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:

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

\quad (1)

where \( x_i = (x_1, x_2, \ldots, x_n)^T \) 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: \( u_{NN}(t) \) 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:

\[ u(t) = u_{SMC}(t) + u_{NN}(t) \quad (2) \]
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:
\[
\begin{align*}
\dot{x}_i &= f_i(x_i, x_2, \ldots, x_{i-1}, x_{i+1}) + g_i(x_i, x_2, \ldots, x_i) u_i, \\
\end{align*}
\]
(3)

\begin{itemize}
  \item \textbf{a) Approximating the function by artificial neural network}

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

\textbf{Hypothesis 2:} the desired state vector \(x_{ji}\) with \(j = 1, 2, \ldots, n+1\) is continuous and known in advance. \(x_{ji} \in \Omega_{oji}\) and \(\Omega_{oji}\) is the compact file.

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

\[g_m(z) = W^T S(V^T Z) \quad (4)\]

With \(Z = (Z^T, 1)^T\) : input vector
\[W = (w_1, w_2, \ldots, w_l)^T \in R^l \text{ and } V = (v_1, v_2, \ldots, v_l)^T \in R^{(m+1)i}\]

\[W_{R} \quad \text{is the weight matrix from grade 2 to grade out and from grade in to layer 2 of MNN. The number of neurons in a class is always satisfied} \quad l > 1. \text{ The output signal vector of layer 1 is as follows:} \]

\[S(V^T Z) = (s(v_1^T Z), s(v_2^T Z), \ldots, s(v_{l+1}^T Z), 1)^T \quad \text{with} \quad S(z_o) = \frac{1}{1 + e^{-\gamma z_o}} ; \quad \text{constant} \gamma > 0\]

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: \(h(Z) = W^T S(V^T Z) + \mu \quad (5)\)

\[\text{with:} \quad \forall Z \in \Omega_x \subset R^m ; \quad \mu \quad \text{is the approximation error of NN and } \Omega_x \text{ is a compact file.}\]

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

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:
\[\hat{W} = \hat{W} - W^* ; \quad \hat{V} = \hat{V} - V^*\]

\textbf{Lemma 1:} The estimated MNN (4) can be expressed as follows:
\[
\hat{W}^T S(\hat{V}^T Z) - W^T S(V^T Z) = \hat{W}^T (\hat{S} - \hat{S} \hat{V}^T Z) + \hat{W}^T \hat{S} \hat{V}^T Z + d_u, \quad (6)
\]

Here \(\hat{S} = S(\hat{V}^T Z), \hat{S} = \text{diag}\{s_1, s_2, \ldots, s_{l+1}\}\) with \(s_i(\hat{v}_i^T Z) = d[s(z_o)]/dz_o \vert_{z_o = \hat{v}_i^T Z} ; i = 1, 2, \ldots, l\),
and residual limits \(d_u\) is blocked by \(|d_u| \leq \|V^*\| \|Z\hat{W}^T \hat{S}\| + \|W^*\| \|\hat{S} \hat{V}^T Z\| + \|W^*\| \quad (7)\)
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

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: $\Gamma_{w_1} = \Gamma_{w_1}^T > 0$,
$\Gamma_{v_1} = \Gamma_{v_1}^T > 0$, 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

$$\Omega_{z_1} = \{(x_i, y_d, \dot{y}_d) \mid z_i(t) \leq \sqrt{2c_0 e^{-\frac{\lambda_i}{2}}} + 2c_1 / \lambda_i, x_{d_2} \in \Omega_{d_2}\}$$

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

$$u_{NN_i} = \frac{1}{g_i(x_i)}[-k_i(t)z_i - \hat{W}_i^T S_i(\hat{V}_i^T Z_i)]$$

$$k_i(t) = \frac{1}{\epsilon_i} \left(1 + \int_0^t g_i(\theta z_i + y_d) d\theta\right)$$

$$\left|Z_i \hat{W}_i^T S_i^j \right| + \left|\hat{S}_i \hat{V}_i^T Z_i \right|^2$$

$$\hat{W}_i = \Gamma_{w_1} \left[\hat{S}_i - \hat{S}_i^T Z_i z_i - \sigma_w \hat{W}_i\right]$$

$$\hat{Z}_i = \Gamma_{v_1} \left[Z_i \hat{W}_i^T \hat{S}_i z_i - \sigma_v \hat{V}_i\right]$$

