INTELLIGENT CONTROL OF A TWO-LINK FLEXIBLE-JOINT ROBOT, USING BACKSTEPPING, RBF NETWORKS, AND DIRECT METHOD

Withit Chatlatanagulchai and Peter H. Meckl
School of Mechanical Engineering
Purdue University
West Lafayette, Indiana, USA
{chatlata, meckl}@purdue.edu

ABSTRACT
We present a state-feedback control of a two-link flexible-joint robot. First, we obtain desired control laws from Lyapunov's second method. Then, we use radial basis function networks (RBFN) to learn unknown parts of the desired control laws. In this way, the control algorithm does not require the mathematical model representing the robot. To show the effectiveness and practicality of this control algorithm, we performed an experiment on one of the robots in our laboratory.

KEY WORDS
Flexible-joint robot, Intelligent control, Backstepping, Radial basis function networks

1. Introduction

The joint flexibility exists in most robots. It arises from driving components such as actuators, gear teeth, or transmission belts. In some applications, the designers incorporate flexible joints into their products intentionally to absorb impact force and to reduce damage to the parts from accidental collision.

Controller designers should explicitly include joint flexibility in their design because joint resonant frequencies, which are located within the control bandwidth, can be excited and cause severe oscillations. The experiment in [1] suggested that the designers should consider joint flexibility in both modeling and control design.

Controller design of two-link flexible-joint robot is challenging because its model is much more complicated than those of rigid-joint robot and one-link flexible-joint robot. Besides, the number of degree of freedom is twice the number of control inputs, which results in the lost of matching property between nonlinearities and the inputs and the lost of passivity from inputs to link velocities.

References [2] and [3] offer two choices of slow and fast variables in order to transform the dynamical model of the flexible-joint robot into the standard singular perturbation model. Slow-control input, which adds damping to the system, drives the closed-loop system to a quasi-steady state system that has the structure of a rigid-joint robot. Then, fast-control input can be designed using available techniques for the rigid-joint robot.

Reference [4] presents static feedback linearization method. Under the assumption that the kinetic energy of the motor is due mainly to its own rotation, the flexible-joint robot model is feedback linearizable. Reference [5] relaxes this assumption, and applies the so-called dynamic feedback linearization method to a more general robot model. Reference [6] offers a good comparison of three types of controllers: controller developed from decoupled model, backstepping controller, and passivity-based controller.

Newer results use intelligent systems to learn some or all of the unknown parts of the robot model. Reference [7] extends the work in [2] to the case where model uncertainties exist in the system. They use radial basis function networks to estimate unknown functions, and use discontinuous variable-structure controller to provide robustness to the closed-loop system. Reference [8] uses combinations of orthonormal basis functions to estimate unknown functions; they also present an experimental result of one-link flexible-joint robot. Reference [9] uses feedback linearization method and Takagi-Sugeno fuzzy system to replace model uncertainties.

In this paper, we consider the trajectory-tracking task of a two-link flexible-joint robot in horizontal plane. The controller algorithm does not require closed-form mathematical model of the robot. We design control laws from Lyapunov's second method using backstepping structure. Then, radial basis function networks are used to estimate unknown parts of the desired control laws – usually called direct method by adaptive control community.

We organize this paper as follows. Section 2 contains details on the robot and the experimental setup. Section 3 contains radial basis function network background and controller design. Section 4 contains experimental results. Section 5 is conclusions of the paper.

2. A Two-Link Flexible-Joint Robot

Fig. 1 depicts a robot, for which we are designing the controller. The robot operates in horizontal plane, has two links and two motors. Input torque u_1 is applied to the first motor, which drives the first sprocket through a chain. The sprocket is attached to the first link via the first torsional spring that provides joint flexibility. The second motor is situated on the first link. Input torque u_2 is applied to the second motor, which drives the second sprocket. The second sprocket is attached to the second link via the second torsional spring. Note that the second motor's shaft does not share the same axis with the axis of rotation of the second link. This setting is more practical than the shared-axis cases commonly treated in existing literature.

