Nonlinear Hybrid Controller for Flexible Link Manipulator Based Fuzzy Switch

Aliakbar Naeimi¹, Mohsen Naeimi²

1. Department of Physics Islamic Azad University of Khoram Abad, Iran
2. Ph. D. student, Department of Electrical engineering, Islamic Azad University of Arak, Iran

Corresponding author email: aa.naimi@gmail.com

ABSTRACT: Hybrid Controller has been developed in this study to control the flexible motion of a single-link robotic manipulator. The controller has been designed based on a simplified model of the arm, which only accounts for the first elastic mode of the beam. The controller consists of four parts: linear feedback, a nonlinear sliding mode (SMC) an adaptive fuzzy-neural network (FNN) controller and fuzzy switch.

Keywords: Flexible Link Robot, Linear Feedback Control, Sliding Mode Control, Adaptive Fuzzy-Neural Network Controller and Fuzzy Switch.

INTRODUCTION

Flexible-link robots comprise an important class of systems which include lightweight arms for assembly, civil infrastructure, bridge/vehicle systems, military applications and large-scale space structures. Regarding the recent develops of technology in using robot and also industrial demand to the high speed and quality robots, the idea of using light robots is proposed. Because, a relative deformation happens in the manipulator of high speed robots with heavy load, therefore, the elastic robot is proposed. Thus, modeling and controlling of the robots with elastic links, has its own problems. In addition dynamical structure of such robots includes nonlinear and uncertainty factors in model. So, designing an adaptive controller that is able to provide the performance characteristics such as tracing the reference input, eliminating the disturbance and the conditions of response speed.

In this study, in order to provide above parameters a hybrid controller is used. The controller consists of four parts: linear feedback (PD), a nonlinear sliding mode controller (SMC) an adaptive fuzzy-neural network (FNN) controller and fuzzy switch. The total control signal is computed as follows: where is the linear feedback control, is the sliding mode control and is the adaptive neural control.

A CSMC was used to damp out the vibrations of a flexible cantilevered beam with piezoelectric actuators/sensors. No rigid-body motion is considered in this study (Kim and Inman, 2001) designed a robust controllers and a nonlinear observer for the control of a single-link flexible robotic manipulator were designed (Chalhoub et al., 2006). The controllers consist of a conventional sliding mode controller (CSMC) and a fuzzy-sliding mode controller (FSMC). Moreover, the robust nonlinear observer has been designed based on the sliding mode methodology.

According Gerasimos G. Rigatos (2009) model-based and model-free control of flexible-link robots were presented and compared between representative methods. The paper has examined a particular type of model-based control which is derived from an inversion of the aforementioned dynamic model. However, this inverse dynamics model-based control may result into unsatisfactory performance when an accurate model is unavailable, due to parameters uncertainty or truncation of high order vibration modes.

Dynamics modeling and hybrid suppression control of space robots was presented, performing cooperative object manipulation. In this paper, the Rigid–Flexible Interactive dynamics Modeling (RFIM) approach is introduced as a combination of Lagrange and Newton–Euler methods, in which the motion equations of rigid and flexible members are separately developed in an explicit closed form. To reveal such merits of this new approach, a Hybrid Suppression Control (HSC) for a cooperative object manipulation task will be proposed, and applied to usual space systems (P. Zarafshan, S. Ali A. Moosavian, 2013).

Dynamics model of the flexible manipulator

The physical system consists of a flexible link connected to a revolute joint (see Fig.1). The beam is made of aluminum and has an annular cross section. It is restricted in its motion to the horizontal plane. The
stiffness of the beam in the longitudinal direction is much higher than in flexure. Therefore, only the in-plane transverse deflection of the beam, \(W(x,t)\) is considered in addition to its rigid-body motion. The payload consists of a lumped mass mounted at the free-end of the beam. The dynamic model retains all the coupling terms between the rigid and flexible motions of the beam. The position vector of an arbitrary point on the flexible link is given by

\[
\hat{r}_B = \hat{r}_{B(i,j,k)} + \hat{\omega} \times \hat{r}_B
\]

(1)

Where \(x\) is the time invariant since the longitudinal vibration is neglected (see Fig. 1). The assumed modes method is implemented to approximate \(W(x,t)\), which is considered here to be dominated by the first two elastic modes.