(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 $z_1 = x_2 - \alpha_1$ và $\alpha_1 = u_{NN_1}$ defined in (9) and $\dot{z}_1 = f_1(x_1) + g_1(x_1)(z_2 + \alpha_2) - \dot{y}_d$

Choose $V_1 = \int_0^z \sigma \beta_1(\sigma + y_d) d\sigma$ and transformed by [7], we get:

$$\dot{V}_1 = z_1 \left[\sigma g_1(x_1)(z_2 + \alpha_2) + h_1(Z_1)\right] \Rightarrow \dot{V}_1 = -k_1(t)z_1^2 - \psi_1 z_1 + z_1g_1(x_1)z_2$$

(12)

Step 2: Consider (3) and give $i = 2$ have: $\dot{x}_2 = f_2(x_2) + g_2(x_2)x_3$

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:

$$\dot{z}_2 = z_2 - \alpha_2 = f_2(x_2) + g_2(x_2)(z_3 + \alpha_2) - \dot{\alpha}_2$$

(14)

Choosing: $V_{z_2} = V_1 + \int_0^{z_2} \sigma \beta_2(x_1, \sigma + \alpha_1) d\sigma > 0$

$$\Rightarrow \dot{V}_{z_2} = \dot{V}_1 + z_2z_2^2 f_2(x_2) + \int_0^{z_2} \sigma \left[\frac{\partial \beta_2(x_1, \sigma + \alpha_1)}{\sigma \alpha_1} \dot{\alpha}_2\right] d\sigma$$

(15)

Using (12), (14) we have:
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

\[
\int_0^{z_2} \beta_2(x_2, x_1, \sigma_2, \alpha_2) d\sigma = \hat{\alpha}_2 \int_0^{z_2} \beta_2(x_2, x_1, \sigma_2, \alpha_2) d\sigma
\]

(i)

\[
= \hat{\alpha}_2 \left[ z_2 \beta_2(x_2) - \int_0^{z_2} \beta_2(x_2, \sigma_2, \alpha_2) d\sigma \right]
\]

(ii)

\[
\dot{x}_1 = \frac{\partial \alpha_1}{\partial x_1} \dot{x}_1 + \alpha_1, \quad \dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial x_2} \dot{x}_2
\]

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

\[
\dot{V}_{s2} = -k_2(t)z_2^2 + \psi_1 z_1 + z_2 g_1(x_1)z_2
\]

+ \( z_2 [g_2(x_2)(z_3 + \alpha_2) + h_2(Z_2)] \)

(17)

Here:

\[
h_2(Z_2) = \sum_{i=0}^{1} \frac{\partial \beta_2(x_2, \sigma_2, \alpha_2)}{\partial x_1} d\sigma
\]

\[- \frac{\partial \beta_2(x_2, \sigma_2, \alpha_2)}{\partial x_1} \int_0^{z_2} \beta_2(x_2, \sigma_2, \alpha_2) d\sigma
\]

\[= \beta_2(x_2, f_2(x_2) + \dot{\beta}_2(x_2, \sigma_2, \alpha_2) d\sigma
\]

- \( \dot{\alpha}_2 \int_0^{z_2} \beta_2(x_2, \sigma_2, \alpha_2) d\sigma \)

With: \( Z_2 = [x_2^T, \alpha_2, \dot{\alpha}_2, \dot{x}_2, \alpha_2]^T \in \Omega_{s2} \subset \mathbb{R}^7 \).

