IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 3, Ser. IV (May - June 2020), PP 42-52
www.iosrjournals.org

Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

Nguyen Thi Oanh, Mai Thi Kim Anh
Thai Nguyen University of Information and Communication Technology

Abstract: *This paper presents a method to design adaptive controller for robot based on Lyapunov control function using Backstepping techniques combined sliding mode control and neural network. This study used the neural network to approximate the uncertainty functions, the weight coefficients of the neural network are trained online. The simulation results of the controller on the 2 degrees of freedom robotis is the sustainable control systems with sticking to the trajectory with a zero attachment error, that showed the correctness of the theoretical analysis and the applicability of the adaptive controller using Backstepping techniques and sliding mode control.*

Keywords*: Adaptive Control; Robot Manipulato; Backstepping Technique; Sliding Mode Control (SMC); Adaptive Neural Network Control (ANNC).*

Date of Submission: 06-06-2020 Date of Acceptance: 22-06-2020

I. Introduction

The design of the global asymptotic stabilizer controller by Backstepping technique based on Lyapunov function applied in nonlinear dynamic system is being studied and there are some published works such as [1, 2, 3 , 4, 5, 6, 7, 8]. To further develop this idea, we present the method of designing adaptive controllers using an online training 3-layer neural network to approximate the uncertainty functions of the object, Lyapunov control function associates. matched with the sliding controller to resist interference, ensure the closed system is globally stable and the deviation reaches zero with good quality.

To achieve the desired, in this study, the authors focused on designing the controller including 2 control stages based on Lyapunov control function and sliding control (SMC):

1. Design of a triple-layer noron network controller (MNN)
2. Design of sliding controller
3. Stability analysis

Applying the controller simulation in this research application for 2 degrees of freedom robot.

II. Controller Base Of Design Controller

2.1. Control problem

Considering the object of retrograde transmission, it is expressed in a general form:

$$\begin{cases} \dot{x}_i = f_i\left(x_{1,}x_2,\ldots,x_i\right) + g_i\left(x_1,x_2,\ldots,x_i\right)x_{i+1} \\ \dot{x}_n = f\left(x\right) + g\left(x\right)u + T_d \\ y = x_1 \end{cases} \quad (1)$$

$i = 1, 2, \ldots, n-1$

In which: $x_i = (x_1, x_2, \ldots, x_i)^T$ and state vector with i elements; $x = [x_1 \;\; \ldots \;\; x_n]^T$ The vector state of the system and T_d is noise. Assume that state variables and noise are both bounded and f (x) and g (x) are arbitrary smooth functions.

The design task was set out to find the controller for the object (1) to ensure the global tightness, stability, noise resistance and zero-tolerance.

2.2. Basis of controller design

The idea of the method is to design the controller including 2 stages: $u_{NN}(t)$ is a control channel designed on the basis of Lyapunov and $u_{SMC}(t)$ is a sliding control channel (SMC) used to resist interference, the uncertainty functions of the object are approximated by a three-layer linear transmission neural network (MNN).

Diagram of a closed system control system is described as Figure 1 and the controller is the sum of two control signals:

$$u(t) = u_{SMC}(t) + u_{NN}(t) \qquad (2)$$

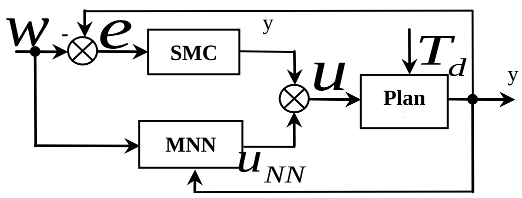

Figure 1. Controller structure diagram

2.2.1. Noron network controller design direct three-layer transmission

Considering the nonlinear object with the inverse propagation structure (1) ignoring the effects of noise we have:

$$\begin{cases} \dot{x}_i = f_i(x_1, x_2, \ldots, x_i) + g_i(x_1, x_2, \ldots, x_i)x_{i+1} \\ \dot{x}_n = f_n(x) + g_n(x)u \\ y = x_1 \end{cases} \qquad (3)$$

a) Approximating the function by artificial neural network

Hypothesis 1: $g_i(x_i)$ is a smooth function that defines a known property and a constant $g_{i0} > 0$ satisfy: $g_i(x_i) \geq |g_i(x_i)| \geq g_{i0}$ với $\forall x_i \in R^i$. The conditions to ensure that the (3) controllable system is $g_i(x_i) \neq 0$.

Hypothesis 2: the desired state vector x_{dj} with $j = 1, 2, \ldots, n+1$ is continuous and known in advance. $x_{dj} \in \Omega_{dj}$ and Ω_{dj} is the compact file.

Use 3-layer MNN to approximate the uncertainty function h (z): $h(z): R^m \to R$.

$$g_{nn}(z) = W^T S(V^T \boldsymbol{Z}) \qquad (4)$$

With $\boldsymbol{Z} = (Z^T, 1)^T$: input vector

$W = (w_1, w_2, \ldots w_l)^T \in R^l$ and $V = (v_1, v_2, \ldots v_l)^T \in R^{(m+1)l}$ is the weight matrix from grade 2 to grade out and from grade in to layer 2 of MNN. The number of neurons in a class is always satisfied $l > 1$. The output signal vector of layer 1 is as follows:

$S(V^T \boldsymbol{Z}) = (s(v_1^T \boldsymbol{Z}), s(v_2^T \boldsymbol{Z}), \ldots, s(v_{l-1}^T, \boldsymbol{Z}), 1)^T$ with $S(z_a) = \dfrac{1}{1 + e^{-\gamma z_a}}$; constant $\gamma > 0$

According to [7], MNN (4) satisfies the Stone-Weierstrass condition and can approximate any continuous function on a compact set with optional precision.

