closed-loop system temporal response for $x(0) = \begin{bmatrix} -0.4274 & -7.7563 \end{bmatrix}^\top$.

in the first instants of simulation.

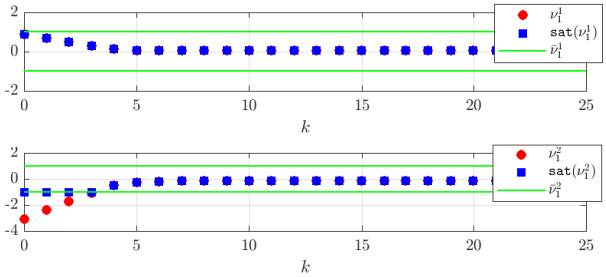

Fig. 4: Outputs of the first neuron of each layer.

VII. Conclusion

This work has studied the ETC problem of LTI systems stabilized by neural networks. We have proposed ETMs to reduce the computational cost associated with the neural network evaluation by updating only a portion of its layers. LMI-based conditions allowed to compute the triggering parameters and an inner approximation of the domain of attraction for the closed-loop system. Simulations tested the effectiveness of the ETMs, showing a significant reduction in the computational burden on the neural network. Future work could be devoted to exploring other types of event-triggering structures and also other abstractions for the nonlinear functions.

Acknowledgment

This work has been supported by ANR, project HANDY 18-CE40-0010.

References

[1] W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson, "Verification for machine learning, autonomy, and neural networks survey," *arXiv preprint arXiv:1810.01989*, 2018.

[2] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[3] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Transactions on Automatic Control*, 2020.

[4] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, 2021.

[5] P. Pauli, J. Köhler, J. Berberich, A. Koch, and F. Allgöwer, "Offset-free setpoint tracking using neural network controllers," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 992–003.

[6] M. Jin and J. Lavaei, "Stability-certified reinforcement learning: A control-theoretic perspective," *IEEE Access*, vol. 8, pp. 229 086–229 100, 2020.

[7] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on automatic control*, vol. 58, no. 4, pp. 847–861, 2012.

[8] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[9] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, 2014.

[10] C. de Souza, S. Tarbouriech, V. J. S. Leite, and E. B. Castelan, "Co-design of an event-triggered dynamic output feedback controller for discrete-time LPV systems with constraints," *Journal of The Franklin Institute*, 2020, Submitted.

[11] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 497–509, 2015.

[12] K. G. Vamvoudakis, "Event-triggered optimal adaptive control algorithm for continuous-time nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 3, pp. 282–293, 2014.

[13] Y. Gao, X. Guo, R. Yao, W. Zhou, and C. Cattani, "Stability analysis of neural network controller based on event triggering," *Journal of the Franklin Institute*, vol. 357, no. 14, pp. 9960–9975, 2020.

[14] C. De Souza, A. Girard, and S. Tarbouriech, "Event-triggered neural network control using quadratic constraints for perturbed systems," *Automatica*, 2022, Submitted.

[15] S. Tarbouriech, G. Garcia, J. M. Gomes da Silva Jr., and I. Queinnec, *Stability And Stabilization Of Linear Systems With Saturating Actuators*. Springer, 2011.

[16] L. Zaccarian and A. R. Teel, *Modern anti-windup synthesis: control augmentation for actuator saturation*. Princeton University Press, 2011, vol. 36.

[17] D. Peaucelle, H. Didier, Y. Labit, and K. Taitz, "User's guide for SEDUMI INTERFACE 1.04," 2002.

[18] H. Yin, P. Seiler, M. Jin, and M. Arcak, "Imitation learning with stability and safety guarantees," *IEEE Control Systems Letters*, vol. 6, pp. 409–414, 2021.