THE UNIVERSITY OF BIRMINGHAM

CONTROL SYSTEMS GROUP

RESEARCHERS

- Mietek A. Brdys (Prof. Tech. Sci., PhD, DSc, CEng, MIEE, SMIEEE, FIMA)
- Duc Thang Nguyen (PhD Student)
- Ruiyun Qi (PhD Student)

Introduction

In the last few decades, nonlinear control using neural networks and fuzzy logic has undergone rapid development. Neural networks offer exciting advantages such as adaptive learning, parallelism, fault tolerance and generalization. Fuzzy logic systems provide human reasoning capabilities to capture uncertainties that cannot be described by precise mathematical models.

Our recent research includes:

- **Neural network modelling and control of an unknown nonlinear continuous-time MIMO plant with unmeasurable states.**
- **T-S fuzzy model based indirect adaptive control for discrete-time MIMO nonlinear systems.**

Dynamic neural network identification and control under unmeasurable plant states

Motivation

The study focuses on identification and control of a dynamic nonlinear MIMO system without plant state measurements using a new dynamic neural network.

Dynamic neural network (DNN)

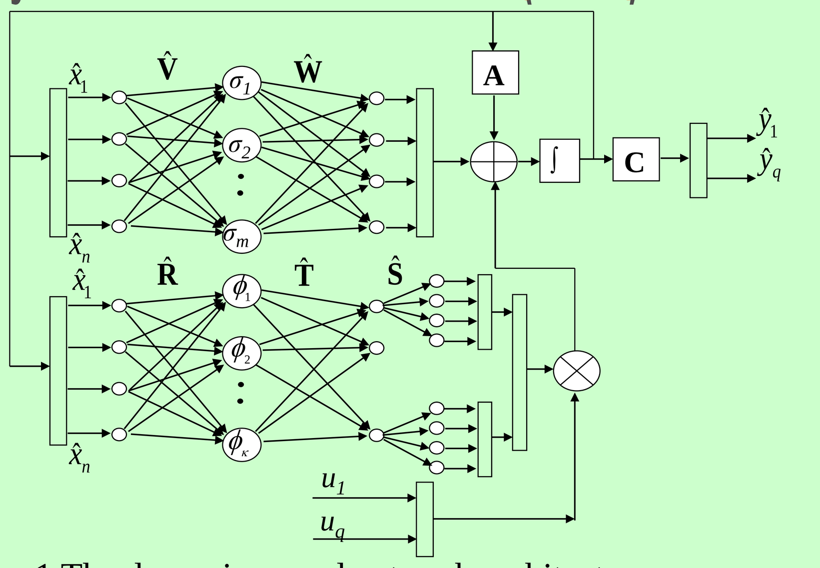

Fig. 1 The dynamic neural network architecture

Mathematically, the DNN can be explained by following matrix form equation:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \hat{\mathbf{W}}\sigma\left(\hat{\mathbf{V}}\hat{\mathbf{x}}\right) + \hat{\mathbf{S}}diag\left(\hat{\mathbf{T}}\phi\left(\hat{\mathbf{R}}\hat{\mathbf{x}}\right)\right)\mathbf{u}$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}}$$

where $\sigma(\cdot)$ and $\phi(\cdot)$ are sigmoid functions.

Learning algorithm

In our work, since the updating is only based on the tracking error, we use gradient descent algorithm for training the model. Following the algorithm, *(k+1)th* update is in the opposite direction to the gradient of the following error criterion

$$E = \frac{1}{2}\int_{kT}^{kT+T} \mathbf{e}_m^T \mathbf{e}_m d\tau$$

Each parameter of the DNN is calculated from previous value as:

$$\hat{w}_i(kT+T) = \hat{w}_i(kT) - \eta\frac{\partial E}{\partial \hat{w}_i}$$

where T is the length of the moving window, η is the learning rate and $\hat{w}_i$ is the updated weight.

Identification results and validation

The nonlinear MIMO system to be controlled is a double link manipulator with its dynamics described as

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta,\dot{\theta})\dot{\theta} + \mathbf{G}(\theta) + \mathbf{F}(\dot{\theta}) = \tau$$

Identification inputs are chosen as sine waves. Identification results and the validation are shown in Fig. 3 and Fig. 4.

DNN model based adaptive control

In this study, feedback linearization control is applied to the DNN model. During control, the control signal is fed to both model and plant. The difference between the model and the plant outputs is used to calculate the updating gradient.