It is written as a linear combination of admissible functions, \(\Phi_i(x)\), of spatial coordinate, and time-dependent generalized coordinates, \(q_{ji}(t)\). The admissible functions are chosen to be the first two Eigen functions of a clamped-free beam derived based on the Euler- Bernoulli beam assumption. Similarly, the position vector of the payload is determined by substituting \(x\) by \(L\) in Eq. 1. The velocity vector of an arbitrary point of the beam is given by

\[
\frac{\partial \hat{W}(x,t)}{\partial \hat{x}} = \frac{\partial \hat{W}(x,t)}{\partial \hat{x}}
\]

(2)

Where \(\hat{\omega}\) is equal to \(\dot{\theta}_k\). The total kinetic energy of the system is written as

\[
T_r = \frac{1}{2} \int_{m_p} (\hat{r}_p, \dot{\hat{r}}_p) dm + \frac{1}{2} m_p (\hat{\dot{r}}_p, \dot{\hat{r}}_p)
\]

(3)

The strain energy stored in the system is expressed as

\[
E_s = \frac{1}{2} \int_0^L EI \left( \frac{\partial^2 W(x,t)}{\partial x^2} \right)^2 dx
\]

(4)

The total virtual work, done on the system, is determined as follows:

\[
\delta W_i = \tau_1 \delta \theta + \frac{1}{2} \int_0^L P(\theta, x) \left( \frac{\partial W(x,t)}{\partial x} \right)^2 dx,
\]

(5)

Where \(\tau_1\) is the non-conservative generalized control torque applied at the base joint.

The second term reflects the stiffening effect of the beam induced by the centrifugal force \(P(\theta, x)\), which can be expressed as

\[
P(\theta, x) = \frac{1}{2} \rho A \dot{\theta}^2 L^2 (1 - \frac{x^2}{L^2}) + m_p L \dot{\theta}^2.
\]

(6)

Note that the variation of the inertial axial force, \(P(\theta, x)\), due to the flexible motion is neglected in this formulation.

The equations governing the rigid and flexible motions of the beam are obtained by implementing the Lagrange principle. The resulting equations of motion are three highly nonlinear, coupled, stiff, second-order ordinary differential equations. These equations are then converted to a set of six first order ordinary differential equations that can be written as
The state equations are solved numerically by using the Gear’s method, which is well suited for solving stiff systems. This model is used, in this study, as a test bed for assessing the combined performances of the controllers and the observer in the presence of both structured and unstructured uncertainties of the plant.

However, it should be emphasized that a simplified version of the model, obtained by ignoring the second elastic mode of the beam, has been used herein in the design of the controllers and the observer. Its equations can be expressed as

\[
\begin{align*}
\dot{x}_1(t) &= f_1(x(t), r_1) \\
\dot{x}_2(t) &= f_2(x(t), r_1) \\
\dot{x}_3(t) &= f_3(x(t), r_1) \\
\dot{x}_4(t) &= f_4(x(t), r_1) \\
\dot{x}_5(t) &= f_5(x(t), r_1) \\
\dot{x}_6(t) &= f_6(x(t), r_1) \\
\end{align*}
\]

Where the state vector is defined to be \( x(t) = [\theta \quad q_{21} \quad \dot{\theta} \quad \dot{q}_{21} \quad \dot{q}_{22}]^T \). The state equations are solved using the Gear’s method, which is well suited for solving stiff systems. This model is used, in this study, as a test bed for assessing the combined performances of the controllers and the observer in the presence of both structured and unstructured uncertainties of the plant.

Novel hybrid controller based fuzzy switch

We have studied linear feedback, a nonlinear sliding mode controller, an adaptive neural network controller, and fuzzy switch which are considered in following parts.

Novel hybrid controller based fuzzy switch

There have been several recent direct adaptive control techniques which have been designed to guarantee overall system stability. The method of uses Lyapunov stability theory in the design of the network learning rule, rather than a gradient descent algorithm like back propagation. The controller consists of four parts: linear feedback, a nonlinear sliding mode controller, an adaptive neural network controller and fuzzy switch (Fig.2).