Approximate function: $h(Z) = W^{*T} S(V^{*T} \boldsymbol{Z}) + \mu \qquad (5)$

with: $\forall Z \in \Omega_Z \subset R^m$; μ is the approximation error of NN and Ω_Z is a compact file.

Hypothesis 3: For smooth functions h (Z) and approximations (4), there always exist ideal weights W^*, V^* để $|\mu| \leq \mu$ với $\mu > 0; \forall Z \in \Omega_Z$. That means the approximation error is always less than or equal to constant μ

Essentially *W** and *V** is unknown, these weights need to be estimated when designing the controller. Call $\hat{W}, \hat{V}$ are the estimated weights of *W** and *V** then the weight estimation error is determined as follows:

$\tilde{W} = \hat{W} - W^*$; $\tilde{V} = \hat{V} - V^*$

Lemma 1: The estimated MNN (4) can be expressed as follows:

$$\begin{aligned} &\hat{W}^T S(\hat{V}^T \boldsymbol{Z}) - W^{*T} S(V^{*T} \boldsymbol{Z}) \\ &= \tilde{W}^T (\hat{S} - \hat{S}' \hat{V}^T \boldsymbol{Z}) + \hat{W}^T \hat{S}' \tilde{V}^T \boldsymbol{Z} + d_u, \end{aligned} \qquad (6)$$

Here $\hat{S} = S(\hat{V}^T \boldsymbol{Z}), \hat{S}' = diag\{\hat{s}_1', \hat{s}_2', ..., \hat{s}_l',\}$ with $\hat{s}_i' = s'(\hat{v}_i^T \boldsymbol{Z}) = d[s(z_a)]/dz_a\big|_{z_a = \hat{v}_i^T Z}, i = 1, 2, ..., l,$

and residual limits d_u is blocked by $|d_u| \le \|V^*\|_F \left\|Z\hat{W}^T\hat{S}'\right\|_F + \|W^*\|\left\|\hat{S}'\hat{V}^T Z\right\| + |W^*|_1$ (7)

Conclusion [7]: Considering closed system including model (3) when $i=1$ and controller (9), if amplifier (10) Constant $\varepsilon_1 > 0$ NN weights updated by (11) with: $\Gamma_{w1} = \Gamma_{w1}^T > 0,$ $\Gamma_{v1} = \Gamma_{v1}^T > 0,$ and $\sigma_{w1}, \sigma_{v1} > 0$, top condition $x_1(0), \hat{W}_1(0)$ and $\hat{W}_1(0)$ is blocked, all signals in a closed system are blocked and vectors Z_1 exist in

$$\Omega_{z1} = \Big\{(x_1, y_d, \dot{y}_d)\big\| z_1(t)\big| \le \sqrt{2c_0 e^{-\lambda_1 t} + 2c_1/\lambda_1}, \boldsymbol{x}_{d2} \in \Omega_{d2}\Big\} \tag{8}$$

with c_0, c_1, λ_1 positive determination constant.

$$u_{NN_1} = \frac{1}{\boldsymbol{g}_1(x_1)}[-k_1(t)z_1 - \hat{W}_1^T S_1(\hat{V}_1^T \boldsymbol{Z}_1)] \tag{9}$$

$$k_1(t) = \frac{1}{\varepsilon_1}\Big(1 + \int_0^1 \theta \boldsymbol{g}_1(\theta z_1 + y_d)d\theta + \left\|\boldsymbol{Z}_1\hat{W}_1^T\hat{S}_1'\right\|_F^2 + \left\|\hat{S}_1'\hat{V}_1^T\boldsymbol{Z}_1\right\|^2\Big) \tag{10}$$

$$\begin{cases}\dot{\hat{W}}_1 = \Gamma_{w1}[(\hat{S}_1 - \hat{S}_1'\hat{V}_1^T\boldsymbol{Z}_1)z_1 - \sigma_{w1}\hat{W}_1] \\ \dot{\hat{V}} = \Gamma_{v1}[\boldsymbol{Z}_1\hat{W}_1^T\hat{S}_1' z_1 - \sigma_{v1}\hat{V}_1]\end{cases} \tag{11}$$

b) Design of adaptive controller by Backstepping technique

In the control design for backward nonlinear systems, the model determines the design of Lyapunov control function by backstepping technique to ensure that the global stable closed system is made easily. But when the object has non-functional components, this controller is no longer available. To achieve this, it is necessary to approximate the uncertain functions of the object. In this paper, we use three-layer linear neural network to approximate. Algorithm for designing adaptive controller by artificial neural network based on backstepping technique is done as follows:

Step 1: Consider (3) when giving $i=1$ is mean: $\dot{x}_1 = f_1(x_1) + g_1(x_1)x_2$. By watching x_2 As a virtual control input and control, we choose a new bias variable $z_2 = x_2 - \alpha_1$ với $\alpha_1 = u_{NN_1}$ defined in (9) and $\dot{z}_1 = f_1(x_1) + g_1(x_1)(z_2 + \alpha_2) - \dot{y}_d$

Choose $V_{z_1} = \int_0^{z_1} \sigma\beta_1(\sigma + y_d)d\sigma$ and transformed by [7], we get:

$$\dot{V}_{z1} = z_1\left[\boldsymbol{g}_1(x_1)(z_2 + \alpha_1) + h_1(Z_1)\right] \Rightarrow \dot{V}_{z1} = -k_1(t)z_1^2 - \psi_1 z_1 + z_1\boldsymbol{g}_1(x_1)z_2 \tag{12}$$