Adaptive model-based control algorithm

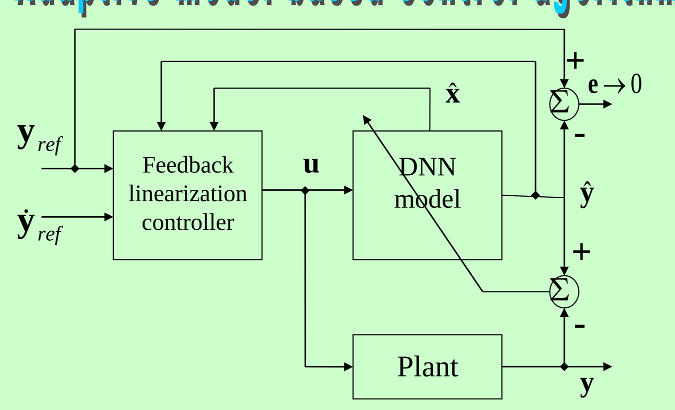

Fig. 2 The control diagram

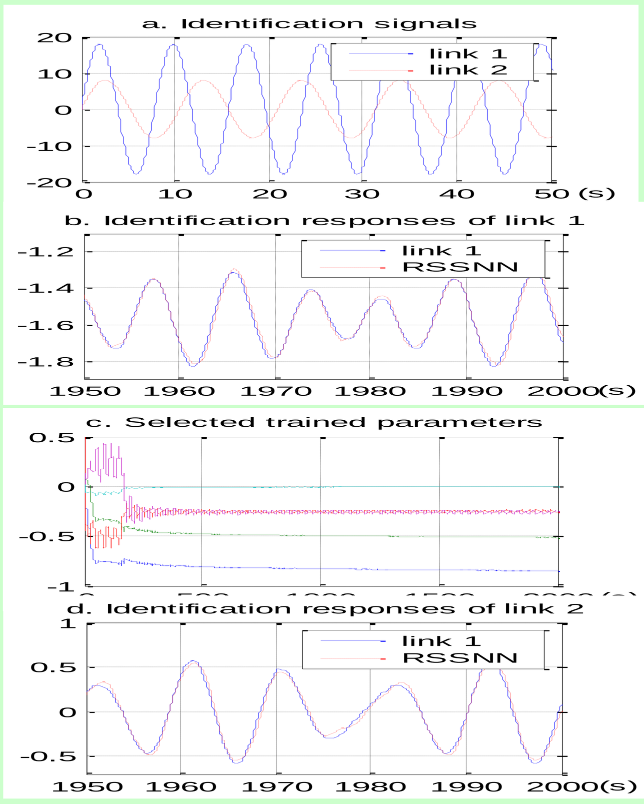

Fig. 3 Double link manipulator identification: 20 nonlinear neurons, b and d: last 50sec.

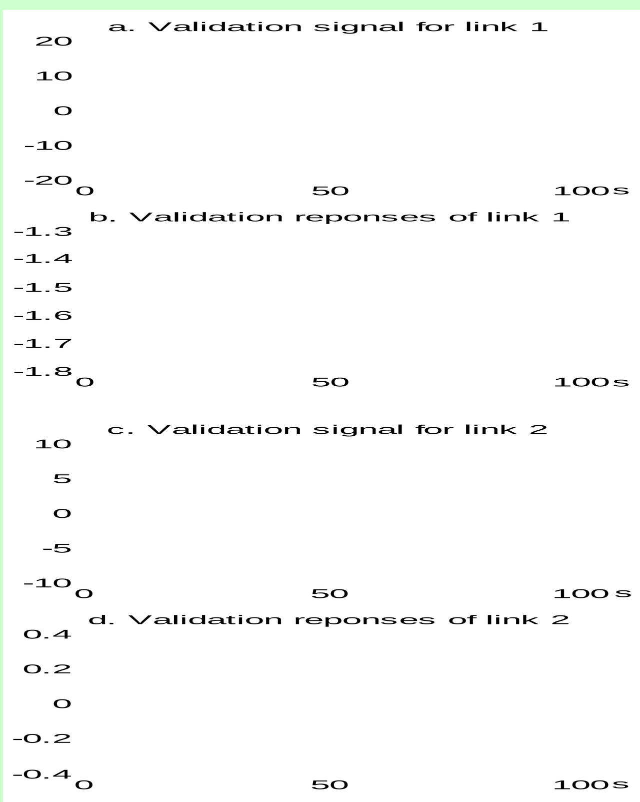

Fig. 4 The validation: red dashed lines: model outputs, blue solid lines: link outputs

Conclusion

The DNN is powerful in capturing the plant dynamics without a priori information about them and the algorithm promises good tracking error convergence.