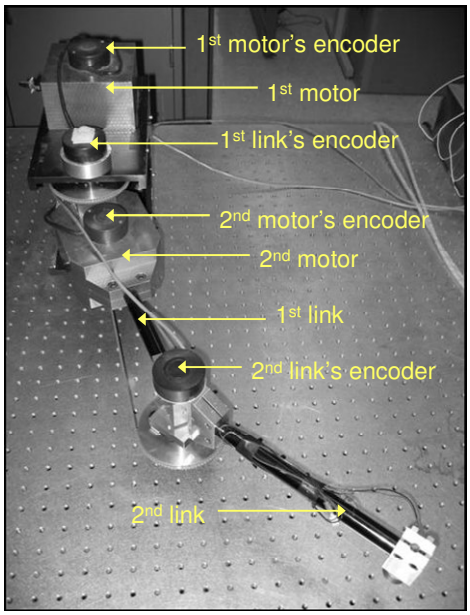

Fig. 1. Photograph of the two-link flexible-joint robot in our laboratory.

There are four optical encoders; each measures angular positions of the two links and the two motors. Angular velocities are obtained from numerical differentiation of the position signals. Two current amplifiers supply current to the two motors.

Fig. 2 depicts overall experimental setup. We use Labview 7.1, Labview Real-Time Module, and Labview FPGA Module to perform hardware-in-the-loop experiment. The data acquisition board is National Instruments' PCI-7831R.

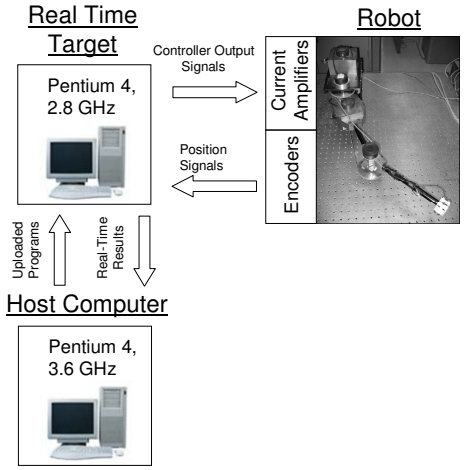

Fig. 2. Diagram showing overall experimental setup.

We let θ_1 be absolute angular position of the first link, θ_2 be relative angular position of the second link, θ_3 be absolute angular position of the first motor, and θ_4 be relative angular position of the second motor. If we let $x_1 = [x_{11}, x_{12}]^T = [\theta_1, \theta_2]^T$ and $x_2 = [x_{21}, x_{22}] = [\dot{\theta}_1, \dot{\theta}_2]^T$ be state vectors representing link position and velocity, $x_3 = [x_{31}, x_{32}] = [\theta_3, \theta_4]^T$ and $x_4 = [x_{41}, x_{42}] = [\dot{\theta}_3, \dot{\theta}_4]^T$ be state vectors representing motor position and velocity, and $u = [u_1, u_2]^T$ be input vector representing input torque, reference [10] shows that the equations of motion of this robot can be put in the following state space form:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= f_2(\bar{x}_2) + g_2(\bar{x}_2)(x_3), \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= f_4(\bar{x}_4) + g_4(\bar{x}_4)(u), \\ y &= x_1, \end{aligned} \tag{1}$$

where $\bar{x}_i = \{x_{11}, x_{12}, \ldots, x_{i1}, x_{i2}\}$; $f_2, f_4 \in \mathbb{R}^2$ and $g_2, g_4 \in \mathbb{R}^{2\times 2}$ are vectors and matrices that contain smooth functions.

3. Controller Design

We, first, present some basics of the radial basis function network followed by the controller design.

3.1 Radial Basis Function Network

Fig. 3 depicts a radial basis function network. Suppose a scalar-valued continuous function $g(z_1, z_2, \ldots, z_n)$ is to be estimated by the neural network, we have $z_1, z_2, \ldots, z_n$ as inputs to the neural network. Variables in the network are defined as follows:

$$Z = [z_1, z_2, \ldots, z_n]^T \in \mathbb{R}^n,$$
$$S(Z) = [s_1(Z), s_2(Z), \ldots, s_l(Z)]^T \in \mathbb{R}^l,$$
$$W = [w_1, w_2, \ldots, w_l]^T \in \mathbb{R}^l,$$
$$g(W,V,Z) = W^T S(Z) \in \mathbb{R}.$$

$s(\bullet)$ is a Gaussian function, which has the form

$$s_i(Z) = \exp\left[-\frac{(Z-\mu_i)^T(Z-\mu_i)}{\sigma_i^2}\right], \quad i = 1, 2, \ldots, l,$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{in}]^T$ is the center of the receptive field and σ_i is the width of the Gaussian function. In this paper, μ_i and σ_i are fixed and the network is linearly parameterized since the weights w_i appear linearly.