Step 2: Consider (3) and give $i=2$ I have: $\dot{x}_2 = f_2(\boldsymbol{x}_2) + g_2(\boldsymbol{x}_2)x_3$ (13)

Looking at x_3 as a virtual console, we can design a controller ando head α_2 for (13). Determined $z_3 = x_3 - \alpha_2$, we have:

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = f_2(\boldsymbol{x}_2) + g_2(\boldsymbol{x}_2)(z_3 + \alpha_2) - \dot{\alpha}_1 \tag{14}$$

Choosing: $V_{s2} = V_{z1} + \int_0^{z_2} \sigma\beta_2(\boldsymbol{x}_1, \sigma + \alpha_1)d\sigma > 0$

$$\Rightarrow \dot{V}_{s2} = \dot{V}_{z1} + z_2\beta_2(\boldsymbol{x}_2)\dot{z}_2 + \int_0^{z_2}\sigma\left[\frac{\partial\beta_2(\boldsymbol{x}_1, \sigma + \alpha_1)}{\partial\boldsymbol{x}_1}\dot{\boldsymbol{x}}_1 + \frac{\partial\beta_2(\boldsymbol{x}_1, \sigma + \alpha_1)}{\partial\alpha_1}\dot{\alpha}_1\right]d\sigma \tag{15}$$

Using (12), (14) we have:

(i)
$$\int_0^{z_2}\sigma\frac{\partial\beta_2(\boldsymbol{x}_1,\sigma+\alpha_1)}{\partial\alpha_1}\dot{\alpha}_1 d\sigma=\dot{\alpha}_1\int_0^{z_2}\sigma\frac{\partial\beta_2(\boldsymbol{x}_1,\sigma+\alpha_1)}{\partial\alpha_1}d\sigma$$
$$=\dot{\alpha}_1\left[z_2\beta_2(\boldsymbol{x}_2)-\int_0^{z_2}\beta_2(\boldsymbol{x}_1,\sigma+\alpha_1)d\sigma\right]$$

(ii)
$$\dot{\alpha}_1=\frac{\partial\alpha_1}{\partial\boldsymbol{x}_1}\dot{\boldsymbol{x}}_1+\omega_1,\omega_1=\frac{\partial\alpha_1}{\partial\boldsymbol{x}_{d2}}\dot{\boldsymbol{x}}_{d2}+\frac{\partial\alpha_1}{\partial\hat{W}_1}\dot{\hat{W}}_1+\sum_{\rho=1}^{l_1}\left[\frac{\partial\alpha_1}{\partial\hat{v}_{1\rho}}\dot{\hat{v}}_{1\rho}\right] \quad (16)$$

With $\dot{\hat{W}}_1$ and $\dot{\hat{v}}_{1\rho}$ is defined in (11), we have:

$$\begin{aligned}\dot{V}_{s2}&=-k_1(t)z_1^2-\psi_1 z_1+z_1\boldsymbol{g}_1(x_1)z_2\\&+z_2[\boldsymbol{g}_2(x_2)(z_3+\alpha_2)+h_2(Z_2)\end{aligned} \quad (17)$$

Here:

$$\begin{aligned}h_2(Z_2)&=\beta_2(\boldsymbol{x}_2)f_2(\boldsymbol{x}_2)+\frac{\dot{x}_1}{z_2}\int_0^{z_2}\sigma\frac{\partial\beta_2(\boldsymbol{x}_1,\sigma+\alpha_1)}{\partial\boldsymbol{x}_1}d\sigma\\&-\frac{\dot{\alpha}_1}{z_2}\int_0^{z_2}\beta_2(\boldsymbol{x}_1,\sigma+\alpha_1)d\sigma\\&=\beta_2(\boldsymbol{x}_2)f_2(\boldsymbol{x}_2)+\dot{x}_1 z_2\int_0^1\theta\frac{\partial\beta_2(\boldsymbol{x}_1,\theta z_2+\alpha_1)}{\partial\boldsymbol{x}_1}d\theta\\&-\dot{\alpha}_1\int_0^1\beta_2(\boldsymbol{x}_1,\theta z_2+\alpha_1)d\theta\end{aligned}$$

with: $Z_2=[\boldsymbol{x}_2^T,\alpha_1,\partial\alpha_1/\partial x_1,\omega_1]^T\in\Omega_{z2}\subset R^5$.

Select control function:

$$\alpha_2=\frac{1}{\boldsymbol{g}_2(x_2)}[-\boldsymbol{g}_1(x_1)z_1-k_2(t)z_2-\hat{W}_2^T S_2(\hat{V}_2^T\boldsymbol{Z}_2) \quad (18)$$

Here:

$$\begin{aligned}k_2(t)&=\frac{1}{\varepsilon_2}\Big(1+\int_0^1\theta\boldsymbol{g}_2(x_1,\theta z_2+\alpha_1)d\theta\\&+\left\|\boldsymbol{Z}_2^T\hat{W}_2^T\hat{S}_2^{'}\right\|_F^2+\left\|\hat{S}_2^{'}\hat{V}_2^T\boldsymbol{Z}_2^T\right\|^2\Big)\end{aligned} \quad (19)$$

With constants $\varepsilon_2>0$, and network weights updated by

$$\begin{cases}\dot{\hat{W}}_2=\Gamma_{w2}\left[(\hat{S}_2-\hat{S}_2^{'}\hat{V}_2^T\boldsymbol{Z}_2)z_2-\sigma_{w2}\hat{W}_2\right]\\\dot{\hat{V}}_2=\Gamma_{v2}\left[(\boldsymbol{Z}_2\hat{W}_2^T\hat{S}_2^{'}z_2-\sigma_{v2}\hat{V}_2\right]\end{cases} \quad (20)$$

With in: $\Gamma_{w2}=\Gamma_{w2}^T>0,\Gamma_{v2}=\Gamma_{v2}^T>0$ and $\sigma_{w2},\sigma_{v2}>0$.