The control system has been proved to be stable through ultimate bounds of all signals and the convergence of the updating parameters.

Indirect Adaptive Fuzzy T-S Control for Nonlinear Systems Based on Local Linear Integral Controllers

Control Objective

Consider a nonlinear system that can be represented by the following discrete nonlinear different equation model

$$\mathbf{y}(k+1) = f[\mathbf{y}(k), \mathbf{y}(k-1), \cdots, \mathbf{y}(k-n+1); \mathbf{u}(k), \mathbf{u}(k-1), \cdots, \mathbf{u}(k-m+1)]$$

The control objective is to make the system output track a vector of specified bounded trajectories $\mathbf{r}(k)$.

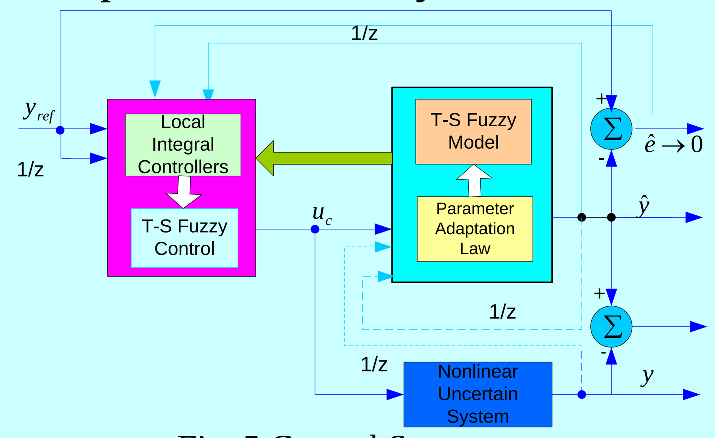

Fig. 5 Control Structure

Control Structure

A diagram of the overall control system structure is shown in Fig. 1. A T-S fuzzy model is used as a design model of the nonlinear system. Local integral controllers are designed for each local linear model of the T-S fuzzy model. Then a T-S fuzzy controller is generated by weighted integration of all the local integral controllers. The parameters of the T-S fuzzy model can be adapted on-line when the nonlinear system varies due to external disturbances or parameter perturbations.

T-S Model based Control Design

The ith rule of T-S fuzzy model is described as follows:

$$\Re^i: \text{ If } \hat{x}_1(k) \text{ is } M_1^i \text{ and } \cdots \text{ and } \hat{x}_4(k) \text{ is } M_4^i, \text{ then}$$
$$\hat{\mathbf{x}}^i(k+1) = A_k^i\hat{\mathbf{x}}(k) + B_k^i\mathbf{u}(k)$$

For each local model, if (A_i, B_i) is controllable, a state feedback controller can be designed. Here we design the controller by using integral control technique in augmented state space.

$$\begin{bmatrix}\hat{\mathbf{x}}_I(k+1)\\ \hat{\mathbf{x}}(k+1)\end{bmatrix} = \begin{bmatrix} I & \mathbf{0}\\ 0 & A^i\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}_I(k)\\ \hat{\mathbf{x}}(k)\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ B^i\end{bmatrix}\mathbf{u}(k) - \begin{bmatrix} I\\ \mathbf{0}\end{bmatrix}\mathbf{r}(k)$$

Define $\hat{\mathbf{z}}(k) = [\hat{\mathbf{x}}_I^T(k), \hat{\mathbf{x}}^T(k)]^T$, then the ith local model in the augmented state space becomes

$$\Re^i: \text{ If } \hat{x}_1(k) \text{ is } M_1^i \text{ and } \cdots \text{ and } \hat{x}_4(k) \text{ is } M_4^i, \text{ then}$$
$$\hat{\mathbf{z}}^i(k+1) = \bar{A}_k^i\hat{\mathbf{z}}(k) + \bar{B}_k^i\mathbf{u}(k) - \bar{R}\mathbf{r}(k)$$

The local integral controller for the local linear model can be designed as

$$\mathbf{u}^i(k) = -L_k^i\hat{\mathbf{z}}(k)$$

where L_k^i is the feedback gain which is obtained by *pole placement.* The overall T-S controller is obtained as

$$\mathbf{u}(k) = -\sum_{i=1}^{N}\omega^i(\hat{\mathbf{x}})L_k^i\hat{\mathbf{z}}(k) / \sum_{i=1}^{N}\omega^i(\hat{\mathbf{x}})$$

Online Parameter Adaptation

Recursive least square estimation (RLSE) can be applied here to find optimal parameters on-line.