Select control function:

\[
\alpha_2 = \frac{1}{g_2(x_2)} \left[ -g_1(x_1)z_1 - k_2(t)z_2 - \dot{W}_2 S_2 \dot{V}_2 Z_2 \right]
\]

(18)

Here:

\[
k_2(t) = \frac{1}{\varepsilon_2} \left[ 1 + \int_0^t \theta g_2(x_1, \theta z_2 + \alpha_2) d\theta
\right]
\]

\[
+ \left[ Z_2^T \dot{W}_2 S_2 \dot{V}_2 + S_2 \dot{V}_2 Z_2 \right]^2 \right]
\]

(19)

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

\[
\dot{W}_2 = \Gamma_{w2} \left[ (S_2 - \dot{S}_2 \dot{V}_2 Z_2 - \sigma_{w2} \dot{W}_2 \right]
\]

\[
\dot{V}_2 = \Gamma_v \left[ (Z_2 \dot{W}_2 \dot{S}_2 \dot{V}_2 - \sigma_{v2} \dot{V}_2 \right]
\]

(20)

With in: \( \Gamma_{w2} = \Gamma_v > 0, \Gamma_{v2} = \Gamma_v > 0 \) and \( \sigma_{w2}, \sigma_{v2} > 0 \).

Through some transformations we have the result:

\[
\dot{V}_{s2} = -\sum_{j=1}^2 [k_j(t)z_j^2 + \psi_j z_j] + z_2 g_2(x_2)z_2
\]

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

\[
V_{sk} = V_{s(k-1)} + \sigma_k \int_0^\infty \beta_2(x_2, x_1, \sigma_2 + \alpha_k) d\sigma
\]

We can design control functions \( \alpha_k \) and jurisprudence \( \dot{W}_k \) and \( \dot{V}_k \) 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:

\[
V_{\infty} = V_{x(n-1)} + \int_{0}^{\infty} \sigma \beta_{n}(x_{n-1}, \sigma + \alpha_{n-1}) \, d\sigma \\
\Rightarrow \dot{V}_{\infty} = \dot{V}_{x(n-1)} + z_{n} \beta_{n}(x_{n}) \dot{z}_{n} + \int_{0}^{\infty} \sigma \left[ \frac{\partial \beta_{n}(x_{n-1}, \sigma + \alpha_{n-1})}{\partial x_{n-1}} \dot{x}_{n-1} + \frac{\partial \beta_{n}(x_{n-1}, \sigma + \alpha_{n-1})}{\partial \alpha_{n-1}} \dot{\alpha}_{n-1} \right] \, d\sigma
\]

Similarly we have:

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

Here

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

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

\[
\dot{V}_{x(n-1)} = -\sum_{j=1}^{n-1} \left[ k_{j}(t) z_{j}^{2} + \psi_{j} z_{j} \right] + z_{n-1} g_{n-1}(x_{n-1}) z_{n-1} + z_{n} \left[ g_{n}(x) u_{NN} + h_{n}(Z_{n}) \right]
\]

Here:

\[
h_{n}(Z_{n}) = \beta_{n}(x_{n}) f_{n}(x_{n}) + z_{n-1} g_{n-1}(x_{n-1}) z_{n-1} + z_{n} \left[ g_{n}(x) u_{NN} + h_{n}(Z_{n}) \right]
\]

The controller is selected as follows:

\[
u_{NN} = \frac{1}{g_{s}(x)} \left[ -g_{s}(x_{n-1}) z_{n-1} - k_{n}(t) z_{n} - \dot{W}_{n} \dot{S}(\dot{V}_{n}^{T} Z_{n}) \right]
\]

with:

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

Select constants \( \varepsilon_{n} > 0 \) and the jurisprudence of the neural network

\[
\begin{bmatrix}
\dot{\hat{W}}_{n} = \Gamma_{\hat{W}} \\
\dot{\hat{S}}_{n} \dot{Z}_{n} = \sigma \hat{W} - \sigma_{\hat{W}} \hat{W}_{n}
\end{bmatrix} \quad (25)
\]

\[
\dot{\hat{V}}_{n} = \Gamma_{\hat{V}} \\
\dot{\hat{S}}_{n} z_{n} = \sigma \hat{V} - \sigma_{\hat{V}} \hat{V}_{n}
\]

with: \( \Gamma_{\hat{W}}, \Gamma_{\hat{V}} > 0, \Gamma_{\hat{W}}, \Gamma_{\hat{V}} > 0 \) and \( \sigma_{\hat{W}}, \sigma_{\hat{V}} > 0 \)
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:

\[
S(e) = e + a_0 \frac{de}{dt} + \cdots + a_{n-2} \frac{d^{n-2}e}{dt^{n-2}} + a_{n-1} \frac{d^{n-1}e}{dt^{n-1}}
\]  \hspace{1cm} \text{(27)}