Through some transformations we have the result:

$$\dot{V}_{s2}=-\sum_{j=1}^{2}\left[k_j(t)z_j^2+\psi_j z_j\right]+z_2\boldsymbol{g}_2(\boldsymbol{x}_2)z_3$$

Step k: The process is done the same for each step: k : $(3\le k\le n-1)$. Consider the system (3) when:

$$i=k;\ \dot{x}_k=f_k(\boldsymbol{x}_k)+g_k(\boldsymbol{x}_k)x_{k+1}$$

Choosing Lyapunov function:

$V_{sk}=V_{s(k-1)}+\int_0^{z_k}\sigma\beta_k(\boldsymbol{x}_{k-1},\sigma+\alpha_{k-1})d\sigma$. We can design control functions α_k and jurisprudence $\hat{W}_k$ and $\hat{V}_k$ have similar forms (18), (19), (20).

Step n: Consider $z_n=x_n-\alpha_{n-1}$ we have:

$$\dot{z}_n = \dot{x}_n - \dot{\alpha}_{n-1} = f_n(x) + g_n(x)u - \dot{\alpha}_{n-1}$$

Select the function Lyapunov:

$$V_{sn} = V_{s(n-1)} + \int_0^{z_n} \sigma\beta_n(\boldsymbol{x}_{n-1}, \sigma + \alpha_{n-1})d\sigma \qquad (21)$$

$$\Rightarrow \dot{V}_{sn} = \dot{V}_{s(n-1)} + z_n\beta_n(\boldsymbol{x}_n)\dot{z}_n$$
$$+\int_0^{z_n} \sigma\left[\frac{\partial\beta_n(\boldsymbol{x}_{n-1}, \sigma+\alpha_{n-1})}{\partial x_{n-1}}\dot{\boldsymbol{x}}_{n-1}\right.$$
$$\left.+\frac{\partial\beta_n(\boldsymbol{x}_{n-1}, \sigma+\alpha_{n-1})}{\partial \boldsymbol{x}_{n-1}}\dot{\alpha}_{n-1}\right]d\sigma$$

Similarly we have:

$$\dot{\alpha}_{n-1} = \frac{\partial\alpha_{n-1}}{\partial\boldsymbol{x}_j}\dot{\boldsymbol{x}}_j + \omega_{n-1}$$
$$= \sum_{j=1}^{n-1}\left\{\frac{\partial\alpha_{n-1}}{\partial x_j}\left[f_j(\boldsymbol{x}_j) + g_j(\boldsymbol{x}_j)x_{j+1}\right]\right\} + \omega_{n-1}$$

Here

$$\omega_{n-1} = \sum_{j=1}^{n-1}\left(\frac{\partial\alpha_{n-1}}{\partial\boldsymbol{x}_{d(j+1)}}\dot{\boldsymbol{x}}_{d(j+1)} + \frac{\partial\alpha_{n-1}}{\partial\hat{\mathrm{W}}_j}\dot{\hat{\mathrm{W}}}_j\right.$$
$$\left.+\sum_{\rho=1}^{l_j}\frac{\partial\alpha_{n-1}}{\partial\hat{v}_{j,\rho}}\dot{\hat{v}}_{j,\rho}\right)$$

with $\dot{\hat{\mathrm{W}}}_j$ and $\dot{\hat{v}}_{j,\rho}$ for $j = 1, 2, ..., n-1$ Designed in the previous steps $n-1$ and we have:

$$\dot{V}_{sn} = -\sum_{j=1}^{n-1}\left[k_j(t)z_j^2 + \psi_j z_j\right] + z_{n-1}\boldsymbol{g}_{n-1}(x_{n-1})z_n$$
$$+z_n\left[\boldsymbol{g}_n(x)u_{NN} + h_n(Z_n)\right] \qquad (22)$$

Here:

$$h_n(Z_n) = \beta_n(x_n)f_n(x_n) + z_n\int_0^1 \theta\frac{\partial\beta_n(x_{n-1}, \theta z_n + \alpha_{n-1})}{\partial x_{n-1}}\dot{x}_{n-1}d\theta$$
$$-\dot{\alpha}_{n-1}\int_0^1 \beta_n(x_{n-1}, \theta z_n + \alpha_{n-1})d\theta$$
$$Z_n = \left[x_n^T, \alpha_{n-1}\frac{\partial\alpha_{n-1}}{\partial x_1}, \frac{\partial\alpha_{n-1}}{\partial x_2}, ..., \frac{\partial\alpha_{n-1}}{\partial x_{n-1}}, \omega_{n-1}\right]^T \in \Omega_{zn} \subset R^{2n+1}$$

The controller is selected as follows:

$$u_{NN} = \frac{1}{\boldsymbol{g}_n(x)}\left[-\boldsymbol{g}_{n-1}(x_{n-1})z_{n-1} - k_n(t)z_n - \hat{W}_n^T S_n(\hat{V}_{\boldsymbol{n}}^{\boldsymbol{T}}\boldsymbol{Z}_n)\right] \qquad (23)$$

with:

$$k_n(t) = \frac{1}{\varepsilon_n}\left(1 + \int_0^1 \theta g_n(x_{n-1}, \theta z_n + \alpha_{n-1})d\theta\right.$$
$$\left.+\left\|\boldsymbol{Z}_n\hat{W}_n^T\hat{S}_n^{'}\right\|_F^2 + \left\|\hat{S}_n^{'}\hat{V}_n^T\boldsymbol{Z}_n\right\|^2\right) \qquad (24)$$