$$\hat{\Theta}_j(k+1) = \hat{\Theta}_j(k) + S(k)\Psi(k)[y_j(k+1) - \hat{\Theta}_j^T(k)\Psi(k)]$$

$$S(k+1) = S(k) - \frac{S(k)\Psi(k)\Psi^T(k)S(k)}{1+\Psi(k)S(k)\Psi^T(k)}$$

Stability Analysis

The plant tracking error

$$\mathbf{e}(k+1) = \mathbf{r}(k+1) - \mathbf{y}(k+1) = \mathbf{C}\tilde{\mathbf{x}}(k+1) - \hat{\mathbf{e}}(k+1)$$

Since the model tracking error $\hat{\mathbf{e}}(k+1)$ converges quickly to the reference trajectory due to the control law. Thus, after some fast transient when the algorithm is initialized, the tracking error is directly related to the state error: $\mathbf{e}(k+1) = \mathbf{C}\tilde{\mathbf{x}}(k+1)$

Therefore, we have there following boundedness of tracking error

$$|\mathbf{e}(k+1)| \le \frac{\bar{C}}{1-\bar{\sigma}}(M_{\tilde{\Theta}}M_{\Psi} + M_{\Theta}M_{\Psi}) \quad \text{when } k \to \infty.$$

Thus, all the signals in the closed-loop are bounded.

Simulation Results

Example: Tracking control of a two-link robot manipulator

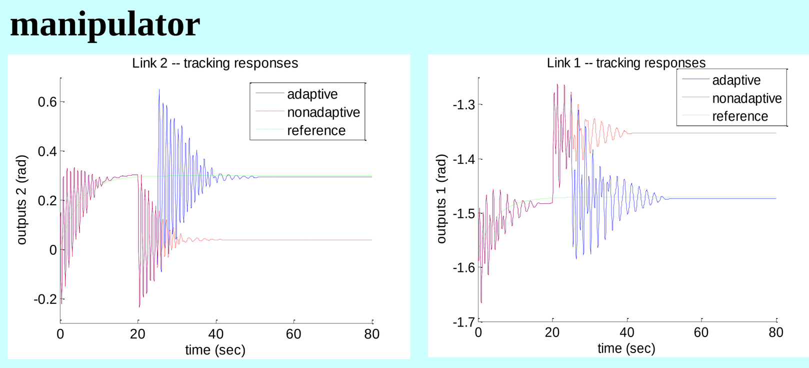

Fig. 6 Tracking responses with disturbance

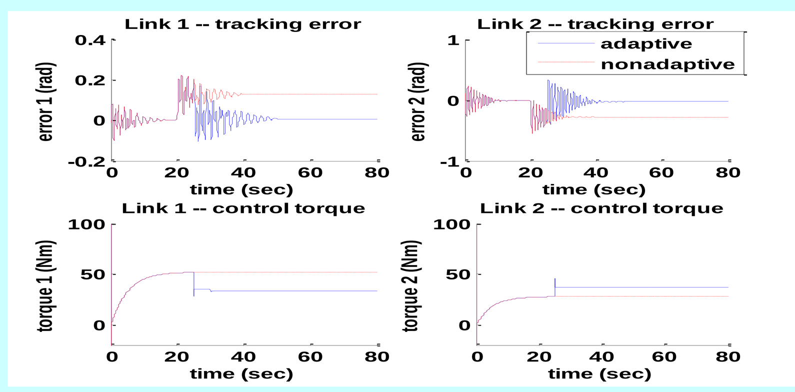

Fig. 7 Tracking error and control torques

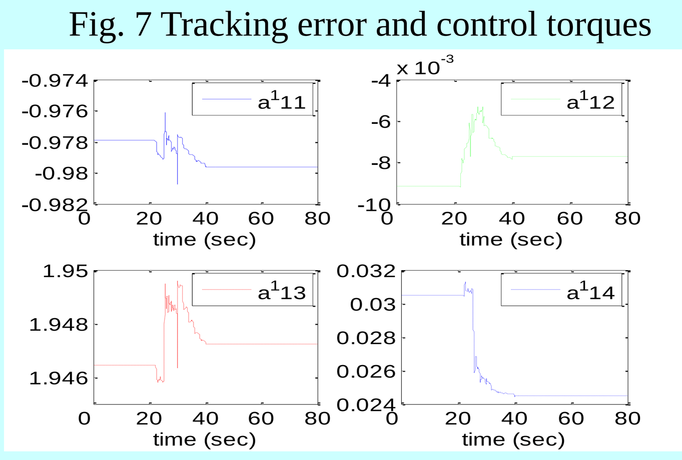

Fig. 8 Parameter adaptation of Rule 1

Tracking responses of DNN Model Based Adaptive Control with disturbances as changes of link masses