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

\[
A(s) = 1 + a_1 s + \cdots + a_{n-2} s^{n-2} + a_{n-1} s^{n-1}
\]  \hspace{1cm} \text{(28)}
such that \( \text{(28) is a Hurwitz polynomial} \). With the slip surface \( \text{(27)} \), the design task is to identify a control signal \( u_{\text{SMC}} \) 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:

\[
V_{\text{SMC}}(S) = \frac{1}{2} S^2
\]  \hspace{1cm} \text{(29)}

And the sliding control signal is determined from the condition

\[
\frac{dV_{\text{SMC}}(S)}{dt} = SS = -K \text{sgn}(S) < 0; K > 0
\]  \hspace{1cm} \text{(30)}

The condition \( \text{(30)} \) is called the sliding condition.

I have:

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

From

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

I have:

\[
u_{\text{SMC}} = \frac{K \text{sgn}(S) + D(e, \ldots, e^{(n-2)}) - a_{n-1} f(x)}{a_{n-1} g(x)}
\]  \hspace{1cm} \text{(31)}

With:

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

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 \( u_{\text{NN}} \) is defined (23) and \( u_{\text{SMC}} \) defined in (31) to ensure that the closed system is globally stable and has an adherence error of \( 0 \).

Prove

Select function:

\[
V = V_{\text{NN}} + V_{\text{SMC}}
\]

For \( V \) to be Lyapunov, the function and function must be Lyapunov.

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

\[
V_{\beta} = \int_{0}^{1} \beta(x_{i-1}, \sigma, \alpha_{i-1}) d\sigma, i = 2, 3, \ldots, n
\]  \hspace{1cm} \text{(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:

\[
1 \leq \beta(x_{i-1}, \sigma, \alpha_{i-1}) \leq \beta_1(x_{i-1}, \sigma, \alpha_{i-1}) / g_{i-0}
\]

and the following properties:

(i) \[
V_{\beta} = \frac{1}{2} \int_{0}^{1} \beta(x_{i-1}, \sigma, \alpha_{i-1}) d\sigma \geq \frac{1}{2} \int_{0}^{\frac{1}{2}} \beta(x_{i-1}, \sigma, \alpha_{i-1}) d\sigma = \frac{1}{2}
\]  \hspace{1cm} \text{(33)}

(ii) \[
V_{\beta} = \frac{1}{2} \int_{0}^{1} \beta(x_{i-1}, \sigma, \alpha_{i-1}) d\sigma \leq \frac{1}{2} \int_{0}^{1} \beta(x_{i-1}, \sigma, \alpha_{i-1}) d\sigma = \frac{1}{2}
\]  \hspace{1cm} \text{(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 \( Z_f \) exist in compact files
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

\[ \Omega_j = \left\{ \sum_{i=1}^n z^2_i(t) \leq C_0, \sum_{i=1}^n \| \hat{W}_i \| \leq C_0 \right\} \]

\[ \sum_{j=1}^n \| \tilde{V}_j \|^2 \leq \frac{C_0}{\lambda_{\min}^{-1}(\Gamma_w)}; x_{d(j+1)} \in \Omega_{d(j+1)} \]  \hspace{1cm} (35)

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

\[ \lim_{t \to \infty} \frac{1}{t} \int_0^t z^2_i(\tau) d\tau \leq \frac{2c_i}{1 + g_{j0} \sum c_i} \]

\[ \sum_{i=1}^n z^2_i(t) \leq 2V_{i}(0)e^{-\lambda_s} + \frac{2}{\lambda_s} \sum_{i=1}^n c_i, \forall t \geq 0 \]  \hspace{1cm} (36)

with \( c_i, V_{i}(0), \lambda_s \) are positive constants.