Select constants $\varepsilon_n > 0$ and the jurisprudence of the neural network

$$\begin{cases}\dot{\hat{w}}_n = \Gamma_{\backslash} \ (25) \quad \hat{S}_n^{'}\hat{V}_n^T\boldsymbol{Z}_n)z_n - \sigma_{wn}\hat{w}_n\big] \\ \dot{\hat{V}}_n = \Gamma_v \ (26) \quad \hat{S}_n^{'}z_n - \sigma_{vn}\hat{V}_n\big]\end{cases}$$

with: $\Gamma_{wn} = \Gamma_{wn}^T > 0, \Gamma_{vn} = \Gamma_{vn}^T > 0$ and $\sigma_{wn}, \sigma_{vn} > 0$

2.2.2. Design sliding controller

Consider (1) the slider controller design task is to give (T_d) noise resistance $e \to 0$. Definition of a sliding surface:

$$S(e) = e + a_1 \frac{de}{dt} + \cdots + a_{n-2} \frac{d^{n-2}e}{dt^{n-2}} + a_{n-1} \frac{d^{n-1}e}{dt^{n-1}} \quad (27) \qquad \text{with } S(e=0)=0.$$

To make sure $\lim_{t\to\infty} e(t) = 0$ then we have to choose coefficients a_i of polynomial characteristics

$A(s) = 1 + a_1 s + \cdots a_{n-2} s^{n-2} + a_{n-1} s^{n-1}$ (28) such that (28) is a Hurwitz polynomial. With the slip surface (27), the design task is to identify a control signal u_{SMC} so that when there is interference, the system will leave the sliding surface, this control signal will pull the system back to the sliding surface. The design of the slider control signal [8] is based on the Lyapunov function:

$$V_{SMC}(S) = \frac{1}{2} S^2 \quad (29)$$

And the sliding control signal is determined from the condition

$$\frac{dV_{SMC}(S)}{dt} = S\dot{S} = -KSsgn(S) < 0; K > 0 \quad (30)$$

The condition (30) is called the sliding condition.

From

$$\dot{S}(e) = \dot{e} + a_1 \ddot{e} + ... + a_{n-2} e^{(n-1)} + a_{n-1}\left(r^{(n)} - \left(f(x) + g(x)u_{SMC}\right)\right) = -K\,\text{sgn}(S)$$

I have:

$$u_{SMC} = \frac{K\,\text{sgn}(S) + D(\dot{e},...,e^{(n-2)}) - a_{n-1} f(x)}{a_{n-1} g(x)} \quad (31)$$

With: $$D(\dot{e},...,e^{(n-1)}) = \dot{e} + a_1 \ddot{e} + ... + a_{n-2} e^{(n-1)} + a_{n-1} w^{(n)}$$

2.2.3. Stability analysis

Theorem: *Reverse propagation system (1) provided that the observed state variables are directly and blocked, indefinite functions* $f(x)$ *and smooth uncertainty functions* $g(x)$ *are blocked, the controller (2) with components* u_{NN} *is defined (23) and* u_{SMC} *defined in (31) to ensure that the closed system is globally stable and has an adherence error of 0.*

Prove

Select function: $V = V_{NN} + V_{SMC}$

For V to be Lyapunov, the function and function must be Lyapunov.

We see after each design step a positive determination function appears:

$$V_{zi} = \int_0^{z_i} \sigma\beta_i(x_{i-1}, \sigma + \alpha_{i-1}) d\sigma,\ i = 2,3,...,n \quad (32)$$

So we choose it as Lyapunov function and this is an important key point of the method.

According to Hypothesis 1, we know that:

$$1 \le \beta_i(x_{i-1}, \sigma + \alpha_{i-1}) \le \boldsymbol{g}_1(x_{i-1}, \sigma + \alpha_{i-1}) / g_{i0}$$

and the following properties:

$$\text{(i)}\ V_{zi} = z_i^2 \int_0^1 \theta\beta_i(\boldsymbol{x}_{i-1}, \theta z_i + \alpha_{i-1}) d\theta \ge z_i^2 \int_0^1 \theta d\theta = \frac{z_i^2}{2} \quad (33)$$

$$V_{zi} = z_i^2 \int_0^1 \theta\beta_i(\boldsymbol{x}_{i-1}, \theta z_i + \alpha_{i-1}) d\theta$$

$$\text{(ii)}\ \le \frac{z_i^2}{g_{i0}} \int_0^1 \theta \boldsymbol{g}_i(\boldsymbol{x}_{i-1}, \theta z_i + \alpha_{i-1}) d\theta \quad (34)$$

Theorem 2 [7]: Consider a closed system including a tight backward transmission system (3) satisfying the assumption 1, the controller (23) and the law of updating the weight of NN (25). For the initial condition is blocked.

