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 robot 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 Manipulator; Backstepping Technique; Sliding Mode Control (SMC); Adaptive Neural Network Control (ANNC).

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:

\[
\begin{align*}
\dot{x}_i &= f_i(x_1, x_2, \ldots , x_n) + g_i(x_1, x_2, \ldots , x_n) \dot{x}_{i+1} \\
\dot{\dot{x}}_n &= f(x) + g(x) u + T_d \\
y &= x_1
\end{align*}
\]

(1)

In which, \(x_i = (x_1, x_2, \ldots , x_n)^T\) 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: \(u_{\text{NN}}(t)\) is a control channel designed on the basis of Lyapunov and \(u_{\text{SMC}}(t)\) 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:

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

**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:

\[
\begin{align*}
\dot{x}_i &= f_i(x_1, x_2, \ldots, x_n) + g_i(x_1, x_2, \ldots, x_n) x_{i+1} \\
\dot{x}_n &= f_n(x) + g_n(x) u \\
y &= x_i
\end{align*}
\]

(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: $g_i(x_i) \geq |g_i(x_i)| \geq g_{i0}$ for all $x_i \in \mathbb{R}$. The conditions to ensure that the (3) controllable system is $g_i(x_i) \neq 0$.

**Hypothesis 2:** the desired state vector $x_d$ with $j = 1, 2, \ldots, n + 1$ is continuous and known in advance. $x_d \in \Omega_d$ and $\Omega_d$ is the compact file.

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

$$g_{nn}(z) = W^TS(V^TZ) \quad (4)$$

With $Z = (Z^T, 1)^T$: input vector

$$W = (w_1, w_2, \ldots, w_l)^T \in \mathbb{R}^l$$
and $V = (v_1, v_2, \ldots, v_l)^T \in \mathbb{R}^{(m + 1)l}$ 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:

$$S(V^TZ) = (s(v_1^T Z), s(v_2^T Z), \ldots, s(v_l^T Z), 1)^T$$

with $S(z_{i0}) = \frac{1}{1 + e^{-\gamma z_i}}$; 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: $h(Z) = W^TS(V^TZ) + \mu \quad (5)$

with: $\forall Z \in \Omega_z \subseteq \mathbb{R}^m$; $\mu$ is the approximation error of NN and $\Omega_z$ is a compact file.

**Hypothesis 3:** For smooth functions $h(Z)$ and approximations (4), there always exist ideal weights $W, V$ and $d$ such that $|||l \leq \mu \leq \epsilon$ and $\forall Z \in \Omega_z$. That means the approximation error is always less than or equal to constant $\mu$.

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

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

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

$$\hat{W}^T S(V^T \hat{Z}) - W^T S(V^T Z) = \hat{W}^T (\hat{S} - S \hat{V}^T Z) + \hat{W}^T \hat{S} \hat{V}^T Z + d_z.$$  

(6)

Here $\hat{S} = S(V^T \hat{Z}), \hat{S} = diag\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_i\}$ with $\hat{s}_i = s(v_i^T Z) = \frac{d[s(z_{i0})]}{dz_{i0}} |_{z_{i0} = z_i}, i = 1, 2, \ldots, l$.
and residual limits $d_x$ is blocked by \[|d_x| \leq \|W^T\| \left\|Z \hat{W}^T \hat{S}\right\| + \|W^T\| \left\|\hat{S} \hat{W}^T Z + |W^T|\right\| \] (7)

Conclusion [7]: Considering closed system including model (3) when $i = 1$ and controller (9), if amplifier (10)

Constant $\epsilon_i > 0$, NN weights updated by (11) with: $\Gamma_w = \Gamma_{w1} > 0$,

$\Gamma_v = \Gamma_{v1} > 0$, and $\sigma_w, \sigma_v > 0$. top condition $x_i(0), \hat{W}_i(0)$ and $\hat{W}_i(0)$ is blocked, all signals in a closed system are blocked and vectors $Z_i$ exist in

\[
\Omega_{z_i} = \{ (x_i, y_d, \hat{y}_d) \left| \|z_i(t)\| \right. \}
\leq \sqrt{2c_0e^{-\lambda t} + 2c_1 / \lambda}, x_{d2} \in \Omega_{d2}\] (8)

with $c_0, c_1, \lambda$ positive determination constant.