Selected function Lyapunov:

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

The final derivative and transformation we have

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

In that place:

\[ c_i = \varepsilon_j \left( \frac{1}{4} \| \hat{W}_j^* \|^2 + \frac{1}{4} \| \hat{V}_j^* \|^2 + \| \hat{W}_j^* \| + \mu_j^2 \right) \]

\[ + \frac{\sigma_{w}^2}{2} \| \hat{W}_j^* \|^2 + \frac{\sigma_{s}^2}{2} \| \hat{V}_j^* \|^2 \]

\[ \lambda_s = \min \left\{ g_{10} / \varepsilon_1, g_{20} / \varepsilon_2, \ldots, g_{n0} / \varepsilon_n, \sigma_w / \lambda_{\max}^{-1}(\Gamma_w^{-1}), \sigma_s / \lambda_{\max}^{-1}(\Gamma_v^{-1}) \right\} \]

and I have

\[ \begin{cases} V_s(t) \leq V_s(0)e^{-\lambda_s t} + \frac{1}{\lambda_s} \sum_{j=1}^n c_j; \forall t \geq 0 \\ V_s(t) = V_m = \sum_{j=1}^n V_{q_j} \geq \frac{1}{2} \sum_{j=1}^n z^2_i(t) \end{cases} \]  \hspace{1cm} (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 \( \Omega_{zi} \) like that \( Z_i \in \Omega_{zi} \) with every moment.

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

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

\textbf{Slide control}

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

Derivative (38) from time to time is obtained

\[ \frac{dV_{SMC}}{dt} = S \dot{S} \]

with:

DOI: 10.9790/0050-07012535  www.iosrjournals.org 31 | Page
\[
\dot{S}(e) = -K \text{sgn}(S) \Rightarrow \dot{V}_{\text{SMC}} = -KS \text{sgn}(S) \quad (40)
\]

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

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

**Result:** \( V_{NN} \) and function \( V_{\text{SMC}} \) are Lyapunov functions so that \( V \) is a Lyapunov function.

\[
V = V_{NN} + V_{\text{SMC}} > 0 \Rightarrow \dot{V} = \dot{V}_{NN} + \dot{V}_{\text{SMC}}
\]

\( \dot{V}_{\text{SMC}} < 0 \); \( \dot{V}_{NN} \leq 0 \) 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 of freedom is expressed as follows:

\[
H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_a \dot{q} + F_s \dot{q} + T_d = M \quad (41)
\]

Set \( A(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F_a \dot{q} + F_s \dot{q} + T_d \) 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 radical moment vector; \( G(q) \in R^n \) is the gravity vector; \( F_a \in R^{n \times n} \) is a diagonal matrix of viscous friction coefficient; \( F_s(\dot{q}) \in R^n \) is the dry friction coefficient; \( T_d \in R^n \) is noise.

Set state variables:

\[
\dot{X} = (\dot{x}_1, ..., \dot{x}_n)^T = (\dot{x}_{11}, \dot{x}_{12}, ..., \dot{x}_{n_1}, \dot{x}_{n_2})^T \quad M = (\tau_1, \tau_2, ..., \tau_n) ; \quad u = (u_{11}, u_{12}, ..., u_{n_1}, u_{n_2})
\]

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

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

\[
\dot{X} = f_q(x) + g_q(x)u_j ; j = 1 + n; j = 1 + 2 \quad (44)
\]

In the place: \( f_q(x) = 0 \) and \( u_j = x_{i_2} \) at \( j = 1, \forall i ; \quad g_q(x) = 1, \forall i, j \)

\( f_q(x) = H(q)^{-1}.C(q, \dot{q})\dot{q} \) and \( u_j = H(q)^{-1}.M \) at \( j = 2, \forall i \)

So (44) is rewritten:

\[
\dot{X} = (0, f_{x_2}(x), 0, f_{x_2}(x), ..., 0, f_{x_2}(x))^T + g_{x_2}(x_{i_2}, u_{i_2}, ..., x_{n_2}, u_{n_2})^T \quad (45)
\]

With functions \( f_{i_2}(x_{i_2}, \theta_i) \) and \( g_{x_2}(x_{i_2}, \theta_i) \):

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

\[
\begin{align*}
\dot{x}_1 &= x_{i_2} \\
\dot{x}_2 &= f(x_{i_2}, \theta_i) + g(x_{i_2}, \theta_i)u_i + d(t)
\end{align*} \quad (46)
\]

\[
i = 1, 2, ..., n; \quad g_{i_2} = 0
\]
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 neural network controller and the sliding control.