The total control signal is computed as follows: where is the linear feedback control, is the sliding mode control and is the adaptive neural control. In fact fuzzy switch in order to control situation, selects the appropriate signal for applying to robot.
**Design of Controller**

As discussed earlier in the previous section, our controller has four parts and we will design each of them separately. The resulted equations in the previous part for the robot with flexible link are all nonlinear in which indefinite terms are observed as well. The inaccuracy in the model may be due to uncertainty in basic plan or because of ignoring dynamic parameters such as friction, coriolis acceleration etc. Therefore, designing a control with some robust parts seems inevitable. A SMC controller can be a good choice to protect against the above-mentioned conditions. In addition, a SMC can guarantee system stability because Lyapunov criterion. However, a linear controller (PD) is suitable for the system transient respond. Such parameters of rise time and overshoot can be improved by setting coefficients $K_P$, $K_D$. Using a fuzzy-neural controller, we consider the issue of the control’s adaptation with plant. The adaptive controller FNN improves the system through minimizing the tracking error of the steady state response.

**Design of Nonlinear Sliding Mode Controller**

In order to design nonlinear sliding mode control, we should look at Eq. 9 and 10. In these equations, we choose the $\theta$ link angle with the horizon as the output of the system, whereas the applied torque to the joint $\tau_1$ is considered as the input of the system, so that the outcome system is an SISO.

All controllers are designed based on the following $\theta$ equation:

$$\dot{\theta} = f_{r_1}(\tilde{x}_r, \tau_1) = g_{r_1}(\tilde{x}_r) + b_{\tau_1}(\tilde{x}_r)\tau_1$$

The term $b_{\tau_1}(\tilde{x}_r)$ is considered to be fully known. However, $g_{r_1}(\tilde{x}_r)$ is treated as an unknown term. It has been approximated by the following nominal function $\tilde{g}_{r_1}$:

$$\tilde{g}_{r_1}(\tilde{x}_r) = -\frac{x_{r_2}(700x_{r_3}x_{r_4} - 70x_{r_3}^2 - 770,000)}{300x_{r_2}^2 + 6}.$$  

Only the upper bound of the model imprecision is assumed to be known. It is defined as

$$|\Delta f_{r_3}| = |f_{r_3} - \tilde{f}_{r_3}| = |g_{r_1} - \tilde{g}_{r_1}| \leq F_3$$

Since the task of the controller is to force $\theta$ to track the desired angular displacement $\theta_d$, then the tracking error is defined to be
\[ e_i = x_{r1} - x_{rlid} = \theta - \theta_d \]  

(16)

Accordingly, the sliding surface is expressed as

\[ s(\vec{e}, t) = \dot{\vec{e}}_i + \lambda \vec{e}_i \]  

(17)

Based on the nominal function \( \hat{g}_{r1} \) of the system, the continuous control law \( \tau_{eq} \), satisfying \( \dot{s}(\vec{e}, t) = 0 \), is expressed as

\[ \tau_{eq} = b_i^{-1}\{ -\hat{g}_{r1} + \dot{x}_{id} - \lambda \dot{\vec{e}}_i \} \]  

(18)

Once on the surface, the dynamic response of the system is governed by

\[ \left\{ \frac{d}{dt} + \lambda \right\} e_i = 0 \]  

(19)

The tracking error will be driven to zero by selecting \( \lambda \) to be a strictly positive constant. To force the system trajectory to converge to the sliding surface in the presence of both model uncertainties and disturbances, the feedback control torque \( \tau_i \) is defined as

\[ \tau_i = \tau_{eq} - b_i^{-1}k \text{sgn}(s) \]  

(20)

Where \( k \) is determined by satisfying the following sliding condition:

\[ \frac{d}{dt} V_i = \frac{1}{2} \frac{d}{dt} s^2(\vec{e}, t) \leq -\eta|s| \]  

(21)