\[
u_{NN}(t) = \frac{1}{g_i(x_i)}[-k_1(t)z_{i1} - \hat{W}_i^T S_1 V_i^T Z_i] \] (9)

\[
k_i(t) = \frac{1}{c_1}(1 + \int_0^1 \theta g_i(\theta z_1 + y_d)d\theta
\]

\[
\left\|Z_i \hat{W}_i^T \hat{S}_1 \right\| + \left\|\hat{S}_1 \hat{W}_i^T Z_i\right\|
\]

\[
\begin{align*}
\hat{W}_i &= \Gamma_w(\hat{S}_1 - \hat{S}_1^T Z_i)z_{i1} - \sigma_w \hat{W}_i
\end{align*}
\]

\[
\begin{align*}
\hat{v}_i &= \Gamma_v[Z_i \hat{W}_i^T \hat{S}_1 z_{i1} - \sigma_v \hat{v}_i]
\end{align*}
\]

(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: $\dot{x}_i = f_i(x_i) + g_i(x_i)x_{2i}$. By watching $x_2$ as a virtual control input and control, we choose a new bias variable $z_2 = x_2 - \alpha_1$ vài $\alpha_1 = u_{NN}$ defined in (9) and $\dot{z}_i = f_i(x_i) + g_i(x_i)(z_2 + \alpha_2) - \hat{y}_d$

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

\[
\hat{V}_{z_1} = z_1 \left[ g_i(x_i)(z_2 + \alpha_2) + h_i(Z_i) \right] \Rightarrow \hat{V}_{z_1} = \left(-k_1(t)z_{i1} - \psi_1 z_1 + z_1 g_i(x_i)z_2 \right)
\]

(12)

**Step 2:** Consider (3) and give $i = 2$ have: $\dot{x}_2 = f_2(x_2) + g_2(x_2)x_{32}$

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

\[
\dot{z}_3 = \dot{x}_3 - \alpha_2 = f_2(x_2) + g_2(x_2)(z_3 + \alpha_2) - \hat{a}_1
\]

Choosing: $V_{z_2} = V_{z_1} + \int_0^{\hat{y}_d} \beta_2(x_1, \sigma + \alpha_1)d\sigma > 0$

\[
\Rightarrow \hat{V}_{z_2} = \hat{V}_{z_1} + \beta_2(x_3, \sigma + \alpha_1) + \int_0^{\hat{y}_d} \left[ \frac{\partial \beta_2(x_1, \sigma + \alpha_1)}{\partial x_1} \dot{x}_1 + \frac{\partial \beta_2(x_1, \sigma + \alpha_1)}{\partial \alpha_1} \right] d\sigma
\]

(15)
Using (12), (14) we have:

\[
\int_{0}^{z} \frac{\partial \beta_{2}(x_{t}, \sigma + \alpha_{t})}{\partial \alpha_{t}} d\sigma = \hat{a}_{t} \int_{0}^{z} \sigma \frac{\partial \beta_{2}(x_{t}, \sigma + \alpha_{t})}{\partial \alpha_{t}} d\sigma \Rightarrow \hat{a}_{t} = \frac{\partial \alpha_{t}}{\partial x_{2}} \dot{x}_{t} + \sigma_{t} = \frac{\partial \alpha_{t}}{\partial x_{2}} \dot{x}_{d2}
\]

(i) \quad \Rightarrow \quad \dot{z}_{2} = \hat{a}_{t} \left[ z_{2} \beta_{2}(x_{2}) - \int_{0}^{z} \beta_{2}(x_{t}, \sigma + \alpha_{t}) d\sigma \right] \quad (ii) + \frac{\partial \alpha_{t}}{\partial \dot{\hat{w}}_{1}} \dot{\hat{w}}_{1} + \sum_{p=1}^{k} \frac{\partial \alpha_{t}}{\partial \dot{\hat{v}}_{1_{p}}} \dot{\hat{v}}_{1_{p}} \quad (16)