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

The change \( \dot{q}_1 \) in tight backward transmission is as follows:

\[
\begin{align*}
\dot{x}_{11} & = x_{12} \\
\dot{x}_{12} & = f_{12}(x_{11}, x_{12}) + g_{12}(x_{11}, x_{12})u_{12} \quad (47) \\
q_1 & = x_{11} \\
\end{align*}
\]

With:

\[
\begin{align*}
f_{12}(x_{11}, x_{12}) & = k(-h_2\alpha_1 + h_1\alpha_2) \\
g_{12}(x_{11}, x_{12}) & = 1 \\
u_{12} & = k(h_2\tau_1 - h_1\tau_2)
\end{align*}
\]

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

\[
\begin{align*}
\dot{x}_{21} & = x_{22} \\
x_{22} & = f_{22}(x_{21}, x_{22}) + g_{22}(x_{21}, x_{22})u_{22} \quad (48) \\
q_2 & = x_{21} \\
\end{align*}
\]

With:

\[
\begin{align*}
f_{22}(x_{21}, x_{22}) & = k(h_2\alpha_1 - h_1\alpha_2) \\
g_{22}(x_{21}, x_{22}) & = 1 \\
u_{22} & = k(-h_2\tau_1 + h_1\tau_2)
\end{align*}
\]

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: \( \theta_1, \theta_2 \)

- Select controller:

\[
u_N = -z_1 - k_2(t)z_2 - W_2^TS_2(V_2^TZ_2) \quad (49)
\]

With \( z_1 = x_1 - y_d \), \( z_2 = x_2 - \alpha_1 \) and \( Z_2 = \begin{bmatrix} x_1, x_2, \alpha_1, \frac{\partial \alpha_1}{\partial \alpha_2}, \alpha_2, 1 \end{bmatrix}^T \)

with: \( \alpha_1 = -k(t)z_1 - \hat{W}_1^S\dot{S}_1(V_1^TZ_1) \); \( Z_1 = \begin{bmatrix} x_1, y_d, \dot{y}_d, 1 \end{bmatrix}^T \)

\[
\begin{align*}
\omega_1 & = \frac{\partial \alpha_1}{\partial \dot{y}_d} \dot{y}_d + \frac{\partial \alpha_1}{\partial \dot{y}_d} \dot{y}_d + \frac{\partial \alpha_1}{\partial \dot{v}_d} \dot{v}_d + \sum_{j=1}^{R} \frac{\partial \alpha_1}{\partial \dot{v}_{i,j}} \dot{v}_{i,j} \\
k_j(t) & = \frac{1}{\varepsilon_j} \left( \frac{3}{2} + \|Z_1^TV_1^TS_1\|_F^2 + \|\dot{S}_1^TV_1^TS_1\|_F^2 \right); \quad j = 1, 2
\end{align*}
\]

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

Choose the coefficients:
Adaptive Control for Robot Based On Backstepping Technique and Sliding Mode Control

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

\( \Gamma_{v_1} = \Gamma_{v_2} = \text{diag} \{10.0\} \)

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 \( u_{\text{SMC}} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \) and

\[
\begin{cases}
  u_1 = \text{sgn}(s_1(e_1)) + k_1 \hat{W}_1 + \hat{W}_1 - k_{12} x_{12} - f_{12} \\
  u_2 = \text{sgn}(s_2(e_2)) + k_2 \hat{W}_2 + \hat{W}_2 - k_{22} x_{22} - f_{22}
\end{cases}
\]

(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)

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)

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)

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

![Figure 6](image4)

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