Assumption 1: Any smooth nonlinear function $h_i^*(\cdot) \in \mathbb{R}$ can be represented by a radial basis function network with some constant ideal weights W_i^* as

$$h_i^*(\bullet) = W_i^{*T} S_i(Z_i) + \varepsilon_i$$

where $\|\varepsilon_i\| < \varepsilon_{iU}$ is the approximation error with unknown $\varepsilon_{iU} > 0$.

Assumption 2: On the compact set Ω_z, the ideal weights W_i^* are constant and bounded by

$$\|W_i^*\| \le W_{iU},$$

where W_{iU} is unknown.

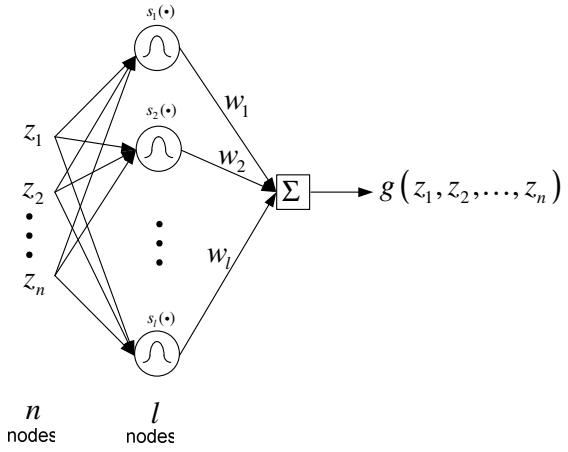

Fig. 3. A radial basis function network. The square represents node whose output contains adjustable parameters.

Since ideal weights are unknown, let $\hat{W}$ be the estimates of W^*. The estimate of the function h is given by

$$\hat{h}(z_1, z_2, \ldots, z_n) = \hat{W}^T S(Z). \qquad (2)$$

3.2 Backstepping Controller

In this section, we design a controller that makes link angular positions θ_1 and θ_2 track desired values whereas all closed-loop signals remain bounded.

The controller, to be designed, is applicable to any plant in the form

$$\begin{aligned} \dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i) x_{i+1}, \ 1 \le i \le 3, \\ \dot{x}_4 &= f_4(\bar{x}_4) + g_4(\bar{x}_4) u, \\ y &= x_1, \end{aligned} \qquad (3)$$

which is more extensive than the plant in (1). We require the following assumptions.

Assumption 3: The inverses of the matrices $g_i, \forall i = 1, \ldots, 4,$ in (3) are nonsingular and positive definite.

Assumption 4: There exist constants $g_{iU} > 0$ such that

$$\left\| \left(g_i^{-1} \right)^{\cdot} \right\| \le g_{iU}, \ \forall i = 1, \ldots, 4.$$

Assumption 5: The desired trajectory x_{1d} and its derivatives, with respect to time, up to the 5^{th} order are known and bounded.

In backstepping design, we try to reduce the error between actual state and desired state of each subsystem. The tracking error is the error of the first subsystem. Let $e_i = [e_{i1}, e_{i2}]^T = x_i - x_{id}$, $i = 1, \ldots, 4$ be those errors.

Step 1: The derivative of the error above is

$$\dot{e}_1 = f_1(x_1) + g_1(x_1) x_2 - \dot{x}_{1d}.$$

Let $x_{2d} \triangleq x_2 \in \mathbb{R}^2$ be the virtual control input. Suppose we know $f_1(\cdot)$ and $g_1(\cdot)$, we should let the virtual input be

$$x_{2d}^* = -c_1 e_1 - g_1(x_1)^{-1} [f_1(x_1) - \dot{x}_{1d}].$$

With Lyapunov candidate $V = \frac{1}{2} e_1^T g_1^{-1} e_1 > 0,$ where $g_1^{-1} > 0,$ we would obtain

$$\dot{V} = -z_1^T c_1 z_1 + \frac{1}{2} z_1^T \left(g_1^{-1} \right)^{\cdot} z_1.$$

Then, $c_1 = c_1^T > 0 \in \mathbb{R}^{2\times 2}$ can be designed to achieve desired stability property.