With \( \hat{W}_{1} \) and \( \dot{\hat{V}}_{1_{p}} \) is defined in (11), we have:

\[
\dot{\hat{V}}_{s2} = \dot{k}_{2}(t)z_{2}^{2} - \psi_{1}z_{2} + z_{2}[g_{2}(x_{2})(z_{2} + \alpha_{t})] + h_{2}(Z_{2})
\]

Here:

\[
h_{2}(Z_{2}) = \beta_{2}(x_{2})f_{2}(x_{2}) + \frac{\hat{a}_{t}}{z_{2}} \int_{0}^{z} \frac{\partial \beta_{2}(x_{t}, \sigma + \alpha_{t})}{\partial \alpha_{t}} d\sigma
\]

\[
= \beta_{2}(x_{2})f_{2}(x_{2}) + \hat{a}_{t} \int_{0}^{z} \beta_{2}(x_{t}, \sigma + \alpha_{t}) d\sigma
\]

\[
= \beta_{2}(x_{2})f_{2}(x_{2}) + \hat{a}_{t} \int_{0}^{z} \beta_{2}(x_{t}, \sigma + \alpha_{t}) d\sigma
\]

\[
= \beta_{2}(x_{2})f_{2}(x_{2}) + \hat{a}_{t} \int_{0}^{z} \beta_{2}(x_{t}, \sigma + \alpha_{t}) d\sigma
\]

\[
\frac{\partial \alpha_{t}}{\partial \dot{\hat{w}}_{1}} \dot{\hat{w}}_{1} + \sum_{p=1}^{k} \frac{\partial \alpha_{t}}{\partial \dot{\hat{v}}_{1_{p}}} \dot{\hat{v}}_{1_{p}} \quad (19)
\]

with: \( Z_{2} = [x_{2}^{T}, \alpha_{t}, \sigma_{t}, \sigma_{t} / \partial \alpha_{t}, \sigma_{t}]^{T} \in \Omega_{z2} \subset R^{5} \).

Select control function:

\[
a_{2} = \frac{1}{g_{2}(x_{2})} [ - g_{2}(x_{2})z_{2} - \dot{k}_{2}(t)z_{2} - \dot{\hat{w}}_{2}^{T} \hat{S}_{2}[\hat{V}_{2}^{T}]Z_{2} ]
\]

Here:

\[
k_{2}(t) = \frac{1}{e_{2}} \left( 1 + \int_{0}^{t} \theta g_{2}(x_{2}, \theta z_{2} + \alpha_{t}) d\theta \right)
\]

(19)

(20)

With constants \( e_{2} > 0 \), and network weights updated by

\[
\dot{\hat{w}}_{2} = \Gamma_{w2} \left( \hat{S}_{2} - \hat{S}_{2}^{T} \hat{S}_{2} \right) z_{2} - \sigma_{w2} \hat{w}_{2}
\]

\[
\dot{\hat{v}}_{2} = \Gamma_{v2} \left( Z_{2}^{T} \hat{S}_{2} \hat{V}_{2} - \sigma_{v2} \hat{V}_{2} \right)
\]

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

\[
i = k; \quad \dot{x}_{k} = f_{k}(x_{k}) + g_{k}(x_{k})x_{k+1}
\]

Choosing Lyapunov function:

\[
\dot{V}_{k} = V_{s(k-1)} + \int_{0}^{z} \sigma \beta_{k}(x_{k}, \sigma + \alpha_{k-1}) d\sigma \quad (20)
\]

We can design control functions \( \alpha_{k} \) and jurisprudence \( \hat{W}_{k} \) and \( \hat{V}_{k} \) have similar forms (18), (19), (20).

Step n: Consider \( x_{n} = x_{n} - \alpha_{n-1} \) we have:
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

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

Select the function Lyapunov:

\[ V_{sn} = V_{s(n-1)} + \int_0^1 \sigma \beta_n(x_{n-1}, \sigma + \alpha_{n-1}) d\sigma \]

\[ \Rightarrow \dot{V}_{sn} = \dot{V}_{s(n-1)} + z_n \beta_n(x_n) \dot{z}_n \]

\[ + \int_0^1 \sigma \left( \frac{\partial \beta_n(x_{n-1}, \sigma + \alpha_{n-1})}{\partial x_{n-1}} \right) \dot{x}_{n-1} \]

\[ + \frac{\partial \beta_n(x_{n-1}, \sigma + \alpha_{n-1})}{\partial x_{n-1}} \dot{\alpha}_{n-1} d\sigma \]