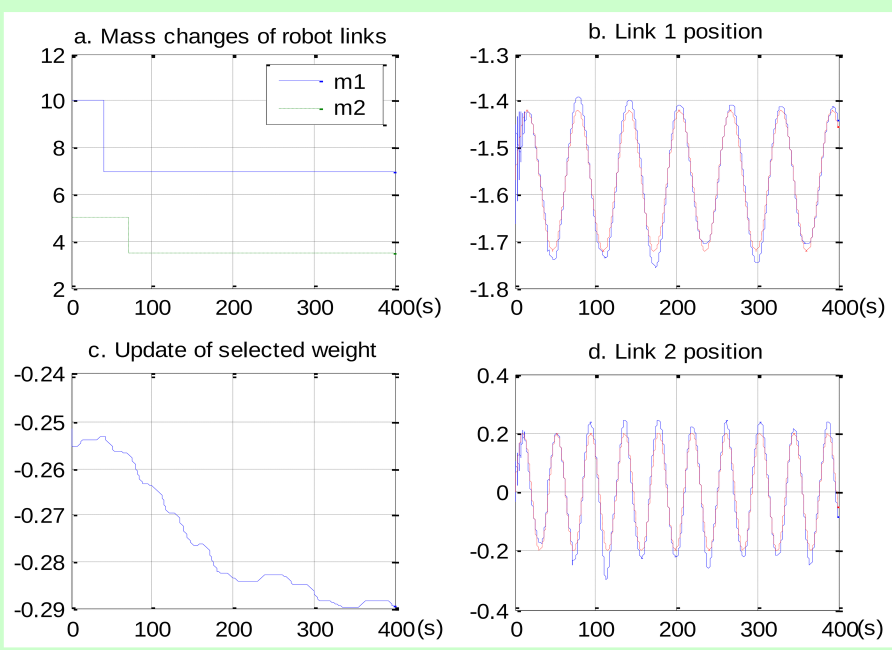

Fig. 9 Sine references (red: model, blue: plant)

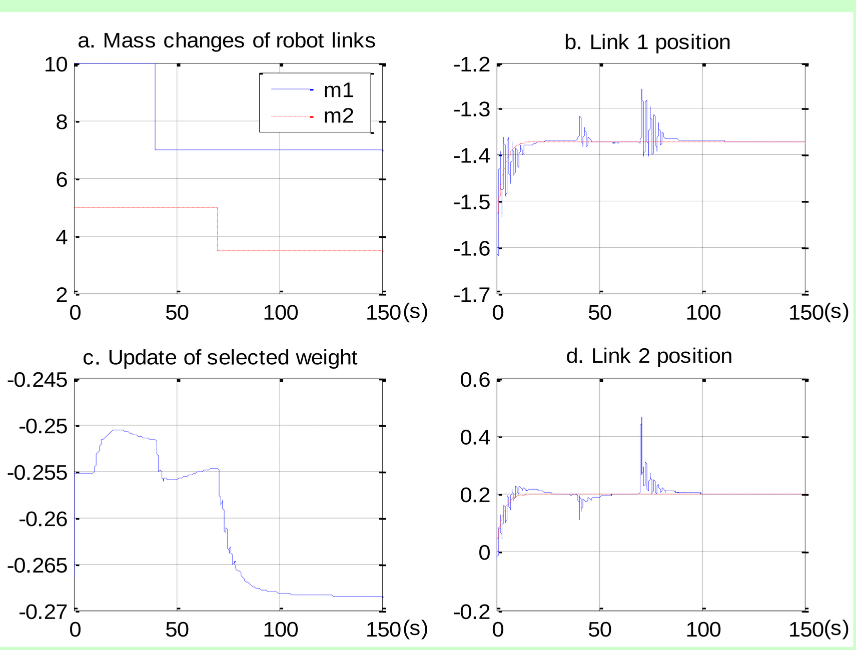

Fig. 10 Constant references (red: model, blue: plant)

Control System Group, Department of Electronic, Electrical and Computer Engineering, University of Birmingham, B15 2TT, UK

Tel: +44 (0)121 414 4354; Fax: +44 (0)121 414 4291

Emails: m.brdys@bham.ac.uk;

Poster designed by Duc Thang Nguyen and Ruiyun Qi