But because we don't know $f_i(\cdot)$ and $g_i(\cdot)$, we need to find the approximation of x_{2d}^*. Since the unknown part $h_1^*(Z_1) \triangleq g_1(x_1)^{-1} [f_1(x_1) - \dot{x}_{1d}]$ contains two smooth functions of x_1 and $\dot{x}_{1d}$, we proceed by estimating each element of this unknown part by using a radial basis function network.

Using the estimated function in (2), we have

$$x_{2d} = -c_1 e_1 - \begin{bmatrix} \hat{W}_{11}^T S_{11}(Z_{11}) \\ \hat{W}_{12}^T S_{12}(Z_{12}) \end{bmatrix} \triangleq -c_1 e_1 - \hat{W}_1^T S_1(Z_1).$$

Definition 1: We define $(\tilde{\bullet}) = (\bullet^*) - (\hat{\bullet})$, where $(\bullet^*)$ is the actual value to be estimated, $(\tilde{\bullet})$ is the estimated error, and $(\hat{\bullet})$ is the estimated value.

The $\dot{e}_1$ equation becomes

$$\begin{aligned}
\dot{e}_1 &= \dot{x}_1 - \dot{x}_{1d} = f_1 + g_1(e_2 + x_{2d}) - \dot{x}_{1d} \\
&= f_1 + g_1\left(e_2 - c_1 e_1 - \hat{W}_1^T S_1(Z_1)\right) - \dot{x}_{1d} \\
&= f_1 + g_1\left(e_2 - c_1 e_1 - W_1^{*T} S_1(Z_1) + \tilde{W}_1^T S_1(Z_1)\right) - \dot{x}_{1d} \\
&= f_1 + g_1\left(e_2 - c_1 e_1 - g_1^{-1}[f_1 - \dot{x}_{1d}] + \varepsilon_1 + \tilde{W}_1^T S_1(Z_1)\right) - \dot{x}_{1d} \\
&= g_1\left(e_2 - c_1 e_1 + \tilde{W}_1^T S_1(Z_1) + \varepsilon_1\right).
\end{aligned}$$

We choose the Lyapunov function

$$V_1 = \frac{1}{2} e_1^T g_1^{-1} e_1 + \sum_{j=1}^{2} \frac{1}{2} tr\left\{\tilde{W}_{1j}^T \Gamma_{1j}^{-1} \tilde{W}_{1j}\right\},$$

where $\Gamma_{1j} = \Gamma_{1j}^T > 0 \in \mathbb{R}^{l_{1j} \times l_{1j}}$ is a constant gain matrix, l_{1j} is the number of nodes of each neural network. Consider the following adaptation laws

$$\dot{\hat{W}}_{1j} = -\dot{\tilde{W}}_{1j} = \Gamma_{1j} S_{1j}(Z_{1j}) e_{1j} - \sigma_{1j} \Gamma_{1j} \|e_1\| \hat{W}_{1j},$$

where $\sigma_{1j} > 0$ is a scalar design parameter, Z_{1j} is a vector of inputs to the RBFN, the derivative of V_1 is given by