Similarly we have:

\[ \dot{\alpha}_{n-1} = \frac{\partial \alpha_{n-1}}{\partial x_j} \dot{x}_j + \omega_{n-1} \]

\[ = \sum_{j=1}^{n-1} \left( \frac{\partial \alpha_{n-1}}{\partial x_j} \dot{x}_{j+1} + \frac{\partial \alpha_{n-1}}{\partial x_j} \dot{W}_j \right) \]

Here

\[ \omega_{n-1} = \sum_{j=1}^{n-1} \left( \frac{\partial \alpha_{n-1}}{\partial x_j} \dot{x}_{j+1} + \frac{\partial \alpha_{n-1}}{\partial x_j} \dot{W}_j \right) \]

with \( \dot{W}_j \) and \( \dot{\alpha}_{j,\rho} \) for \( j = 1, 2, ..., n-1 \) Designed in the previous steps \( n-1 \) and we have:

\[ \dot{V}_{sn} = \sum_{j=1}^{n-1} \left[ \dot{\alpha}_{j-1} \dot{x}_{j+1} + \dot{\alpha}_{j,\rho} \dot{W}_j \right] + z_n \beta_n(x_{n-1}) \dot{z}_n \]

(22)

Here:

\[ h_n(Z_n) = \beta_n(x_n) \dot{x}_n + z_n \left( \int_0^1 \sigma \beta_n(x_{n-1}, \sigma + \alpha_{n-1}) d\sigma \right) \dot{x}_{n-1} d\theta \]

\[ - \dot{\alpha}_{n-1} \int_0^1 \beta_n(x_{n-1}, \sigma + \alpha_{n-1}) d\theta \]

\[ Z_n = \left[ x_1', \alpha_{n-1}, \frac{\partial \alpha_{n-1}}{\partial x_1}, ..., \frac{\partial \alpha_{n-1}}{\partial x_{n-1}} \right] \in \Omega \subset R^{2n+1} \]

The controller is selected as follows:

\[ u_{sn} = \frac{1}{g_n(x)} \left[ -g_{n+1}(x_{n+1}) \dot{z}_{n+1} - k_n(t) \dot{z}_n - \dot{W}_n S_n V_n^T Z_n \right] \]

(23)

with:

\[ k_n(t) = \frac{1}{\epsilon_n} \left( 1 + \int_0^1 \sigma g_n(x_{n-1}, \sigma + \alpha_{n-1}) d\sigma \right) \]

\[ + \left\| Z_n \dot{W}_n S_n V_n^T Z_n \right\| + \left\| \dot{S}_n V_n^T Z_n \right\| \]

(24)

Select constants \( \epsilon_n > 0 \) and the jurisprudence of the neural network

\[ \dot{\hat{V}}_n = \Gamma_n \left( 2 \dot{\hat{S}}_n V_n^T Z_n \right) \]

(25)

\[ \dot{\hat{V}}_n = \Gamma_n \left( \dot{\hat{S}}_n z_n - \sigma_{wn} \hat{V}_n \right) \]

(26)
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

with: \( \Gamma_{wn} = \Gamma_{wn}^T > 0, \Gamma_{wn} = \Gamma_{wn}^T > 0 \) and \( \sigma_{wn}, \sigma_{wn} > 0 \)

2.2.2. Design sliding controller

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

\[
S(e) = e + a_1 \frac{de}{dt} + \cdots + a_{n-2} \frac{d^{n-2}e}{dt^{n-2}} + a_{n-1} \frac{d^{n-1}e}{dt^{n-1}} \quad (27)
\]

With \( S(e) = 0 \).