(i) all signals in the closed loop system are blocked, and vectors Z_j exist in compact files

$$\Omega_{zj} = \left\{ Z_j \left| \sum_{i=1}^{n} z_i^2(t) \le C_0, \sum_{i=1}^{n} \left\| \tilde{W}_i \right\|^2 \le \frac{C_0}{\lambda_{\min}(\Gamma_w^{-1})} \right. \right. \quad (35)$$
$$\left. \sum_{i=1}^{n} \left\| \tilde{V}_i \right\|_F^2 \le \frac{C_0}{\lambda_{\min}(\Gamma_v^{-1})}; x_{d(j+1)} \in \Omega_{d(j+1)} \right\}$$

with the constant C_0 and
(ii) inequality

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t z_i^2(\tau) d\tau \le \frac{2\varepsilon_j}{1+g_{j0}} \sum_{i=1}^{n} c_i \quad (36)$$

$$\sum_{i=1}^{n} z_i^2(t) \le 2V_s(0)e^{-\lambda_s t} + \frac{2}{\lambda_s} \sum_{i=1}^{n} c_i,\ \forall t \ge 0 \quad (37)$$

with $c_i, V_s(0), \lambda_s$ are positive constants.
Selected function Lyapunov:

$$V_{NN} = V_s = V_{sn} + \frac{1}{2} \sum_{j=1}^{n} \left[\tilde{W}_j^T \Gamma_w^{-1} \tilde{W}_j + tr\left\{ \tilde{V}_j^T \Gamma_w^{-1} \tilde{V}_j \right\} \right]$$

The final derivative and transformation we have

$$\dot{V}_s \le -\lambda_s V_s + \sum_{j=1}^{n} c_j$$

In that place:

$$c_j = \varepsilon_j \left(\frac{1}{4} \left\| W_j^* \right\|^2 + \frac{1}{4} \left\| V_j^* \right\|_F^2 + \left| W_j^* \right|_1^2 + \boldsymbol{\mu}_j^2 \right)$$
$$+ \frac{\sigma_w^2}{2} \left\| W_j^* \right\|^2 + \frac{\sigma_v^2}{2} \left\| V_j^* \right\|_F^2$$
$$\lambda_s = \min \left\{ g_{10} / \varepsilon_1, g_{20} / \varepsilon_2, ..., g_{n0} / \varepsilon_n, \right.$$
$$\left. \sigma_w / \lambda_{\max}(\Gamma_w^{-1}), \sigma_v / \lambda_{\max}(\Gamma_v^{-1}) \right\}$$

and I have

$$\begin{cases} V_s(t) \le V_s(0)e^{-\lambda_s t} + \dfrac{1}{\lambda_s} \displaystyle\sum_{j=1}^{n} c_j; \forall t \ge 0 \\ V_s(t) \ge V_{sn} = \displaystyle\sum_{j=1}^{n} V_{zj} \ge \dfrac{1}{2} \displaystyle\sum_{j=1}^{n} z_j^2(t) \end{cases} \quad (38)$$

This confirms for the initial condition is blocked, all signals z_i, $\hat{W}_i$ and $\hat{V}_i$, of a closed system and a set exists Ω_{zi} like that $Z_i \in \Omega_{zi}$ with every moment.
Replace the controller (23) and (22) with some of the last transforms we have:

$$\dot{V}_{sn} = -\sum_{j=1}^{n} \left[k_j(t) z_j^2 + \psi_j z_j \right]$$

Slide control

Select function $V_{SMC} = \frac{1}{2} S^2 > 0$ (39)

Derivative (38) from time to time is obtained $\frac{dV_{SMC}}{dt} = S.\dot{S}$

with:

$$\dot{S}(e) = -Ksgn(S) \Rightarrow \dot{V}_{SMC} = -KS\,\text{sgn}(S) \quad (40)$$

So $K > 0$ is $\dot{V}_{SMC} = -KS\,\text{sgn}(S) < 0$

If you choose larger, the sliding speed on the slip surface of the deviation e more faster.

Result: V_{NN} and function V_{SMC} are Lyapunov functions so that V is a Lyapunov function.

$$V = V_{NN} + V_{SMC} > 0 \Rightarrow \left\{ \dot{V} = \dot{V}_{NN} + \dot{V}_{SMC} \right.$$

$\dot{V}_{SMC} < 0$; $\dot{V}_{NN} \le 0$ and condition (38) so the closed system is globally stable and the error adheres to zero.

III. Application Of Controller For Robot

In this section, the authors will apply the controller and simulate the applicability of the ando controller to simulate verification on n-degree robots.

3.1. N-level robotic mathematical model

The dynamic equation of the robot of degrees n degrees of freedom is expressed as follows:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_d\dot{q} + F_r(\dot{q}) + T_d = M \quad (41)$$

Set: $A(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F_d\dot{q} + F_r(\dot{q}) + T_d$

So (41) can rewrite:

$$M(t) = H(q)\ddot{q} + A(q,\dot{q}) \quad (42)$$

with: $H(q) = H_k(q) + H_u(q)$; $A(q,\dot{q}) = A_k(q,\dot{q}) + A_u(q,\dot{q})$

$H_k(q); A_k(q,\dot{q}) = C_k(q,\dot{q})\dot{q} + G_k(q) + F_{dk}\dot{q}$ are known components.

$H_u(q); A_u(q,\dot{q}) = C_u(q,\dot{q})\dot{q} + G_u(q) + F_{du}\dot{q} + F_s(\dot{q}) + T_d$ are indeterminate components

Therein $q \in R^n$, $\dot{q}, \ddot{q}$ respectively, angle, speed and acceleration of the matching variables; $H(q) \in R^{n\times n}$ is an inertial, symmetric positive positive matrix; $C(q,\dot{q})\dot{q} \in R^n$ is a connected and radial moment vector; $G(q) \in R^n$ is the gravity vector; $F_d \in R^{n\times n}$ is a diagonal matrix of viscous friction coefficient; $F_r(\dot{q}) \in R^n$ is the dry friction coefficient; $T_d \in R^n$ is noise.