$$\begin{aligned}
\dot{V}_1 &= e_1^T g_1^{-1} \dot{e}_1 + \frac{1}{2} e_1^T \left(\dot{g_1^{-1}}\right) e_1 + \sum_{j=1}^{2} \tilde{W}_{1j}^T \Gamma_{1j}^{-1} \dot{\tilde{W}}_{1j} \\
&= e_1^T e_2 - e_1^T c_1 e_1 + e_1^T \varepsilon_1 + \frac{1}{2} e_1^T \left(\dot{g_1^{-1}}\right) e_1 + \|e_1\| \sum_{j=1}^{2} \sigma_{1j} \tilde{W}_{1j}^T \hat{W}_{1j} \\
&\le e_1^T e_2 - \lambda_{\min}\{c_1\} \|e_1\|^2 + \varepsilon_{1U} \|e_1\| + \frac{1}{2} g_{1U} \|e_1\|^2 \\
&\quad + \|e_1\| \sum_{j=1}^{2} \sigma_{1j} \|\tilde{W}_{1j}\| \left(W_{1jU} - \|\tilde{W}_{1j}\|\right) \\
&= e_1^T e_2 - \|e_1\| \left(\lambda_{\min}\{c_1\} \|e_1\| - \varepsilon_{1U} - \frac{1}{2} g_{1U} \|e_1\| \right. \\
&\quad \left. + \sum_{j=1}^{2} \left\{ \sigma_{1j} \left(\|\tilde{W}_{1j}\| - \frac{W_{1jU}}{2} \right)^2 - \sigma_{1j} \frac{{W_{1jU}}^2}{4} \right\} \right),
\end{aligned}$$

where $\|W_{1j}^*\| \le W_{1jU}$, $\forall j = 1,2$. The coupling term $e_1^T e_2$ will be cancelled in the next step.

Step i: $i = 2$ and 3.
Similar to the first step, we have the ideal virtual control input

$$x_{(i+1)d}^* = -e_{i-1} - c_i e_i - g_i(\bar{x}_i)^{-1} \left[f_i(\bar{x}_i) - \dot{x}_{id} \right].$$

The term $-e_{i-1}$ is there to cancel with the coupling term $e_{i-1}^T e_i$ from the previous step. Since $x_{idj}, \forall j = 1,2$ is a function of $x_k, x_{kd}, \hat{W}_{kj}$, $k = 1, \ldots, i-1$, their derivatives $\dot{x}_{idj}$ can be computed analytically. By employing two RBFNs, we have

$$\begin{aligned}
x_{(i+1)d}^* &= \begin{bmatrix} x_{(i+1)d1}^* \\ x_{(i+1)d2}^* \end{bmatrix} \in \mathbb{R}^2 = -e_{i-1} - c_i e_i - \begin{bmatrix} W_{i1}^{*T} S_{i1}(Z_{i1}) + \varepsilon_{i1} \\ W_{i2}^{*T} S_{i2}(Z_{i2}) + \varepsilon_{i2} \end{bmatrix} \\
&\triangleq -e_{i-1} - c_i e_i - \left[W_i^{*T} S_i(Z_i) + \varepsilon_i \right].
\end{aligned}$$

Z_{ij} are the inputs to the RBFN and are given by $Z_{ij} = \{\bar{x}_i, \dot{x}_{idj}\}$. Replace W_{ij}^* with their estimates $\hat{W}_{ij}$,

$$\begin{aligned}
x_{(i+1)d} &= \begin{bmatrix} x_{(i+1)d1} \\ x_{(i+1)d2} \end{bmatrix} \in \mathbb{R}^2 = -e_{i-1} - c_i e_i - \begin{bmatrix} \hat{W}_{i1}^T S_{i1}(Z_{i1}) \\ \hat{W}_{i2}^T S_{i2}(Z_{i2}) \end{bmatrix} \\
&\triangleq -e_{i-1} - c_i e_i - \left[\hat{W}_i^T S_i(Z_i) \right],
\end{aligned}$$

we can obtain the $\dot{e}_i$ equation by using similar derivation to the first step

$$\dot{e}_i = g_i \left[-e_{i-1} + e_{i+1} - c_i e_i + \tilde{W}_i^T S_i(Z_i) + \varepsilon_i \right].$$

The Lyapunov function is chosen to be

$$V_i = V_{i-1} + \frac{1}{2} e_i^T g_i^{-1} e_i + \sum_{j=1}^{2} \frac{1}{2} tr\left\{\tilde{W}_{ij}^T \Gamma_{ij}^{-1} \tilde{W}_{ij}\right\},$$

where $\Gamma_{ij} = \Gamma_{ij}^T > 0$ are constant gain matrices. The adaptation laws are given as

$$\dot{\hat{W}}_{ij} = -\dot{\tilde{W}}_{ij} = \Gamma_{ij} S_{ij}(Z_{ij}) e_{ij} - \sigma_{ij} \Gamma_{ij} \|e_i\| \hat{W}_{ij},$$