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

\[
A(s) = 1 + a_1 s + \cdots + a_{n-2} s^{n-2} + a_{n-1} s^{n-1} \quad (28)
\]

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

\[
\frac{dV_{SMC}(S)}{dt} = SS = -K \text{sgn}(S) < 0; K > 0 \quad (30)
\]

The condition (30) is called the sliding condition.

\[
\dot{S}(e) = \dot{e} + a_1 \dot{e} + \cdots + a_{n-2} e^{(n-2)} + a_{n-1} e^{(n-1)}
\]

From

\[
+a_{n-1} \left( f^{(n)}(x) + g(x) u_{SMC} \right) = -K \text{sgn}(S)
\]

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:

\[
D(\dot{e}, ..., e^{(n-1)}) = \dot{e} + a_1 \dot{e} + \cdots + a_{n-2} e^{(n-2)} + a_{n-1} w^{(n)}
\]

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.

Proof

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:

\[
V_{\sigma} = \int_{0}^{1} \sigma \beta (x_{i-1}, \sigma + \alpha_{i-1}) d\sigma, i = 2, 3, ..., n \quad (32)
\]

So we choose it as Lyapunov function and this is an important key point of the method. According to Hypothesis 1, we know that:

\[
1 \leq \beta (x_{i-1}, \sigma + \alpha_{i-1}) \leq g_i (x_{i-1}, \sigma + \alpha_{i-1}) / g_{i0}
\]

and the following properties:

(i) \( V_{\sigma} = \frac{z^2}{2} \int_{0}^{r} \theta \beta (x_{i-1}, \theta z_i + \alpha_{i-1}) d\theta \geq \frac{z^2}{2} \int_{0}^{r} \theta d\theta = \frac{z^2}{2} \quad (33) \)

(ii) \( V_{\sigma} = \frac{z^2}{2} \int_{0}^{r} \theta g_i (x_{i-1}, \theta z_i + \alpha_{i-1}) d\theta \quad (34) \)

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

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

\[
\Omega_j = \left\{ Z_j \left| \sum_{i=1}^{n} z_i^2(t) \leq C_0, \sum_{i=1}^{n} \| \tilde{W}_j \|_F^2 \leq \frac{C_0}{\lambda_{\min}(\Gamma_w)} \right. \right\} \quad (35)
\]

with the constant \( C_0 \) and (ii) inequality

\[
\lim_{t \to \infty} \int_{0}^{t} z_i^2(\tau) d\tau \leq \frac{2\epsilon_j}{1+g_{j0}} \sum_{i=1}^{n} c_i \quad (36)
\]

\[
\sum_{i=1}^{n} z_i^2(t) \leq 2V_x(0)e^{-\lambda_t} + \frac{2}{\lambda_s} \sum_{i=1}^{n} c_i, \forall t \geq 0 \quad (37)
\]

with \( c_i, V_x(0), \lambda_s \) are positive constants.

Selected function Lyapunov:

\[
V_{AV} = V_x = V_m + \frac{1}{2} \sum_{j=1}^{n} \left[ \tilde{W}_j^T \Gamma_w^{-1} \tilde{W}_j + tr \left\{ \tilde{V}_j^T \Gamma_w^{-1} \tilde{V}_j \right\} \right]
\]

The final derivative and transformation we have

\[
\dot{V}_x \leq -\lambda_s V_x + \sum_{j=1}^{n} c_j
\]

In that place:

\[
c_j = \epsilon_j \left( \frac{1}{4} \| \tilde{W}_j \|_F^2 + \frac{1}{4} \| \tilde{V}_j \|_F^2 + \| \tilde{W}_j \|_F^2 + \mu_j^2 \right) + \frac{\sigma_u^2}{2} \| \tilde{W}_j \|_F^2 + \frac{\sigma_v^2}{2} \| \tilde{V}_j \|_F^2
\]

\[
\lambda_s = \min \left\{ \frac{g_{j0}}{\epsilon_1}, \frac{g_{j0}}{\epsilon_2}, ..., \frac{g_{j0}}{\epsilon_n}, \sigma_u \lambda_{\max}(\Gamma_w^{-1}), \sigma_v \lambda_{\max}(\Gamma_w^{-1}) \right\}
\]

and I have

\[
\begin{cases}
V_x(t) \leq V_x(0)e^{-\lambda_t} + \frac{1}{\lambda_s} \sum_{i=1}^{n} c_i, \forall t \geq 0 \\
V_x(t) \geq V_m = \sum_{j=1}^{n} V_j \geq \frac{1}{2} \sum_{j=1}^{n} z_j^2(t) \quad (38)
\end{cases}
\]

This confirms for the initial condition is blocked, all signals \( z_i, \tilde{W}_i \) and \( \tilde{V}_i \) of a closed system and a set exists \( \Omega_{zi} \) like that \( Z_i \in \Omega_{zi} \) with every moment.