Set state variables:

$$\dot{X} = \left(\dot{X}_1, ..., \dot{X}_n\right)^T = \left(\dot{x}_{11}, \dot{x}_{12}, ..., \dot{x}_{n1}, \dot{x}_{n2}\right)^T = \left(\dot{q}_{11}, \dot{q}_{12}, ..., \dot{q}_{n1}, \dot{q}_{n2}\right)^T$$

$M = \left(\tau_1, \tau_2, ..., \tau_n\right)$; $\underline{u} = \left(u_{11}, u_{12}, ..., u_{n1}, u_{n2}\right)$

From (42) I have: $\ddot{q} = H^{-1}\left(M - A(q,\dot{q})\right)$ (43)

If we consider cross-linking as uncertainty, we have a general model as follows:

$$\dot{X} = f_{ij}(\underline{x}) + g_{ij}(\underline{x})u_{ij};\ i = 1 \div n;\ j = 1 \div 2 \quad (44)$$

In the place: $f_{ij}(\underline{x}) = 0$ and $u_{ij} = x_{i2}$ at $j = 1, \forall i$; $g_{ij}(\underline{x}) = 1$, $\forall i, j$

$f_{ij}(\underline{x}) = H(q)^{-1}.C(q,\dot{q})\dot{q}$ and $\underline{u}_{ij} = H(q)^{-1}.M$ at $j = 2, \forall i$

So (44) is rewritten:

$$\dot{X} = \left(0, f_{12}(\underline{x}), 0, f_{22}(\underline{x}), ..., 0, f_{n2}(\underline{x})\right)^T + g_{ij}.\left(x_{12}, u_{12}, x_{22}, u_{22}, ..., x_{n2}, u_{n2}\right)^T \quad (45)$$

With functions $f_{i2}\left(\underline{x}_i, \underline{\theta}_i\right)$ and $g_{ij}\left(\underline{x}_i, \underline{\theta}_i\right)$;

$i = 1, 2, ..., n;\ j = 1, 2$ are indeterminate functions because they contain an infinite parameter vector of each match $\underline{\theta}_i$, We can show the model (45) through n tightly propagated models as follows:

$$\begin{cases} \dot{x}_{i1} = x_{i2} \\ \dot{\underline{x}}_{i2} = f(\underline{x}_i, \underline{\theta}_i) + g_{i2}(\underline{x}_i, \underline{\theta}_i)u_i + d(t) \end{cases} \quad (46)$$

$i = 1, 2, ..., n;\ g_{i1} = 0.$

The robot model presented in (46) allows the use of a controller design based on the Lyapunov control function adapted by Backstepping and artificial neural networks to approximate the uncertainty function associated with the controller. controls sliding to resist interference.

3.2 Verification simulation on 2 DOF robots

3.2.1. Simulation parameters

In this section, the study simulates the separate and concurrent effects of the noron network controller and the sliding control.

- Robot model: Consider Planar 2 DOF robot as Figure 2:

The change $\ddot{q}_1$ in tight backward transmission is as follows:

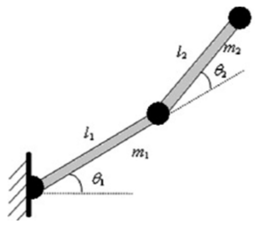

Figure 2. Flat robot structure 2 DOF

$$\begin{cases} \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = f_{12}(x_{11}, x_{12}) + g_{12}(x_{11}, x_{12}) u_{12} \\ q_1 = x_{11} \end{cases} \quad (47)$$

with: $\begin{cases} f_{12}(x_{11}, x_{12}) = k(-h_{22}a_1 + h_{12}a_2) \\ g_{12}(x_{11}, x_{12}) = 1 \\ u_{12} = k(h_{22}\tau_1 - h_{12}\tau_2) \end{cases}$

Similarly with $\ddot{q}_2$ I have:

$$\begin{cases} \dot{x}_{21} = x_{22} \\ \dot{x}_{22} = f_{22}(x_{21}, x_{22}) + g_{22}(x_{21}, x_{22}) u_{22} \\ q_2 = x_{21} \end{cases} \quad (48)$$

With: $\begin{cases} f_{22}(x_{21}, x_{22}) = k(h_{21}a_1 - h_{11}a_2) \\ g_{22}(x_{21}, x_{22}) = 1 \\ u_{22} = k(-h_{21}\tau_1 + h_{11}\tau_2) \end{cases}$

Stitch 1: weight $m_1 = 5kg$, length $l_1 = 0.45m$

Stitch 2: weight $m_2 = 3kg$, length $l_2 = 0.35m$

Two match variables: θ_1, θ_2

- *Select controller:*

$$u_{NN} = -z_1 - k_2(t) z_2 - W_2^T S_2(V_2^T \mathbf{Z}_2) \quad (49)$$

With $z_1 = x_1 - y_d$, $z_2 = x_2 - \alpha_1$ and $Z_2 \left[x_1, x_2, \alpha_1, \frac{\partial \alpha_1}{\partial \alpha_2}, \omega_1, 1 \right]^T$

with: $\alpha_1 = -k(t) z_1 - \hat{W}_1^T S_1(\hat{V}_1^T \mathbf{Z}_1)$; $Z_1 = [x_1, y_d, \dot{y}_d, 1]^T$

$$\omega_1 = \frac{\partial \alpha_1}{\partial y_d} \dot{y}_d + \frac{\partial \alpha_1}{\partial \dot{y}_d} \ddot{y}_d + \frac{\partial \alpha_1}{\partial \hat{W}_1} \dot{\hat{W}}_1 + \sum_{\rho=1}^{l_1} \left[\frac{\partial \alpha_1}{\partial \hat{v}_{1\rho}} \dot{\hat{v}}_{1\rho} \right]$$

$$k_j(t) = \frac{1}{\varepsilon_j} \left(\frac{3}{2} + \left\| \mathbf{Z}_j \hat{W}_j^T \hat{S}_j^{'} \right\|_F^2 + \left\| \hat{S}_j^{'} \hat{V}_j^T \mathbf{Z}_j \right\|^2 \right); j = 1, 2$$

the corresponding weights $\hat{W}_1$, $\hat{V}_1$, $\hat{W}_2$, $\hat{V}_2$ be updated according to the expression (11), (25).