Note that \( \eta \leq 1/2s^2 \) is a positive definite function. It represents the squared distance between the sliding surface and any representative point of the system. The selection of \( \eta \) to be strictly positive will ensure that \( \dot{V}_i \) is negative definite. Therefore, \( V_i \) becomes a Lyapunov function that decreases along all trajectories of the system; thus, causing the sliding surface to become an invariant set. It can be easily proven that the above inequality is satisfied by selecting \( k \) to be

\[ k \geq \eta + F_3 \]  

(22)

To alleviate the chattering problem induced by the switching term in the control signal, the \text{sgn}(s) term in Eq. 20 is often replaced by a saturation function as follows:

\[ \tau_i = \tau_{eq} - b_i^{-1}k \text{sat}\left(\frac{s}{\Phi}\right) \]  

(23)

Where \( \Phi \) is the thickness of the boundary layer. It is considered herein to be time-variant. Therefore, to ensure convergence of the system trajectory to the boundary layer, the sliding condition in Eq. 21 had to be modified to the following form:

\[ \frac{1}{2} \frac{d}{dt} s^2(\vec{e}, t) \leq (\Phi - \eta)|s| \]  

(24)

The above condition can be satisfied by changing the expression of \( \tau_i \) as follows:

\[ \tau_i = \tau_{eq} - b_i^{-1}\tilde{k} \text{sat}\left(\frac{s}{\Phi}\right) \]  

(25)

Where \( \tilde{k} = k - \Phi \). The differential equation governing the behavior of \( \Phi \). It is given by

\[ \Phi + \lambda \Phi = k(\theta_d) \]  

(26)

Where \( k(\theta_d) \) is defined in Eq. 22.

Up to this stage, the formulation has only dealt with the design of the SMC.

**Design of Linear Controller**

To produce the control signal \( u_{pd} \) (linear control output), we consider a simple controller with the following transformation function:

\[ u_{pd} = k_p e(t) + k_d \dot{e}(t) \]  

(27)

\( e \) is the system error which is gained via \( e = \theta - \theta_d \). \( \dot{e} \) is derivative of the system error.
Different methods are suggested for PID controller designing. All these methods, notice the process of choosing the control parameters in order to provide the desired operation characteristics. The method that we have used here is Ziegler-Nichols approximate method. Based on characteristics of transient response of the system under control, Ziegler-Nichols has proposed rules for determining proportional gain $K_p$ and derivation time $T_d$.

**Design of Fuzzy Neural Network Controller**

In this section, we will construct a feed forward four layers fuzzy neural network that is presented by Ching-Hung Lee  Ching-Cheng Teng (2000). To implement the fuzzy control rules stated in Eq. 28 first layer accepts input variables. It nodes represent input linguistic variables second layer is used to calculate gaussian membership values nodes in this layer represent the terms of the receptive linguistic variables. Nodes at third layer represent fuzzy rules. The links between third layer and fourth layer are connected by the weighing values $w_j^i$.

![Figure 3. The configuration of the proposed FNN.](image)

For a multi-input single output FNN system, let $x$ be the input linguistic variable and $\alpha_j$ as the firing strength of rule $j$, which is obtained by the product of the grades of the membership function $\mu_{A_i}(x)$ in the antecedent. If $w_j$ represents the $j$-th consequence link weight, the inferred value $y_i$ is then obtained by taking the weighted sum of its input.

The proposed FNN realized the following fuzzy control rules:

$$R^j: \text{IF } u_{ij} \text{ is } A_{ij}, \ldots, u_{nj} \text{ is } A_{nj}, \text{ then } y = w_j$$

(28)

Where for $i = 1, \ldots, n$, $u_{ij} = x_i$, $A_{ij}$, $A_{nj}$ are fuzzy sets , $w_j$ is a fuzzy singleton and $n$ is the number of inputs.

Finally, the output of FNN is obtained:

$$y^* = \sum_{j=1}^{m} \alpha_j w_j$$

(29)

Where $\alpha_j = \prod_{i=1}^{n} \mu_{A_{ij}}(u_{ij})$

**I. Layered Operation of the FNN**

Next we shall indicate the signal propagation and operation functions of the nodes in each layer. In the following description $U^1_k$ denotes the $i$-th input of a node in the $k$-th layer, $O^i_k$ denotes the $i$-th node output in layer $k$.