Replace the controller (23) and (22) with some of the last transforms we have:

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

**Slide control**

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

Derivative (38) from time to time is obtained

\[
\frac{dV_{SMC}}{dt} = S \dot{S}
\]
with:
\[
\dot{S}(e) = -K \text{sgn}(S) \implies \dot{V}_{SMC} = -KS \text{sgn}(S) \tag{40}
\]
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.
\[
V = V_{NN} + V_{SMC} > 0 \implies \dot{V} = \dot{V}_{NN} + \dot{V}_{SMC}
\]
\( V_{SMC} < 0 \); \( \dot{V}_{NN} \leq 0 \) 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:
\[
H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_{\alpha}\dot{\dot{q}} + F_{\gamma}(\dot{q}) + T_d = M
\tag{41}
\]

Set:
\[
A(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F_{\alpha}\dot{\dot{q}} + F_{\gamma}(\dot{q}) + T_d
\]

So (41) can rewrite:
\[
M(t) = H(q)\ddot{q} + A(q,\dot{q})
\tag{42}
\]

with:
\[
H(q) = H_k(q) + H_s(q) ; \quad A(q,\dot{q}) = A_k(q,\dot{q}) + A_s(q,\dot{q})
\]

Herein \( q \in R^k, \dot{q}, \ddot{q} \) respectively, angle, speed and acceleration of the matching variables; \( H(q) \in R^{k\times k} \) 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; \( F_{\alpha}(\dot{q}) \in R^n \) is a diagonal matrix of viscous friction coefficient; \( F_{\gamma}(\dot{q}) \in R^n \) is the dry friction coefficient; \( T_d \in R^n \) is noise.

Set state variables:
\[
\dot{X}_i = \left( x_{i1}, ..., x_{in} \right)^T = (x_{i1}, x_{i2}, ..., x_{i1}, \dot{x}_{i2})^T
\]
\[
= (\dot{q}_{i1}, \dot{q}_{i2}, ..., \dot{q}_{i1}, \ddot{q}_{i2})^T
\]

From (42) I have:
\[
\ddot{q} = H^{-1} \left( M - A(q,\dot{q}) \right)
\tag{43}
\]
If we consider cross-linking as uncertainty, we have a general model as follows:
\[
\dot{X} = f_0(x) + g_0(x)u_j, t = 1 + n; j = 1 + 2
\tag{44}
\]
In the place: \( f_0(x) = 0 \) and \( u_j = x_{i2} \) at \( j = 1, \forall i \); \( g_0(x) = 1 \), \( \forall i, j \)
\[
f_0(x) = H(q)^{-1} C(q,\dot{q})\dot{q} \quad \text{and} \quad u_j = H(q)^{-1} M \quad \text{at} \quad j = 2, \forall i
\]
So (44) is rewritten:
\[
\dot{X} = (0, f_{i2}(x), 0, f_{i2}(x), ..., 0, f_{i2}(x))^T
\]
\[
+ g_0(x_{i2}, u_{i2}, x_{i2}, u_{i2}, ..., x_{i2}, u_{i2})^T
\tag{45}
\]
With functions \( f_{i2}(x, \Theta_i) \) and \( g_0(x, \Theta_i) \):
\[i = 1, 2, ..., n; j = 1, 2\] are indeterminate functions because they contain an infinite parameter vector of each match \( \Theta \).