Choose the coefficients:

$\gamma = 3.0; \varepsilon_1 = 1.0; \varepsilon_2 = 5.0$; $\sigma_{w_1} = \sigma_{w_2} = 1\times10^{-2}$; $\sigma_{v_1} = 1\times10^{-4}; \sigma_{v_2} = 1\times10^{-3}$ $\Gamma_{w_1} = \Gamma_{w_2} = diag\{1.0\}$; $\Gamma_{v_1} = \Gamma_{v_2} = diag\{10.0\}$

The initial weights: $\hat{W}_1(0) = 0.0,\ \hat{W}_2(0) = 0.0$, $\hat{V}_1(0)$, $\hat{V}_2(0)$; take randomized $u_{SMC} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ and

$$\begin{cases} u_1 = \text{sgn}(s_1(e_1)) + k_1\dot{w}_1 + \ddot{w}_1 - k_1x_{12} - f_{12} \\ u_2 = \text{sgn}(s_2(e_2)) + k_2\dot{w}_2 + \ddot{w}_2 - k_2x_{22} - f_{22} \end{cases} \quad (50)$$

3.2.2. Simulation results

Case 1: The effect of a NN controller without an SMC controller is shown in Figure 3:

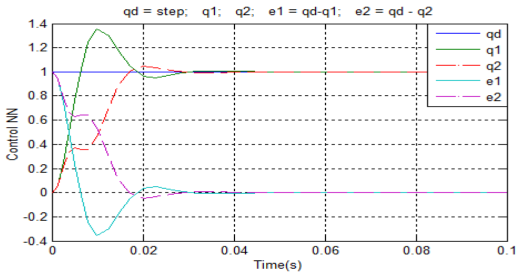

Figure 3. Response result when there is only NN controller (without SMC controller)

Case 2: The impact of a SMC controller without a NN controller is shown in Figure 4:

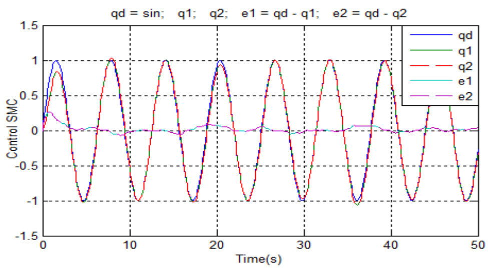

Figure 4. Response result when there is only SMC controller (without NN controller)

Case 3: The impact of both NN and SMC controllers is shown in Figure 5, Figure 6:

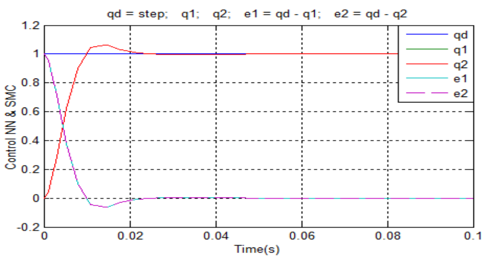

Figure 5. Response results when both NN controllers and SMC controllers are involved (qd = step)

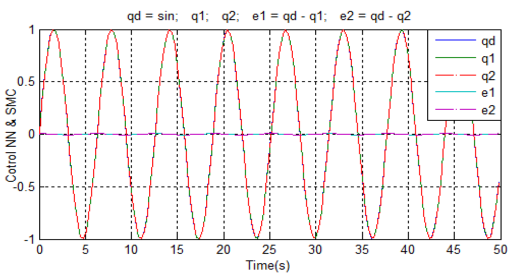

Figure 6. Response results when both NN controllers and SMC controllers are involved (qd = sin)

IV. Conclusion

The simulation results of the controller for 2 degrees of freedom robot show that: thanks to the ability to approximate the high-precision neural network function of the neural network, we do not need to analyze cross-relations between the joints as well as the change of inertia torque, friction force, ... but still ensure the exact trajectory set with good quality.

The simulation results confirm the applicability of the controller to the n-degree robot with an uncertainty model and the influence of noise to ensure a stable stable system, sticking to the trajectory set with a zero-tolerance. without the need to accurately analyze the cross-linking between joints, as well as other uncertainties of the robot such as load, friction ... It is also the advantage of the controller compared to the other adaptive sustainable controllers.

References

[1]. V.H.Dao, *Robot Technique*, Science and Technics Publishing House, 2017.

[2]. D.P.Nguyen, X.M.Phan, T.T.Han, *Nonlinear control theory*, Science and Technics Publishing House, 2019.

[3]. Miroslav Krstic, Loannis Kanellakopoulos, Petar Kokotovic, *Nonlinear and adaptive control design,* 2015.

[4]. S. Sam Ge, T.T.Han, "Semiglobal ISpS Disturbance Attenuation With Output Tracking via Direct Adaptive Design*". IEEE Trans. On NN*. Vol. 18, 2007.

[5]. Li, Zou, Zhao, "Adaptive NN control for a class of perturbed strict feedback nonlinear systems", *Americal control conference 2008*, vol.1, pp.81-86, 2008.

[6]. Gong, Yao,"Neural network adaptive robust control of nonlinear system in semi strict-feedback form", *Automatica 37*, pp.1149-1160, 2001.

[7]. T. Zhang, S. S. Ge*, C. C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design", 2000.

[8]. Wilfrid Perruetti, Jean Pierre Barbot, *Sliding Mode Control in Engineering*, 2002.

Nguyen Thi Oanh, et. al. "Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22(3), 2020, pp. 30-41.