**Layer 1: Input layer:** The nodes in this layer only transmit input valve to the next layer directly.

$$\alpha_i^1 = u_i^1$$

(30)

From this equation, the link weight at first layer $w_j^i$ is unity.

**Layer 2:** Membership layer: In this layer, each node performs a membership function and acts as a unit of memory. The Gaussian function is adopted here as a membership function, thus we have:

$$\alpha_{ij}^2 = \exp \left(- \frac{(u_{ij}^2 - m_{ij})}{(\sigma_{ij})^2} \right)$$

(31)
Where \( m_{ij} \) and \( \sigma_{ij} \) are the center (or mean) and width (or standard deviation – STD) of the Gaussian membership function. The subscript \( ij \) indicates the \( j \)-th term of the \( i \)-th input \( x_i \).

Layer 3: Rules layer: The nodes in this layer one called rate nodes, the following AND operation is applied to each rule node to integrate these fan-in values,

\[
\alpha^3_i = \prod_{j} u^3_i = \exp \left\{ -\frac{1}{2} \left[ D_i \left( u_i^2 - m_i \right) \right]^T \left[ D_i \left( u_i^2 - m_i \right) \right] \right\}
\]  

(32)

Where \( D_i = \text{diag} \left[ 1/\sigma_{i1}, 1/\sigma_{i2}, ..., 1/\sigma_{in} \right] \),
\( u_i = [u_{i1}, u_{i2}, ..., u_{in}]^T \), and \( m_i = [m_{i1}, m_{i2}, ..., m_{in}]^T \).

The output \( o^3_i \) of a rule node represents the “firing strength” of its corresponding rule.

Layer 4: Output layer: Each node in this layer is called an output linguistic node. This layer performance the defuzzification operation. The node output is a linear combination of consequences obtained from each rule, That is

\[
y_j = o^4_j = \sum_{i=1}^{m} u^4_{ij} w^4_{ij}
\]

(33)

Where \( u^4_{ij} = \alpha^3_i \) and \( w^4_{ij} \) (the link weight) is the output action strength of the \( j \)-th output associated with the \( i \)-th rule. The \( w^4_{ij} \) are the tuning factors of this layer.

Finally, the overall representation of input \( x \) and the \( m \)-th output \( y \) is:

\[
y_m(k) = o^4_m(k) = \sum_{j=1}^{n} w^4_{mj} \prod_{i=1}^{n} \exp \left\{ -\frac{\left| x_i(k) - m_{ij} \right|^2}{(\sigma_{ij})^2} \right\}
\]  

(34)

Where \( m_{ij}, \sigma_{ij}, \) and \( w^4_{mj} \) are tuning parameters.

II. Learning Algorithm
Consider the single Output case for simplicity. Our goal is to minimize the following cost function:

\[
E(k) = \frac{1}{2} \left( y(k) - \hat{y}(k) \right)^2 = \frac{1}{2} \sum_{i=1}^{m} \left( y(k) - o^4_i(k) \right)^2
\]

(35)

Where \( y(k) \) is the desired output and \( \hat{y}(k) = O^4(k) \) is the current output for each discrete time \( k \). In each training cycle starting at the input nodes in the current output \( \hat{y}(k) \).

By using BP learning algorithm, the weighing vector of the FNN is adjusted such that the error defined in Eq. 35 is less than a designed threshold value after a given number of training cycles. The well-known algorithm may be written briefly as:

\[
W(k+1) = W(k) + \Delta W(k) = W(k) + \eta \left( -\frac{\partial E(k)}{\partial W} \right)
\]

(36)

Where in this case \( \eta \) and \( w \) represent the learning rate and tuning parameters of the FNN. Let \( e(k) = y(k) - \hat{y}(k) \) and \( w = [m, \sigma, w]^T \) be the training error and weighting vector of the FNN, then the gradient of error \( E(.) \) in Eq. 35 with respect to an arbitrary weighting vector \( w \) is:

\[
\frac{\partial E(k)}{\partial W} = -e(k) \frac{\partial \hat{y}(k)}{\partial W} = -e(k) \frac{\partial o^4(k)}{\partial W}
\]