We can show the model (45) through n tightly propagated models as follows:
\[
\begin{align*}
\dot{x}_{i1} &= x_{i2} \\
\dot{x}_{i2} &= f(x, \Theta) + g(x, \Theta)u_i + d(t)
\end{align*}
\tag{46}
\]

DOI: 10.9790/0661-2203044252 www.iosrjournals.org 49 | Page
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:

\[
\begin{align*}
q_1 &= \dot{x}_1 \\
q_2 &= \dot{x}_2 \\
\dot{q}_1 &= x_1 \\
\dot{q}_2 &= x_2 \\
\end{align*}
\]

with:

\[
\begin{align*}
f_{12}(x_1, x_2) &= k(-h_2a_1 + h_1a_2) \\
g_{12}(x_1, x_2) &= 1 \\
u_{12} &= k(h_2r_1 - h_1r_2) \\
\end{align*}
\]

Similarly with \(\dot{q}_2\) have:

\[
\begin{align*}
x_1 &= x_2 \\
\dot{x}_2 &= f_{22}(x_{21}, x_{22}) + g_{22}(x_{21}, x_{22})u_{22} \\
\dot{q}_2 &= x_{22} \\
\end{align*}
\]

with:

\[
\begin{align*}
f_{22}(x_{21}, x_{22}) &= k(h_2a_1 - h_1a_2) \\
g_{22}(x_{21}, x_{22}) &= 1 \\
u_{22} &= k(-h_2r_1 + h_1r_2) \\
\end{align*}
\]

Stitch 1: weight \(m_1 = 5\text{kg}\), length \(l_1 = 0.45m\)

Stitch 2: weight \(m_2 = 3\text{kg}\), length \(l_2 = 0.35m\)

Two match variables: \(\theta_1, \theta_2\)

\(-\text{Select controller}:\)

\[
u_{NN} = -z_1 - k_2(t)z_2 - W_2^T S_2 (V_2^T Z_2) \tag{49}\]

With \(z_1 = x_1 - y_2; \dot{z}_2 = x_2 - \alpha_1\) and \(Z_2 = [x_1, x_2, \alpha_1, \dot{x}_1, \dot{x}_2, \dot{\alpha}_1, 1]_T\)

with: \(\alpha_i = -k(t)z_1 - \hat{W}_i^T S_i (\hat{V}_i^T Z_i); Z_i = [x_1, x_2, \dot{x}_1, \dot{x}_2, 1]^T\)

\[
\alpha_i = \frac{\partial \alpha_i}{\partial y_d} \hat{y}_d + \frac{\partial \alpha_i}{\partial \hat{y}_d} \hat{\hat{y}}_d + \frac{\partial \alpha_i}{\partial \hat{W}_i} \hat{\hat{W}}_i + \sum_{j=1}^{k} \frac{\partial \alpha_i}{\partial \hat{\hat{v}}_{ij}} \hat{\hat{v}}_{ij} \]

\[
k_i(t) = \frac{1}{\epsilon_i} \left( \frac{3}{2} \left[ \left\| Z_i \right\| \hat{\hat{W}}_i^T S_i \right\| \hat{\hat{V}}_i^T Z_i \right] + \left\| \hat{\hat{V}}_i^T Z_i \right\| \right); j = 1, 2
\]

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:
\[
\gamma = 3.0; \varepsilon_1 = 1.0; \varepsilon_2 = 5.0; \quad \sigma_{w_1} = \sigma_{w_2} = 1 \times 10^{-3}; \quad \sigma_{v_1} = 1 \times 10^{-4}; \sigma_{v_2} = 1 \times 10^{-3} \quad \Gamma_{w_1} = \Gamma_{w_2} = \text{diag}\{1.0\}; \quad \Gamma_{v_1} = \Gamma_{v_2} = \text{diag}\{1.0\}
\]

The initial weights:
\[
\begin{align*}
\hat{W}_1(0) &= 0.0, \quad \hat{W}_2(0) = 0.0, \quad \hat{V}_1(0), \quad \hat{V}_2(0); \quad \text{take randomized } U_{SMC} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
\end{align*}
\]

and
\[
\begin{align*}
\begin{cases}
\dot{u}_1 = \text{sgn}(s_1(e_1)) + k_1 \dot{w}_1 + \dot{w}_1 - k_1 x_{12} - f_{12} \\
\dot{u}_2 = \text{sgn}(s_2(e_2)) + k_2 \dot{w}_2 + \dot{w}_2 - k_2 x_{22} - f_{22}
\end{cases}
\end{align*}
\]

(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](image1.png)

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](image2.png)

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](image3.png)

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

![Figure 6](image4.png)

Figure 6. Response results when both NN controllers and SMC controllers are involved (q = 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