where $\sigma_{ij} > 0$ are scalar design parameters. The derivative of V_i can be done as in the first step with the result as

$$\begin{aligned}
\dot{V}_i \le e_i^T e_{i+1} - \sum_{k=1}^{i} \|e_k\| \Bigg(& \lambda_{\min}\{c_k\} \|e_k\| - \varepsilon_{kU} - \frac{1}{2} g_{kU} \|e_k\| \\
& + \sum_{j=1}^{2} \left\{ \sigma_{kj} \left(\|\tilde{W}_{kj}\| - \frac{W_{kjU}}{2} \right)^2 - \sigma_{kj} \frac{W_{kjU}^2}{4} \right\} \Bigg).
\end{aligned}$$

Step 4: This is the final step. We have the ideal actual control input

$$u^* = -e_3 - c_4 e_4 - g_4(\bar{x}_4)^{-1} \left[f_4(\bar{x}_4) - \dot{x}_{4d} \right].$$

By employing 2 RBFNs, we have

$$\begin{aligned}
u^* &= \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} \in \mathbb{R}^2 = -e_3 - c_4 e_4 - \begin{bmatrix} W_{41}^{*T} S_{41}(Z_{41}) + \varepsilon_{41} \\ W_{42}^{*T} S_{42}(Z_{42}) + \varepsilon_{42} \end{bmatrix} \\
&\triangleq -e_3 - c_4 e_4 - \left[W_4^{*T} S_4(Z_4) + \varepsilon_4 \right].
\end{aligned}$$

Replacing W_{4j}^* with their estimates $\hat{W}_{4j}$, we have the

actual control input as

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \in \mathbb{R}^2 = -e_3 - c_4 e_4 - \begin{bmatrix} \hat{W}_{41}^T S_{41}(Z_{41}) \\ \hat{W}_{42}^T S_{42}(Z_{42}) \end{bmatrix}$$
$$\triangleq -e_3 - c_4 e_4 - \left[\hat{W}_4^T S_4(Z_4) \right].$$

The $\dot{e}_4$ equation becomes

$$\dot{e}_4 = g_4 \left[-e_3 - c_4 e_4 + \tilde{W}_4^T S_4(Z_4) + \varepsilon_4 \right].$$

The Lyapunov function is

$$V_4 = V_3 + \frac{1}{2} e_4^T g_4^{-1} e_4 + \sum_{j=1}^{2} \frac{1}{2} tr\left\{ \tilde{W}_{4j}^T \Gamma_{4j}^{-1} \tilde{W}_{4j} \right\},$$

where $\Gamma_{4j} = \Gamma_{4j}^T > 0$ are constant gain matrices. The adaptation laws are given by

$$\dot{\hat{W}}_{4j} = -\dot{\tilde{W}}_{4j} = \Gamma_{4j} S_{4j}(Z_{4j}) e_{4j} - \sigma_{4j} \Gamma_{4j} \|e_4\| \hat{W}_{4j},$$

where $\sigma_{4j} > 0$ are scalar design parameters. The derivative of V_4 can be done as in the previous step with the result as

$$\dot{V}_4 \le -\sum_{k=1}^{4} \|e_k\| \left(\lambda_{\min}\{c_k\} \|e_k\| - \varepsilon_{kU} - \frac{1}{2} g_{kU} \|e_k\| \right.$$
$$\left. + \sum_{j=1}^{2} \left\{ \sigma_{kj} \left(\|\tilde{W}_{kj}\| - \frac{W_{kjU}}{2} \right)^2 - \sigma_{kj} \frac{W_{kjU}^2}{4} \right\} \right).$$

We can see that, if we select $c_k, \forall k = 1, \ldots, 4$ to have $\left(\lambda_{\min}\{c_k\} - g_{kU}/2 \right) \ge 0$, the overall $\dot{V}_4$ is negative when the error trajectories

$$\|e_k\|, \|\tilde{W}_{kj}\|, \ \forall k = 1, \ldots, 4, \ \forall j = 1, 2$$

are out of the sets

$$\left\{ \|e_k\|, \|\tilde{W}_{kj}\| \ \middle| \ \left(\lambda_{\min}\{c_k\} - \frac{1}{2} g_{kU} \right) \|e_k\| \right.$$
$$\left. + \sum_{j=1}^{2} \left\{ \sigma_{kj} \left(\|\tilde{W}_{kj}\| - \frac{W_{kjU}}{2} \right)^2 \right\} \ge \varepsilon_{kU} + \sum_{j=1}^{2} \sigma_{kj} \frac{W_{kjU}^2}{4} \right\}$$