(37)

By recursive application of the chain rule, the error term for each layer is first calculated, the parameters in the corresponding layers are adjusted, with the FNN Eq. 34 and cost function defined in Eq. 35, derive the update rule of \( w^4_{ij} \):

\[
w^4_{ij}(k+1) = w^4_{ij}(k) - \eta^w \frac{\partial E(k)}{\partial w^4_{ij}}
\]

(38)

Where \( \frac{\partial E(k)}{\partial w^4_{ij}} = -e(k) \alpha^3_i \)
Similarly, the update laws of $m_{ij}, \sigma_{ij}$ are

$$m_{ij}(k+1) = m_{ij}(k) - \eta \frac{\partial E(k)}{\partial m_{ij}}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta \sigma \frac{\partial E(k)}{\partial \sigma_{ij}}$$

(39)

(40)

Where

$$\frac{\partial E(k)}{\partial m_{ij}} = -\sum_k e(k)w_{ik}o_k^2 \frac{2(x_i - m_{ij})^2}{(\sigma_{ij})^2}$$

$$\frac{\partial E(k)}{\partial \sigma_{ij}} = -\sum_k e(k)w_{ik}o_k^2 \frac{2(x_i - m_{ij})^2}{(\sigma_{ij})^3}$$

(41)

(42)

The BP algorithm is a widely used algorithm for training multilayer network by means of error propagation via variation calculus. But its success depends upon the quality of the training data.

**Design the Fuzzy Switch**

In order to design a fuzzy controlling switch which can select appropriate control signal for plan regarding to controlling regions of signal; we tried to use a fuzzy switch. In simplest form of this switch we used a sigmoid function in which average and variance are related to system error. It means system error defines that which rate should be used by function to select outputs of adaptive and sliding mode controls.

![Figure4. The Fuzzy Switch](image)

**RESULTS AND DESCUSSION**

At this stage, the equations are simulated with numerical values of the system listed: Cross sectional area of the beam (A) is equal $7.2839 \times 10^{-4}$ $m^2$, Outer radius of the beam (Ro) 0.038 $m$, Inner radius of the beam (Ri) 0.0349 $m$, Length of the beam (L) 2.3 $m$, Mass of the beam (MB) 4.535 $Kg$, Payload mass (mp) 3.405 $Kg$ and Density of aluminum ($\rho$) $2707 kg/m^3$.

The designed controller is applied to the plant and the results, are then analyzed.

In simulating, the input torque is considered as step input and calculated the step response of system. Fig. 5 shows the system output. The system error tends toward zero and the response is with a desirable rise time and without overshoot. It is observed that from an iteration of about 80, that controller has switched to adaptive, system has suitable output and the tracking error is approximately zero. In Fig. 6 the control signal has also been looked at.

![Figure5. Step Response](image)
A control signal of SMC with a range of about 20 to -130 Nm without severe vibration has resulted. The control signal of PD controller with the range of 0 to 10 Nm at iteration 40 has become zero because the system error has also become zero at this iteration. FNN does not have a role in producing the total control.
signal applied to the plan before the iteration of 40. However, when the parameters of FNN network are formed based on error, it produces the required control signal and takes the control of the system.

CONCLUSIONS

In this study, a kind of nonlinear hybrid controller as the direct adaptive mode for controlling a robot with a flexible link was designed and simulated. There were indefinite and nonlinear terms in the dynamic equations of the robot. Nevertheless, with its robustness, the controller could well face these parameters. Through minimizing the tracking error, the control also showed an adaptive quality. Many studies have been carried out regarding the control of elastic-link robots; while few of them have pursued compound works. By choosing the above compound method, we tried to resolve the inefficiencies of conventional controllers. Comparing the results with Chalhoub’s research, the system response is more desirable. The applied range of torque to the plan is between -14 to 5 Nm, while in this reference with the given equations, it even reaches 305 Nm. We managed to get the steady state error of system close to zero. That is to say, the control has eliminated all the vibrations of the robot’s end effector. The control signal applied to the plan has also the minimum vibrations; therefore, it can be easily produced and applied to the robot in the laboratory.

REFERENCES