From this point on, you can use the standard nonlinear analysis techniques given, for example, in [11] to conclude that all error trajectories,

$$\|e_k\|, \|\tilde{W}_{kj}\|, \ \forall k = 1, \ldots, 4, \ \forall j = 1, 2,$$

are globally uniformly ultimately bounded. The ultimate bound, the time the trajectories enter the bound, and all-time exponential-decay upper bound can also be found, but are omitted here.

4. Experimental Results

We have eight RBFNs. Two RBFNs are used in each of the three virtual control inputs, x_{2d}, x_{3d}, x_{4d}; and two RBFNs are used in the actual control input u. Each RBFN has 10 hidden nodes. The inputs to each RBFN are as follows:

$$Z_{11} = \{x_{11}, x_{12}, \dot{x}_{1d1}\},$$
$$Z_{12} = \{x_{11}, x_{12}, \dot{x}_{1d2}\},$$
$$Z_{21} = \{x_{11}, x_{12}, x_{21}, x_{22}, \dot{x}_{2d1}\},$$
$$Z_{22} = \{x_{11}, x_{12}, x_{21}, x_{22}, \dot{x}_{2d2}\},$$
$$Z_{31} = \{x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, \dot{x}_{3d1}\},$$
$$Z_{32} = \{x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, \dot{x}_{3d2}\},$$
$$Z_{41} = \{x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}, \dot{x}_{4d1}\},$$
$$Z_{42} = \{x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}, \dot{x}_{4d2}\}.$$

The centers μ_{ij} of all inputs are evenly spaced in $[-6, 6]$ and widths are $\sigma_{ij} = 3$. The design parameters are as follows:

$$c_i = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}, \ \forall i = 1, \ldots, 4,$$
$$\Gamma_{ij} = 2I^{10\times 10}, \ \forall i = 1, \ldots, 4, \ \forall j = 1, 2,$$
$$\sigma_{ij} = 1, \ \forall i = 1, \ldots, 4, \ \forall j = 1, 2.$$

All initial values are set to zeros. The sampling period is 10 ms. The desired trajectory is obtained from passing a square wave signal of amplitude 3, and with 40-second period into the filter $1/(s+2)^3$.

Experimental results are given in Fig. 4. The control system achieves good overall tracking performance as can be seen from the results in part (a) and (b). Both link angular positions θ_1 and θ_2 are able to follow their desired trajectories quite closely. Part (c) to (f) show estimated values of the unknown functions. Part (g) and (h) are control inputs to the two current amplifiers. Their values can be converted to voltage by multiplying with $10/2^{16}/2$.

5. Conclusions

The controller achieves good tracking performance. However, there are some interesting questions left, probably, as future work. First, the controller is designed based on the assumption that the actual robot is in the nonlinear form (3). The fact that the actual robot may not be exactly in this form may degrade the controller performance. Second, how will the control system handle time-varying case, for example, the change in payload?

References:

[1] L. M. Sweet & M. C. Good, Re-definition of the robot motion control problem: effects of plant dynamics drive system constraints, and user requirements, *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, NV, 1984, 724-731.

[2] M. W. Spong, Modeling and control of elastic joint robots, *Trans. ASME J. Dynamic Systems, Measurement and Control, 109*(4), 1987, 310-319.

[3] S. S. Ge, Adaptive control design for flexible joint manipulators, *Automatica*, *32*(2), 1996, 273-278.

[4] M. Spong & M. Vidyasagar, *Robot dynamics and control* (New York: Wiley, 1989).

[5] A, De Luca & P. Lucibello, A general algorithm for dynamic feedback linearization of robots with elastic joints, *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, Belgium, 504-510.

[6] B. Brogliato, R. Ortega, & R. Lozano, Global tracking controllers for flexible-joint manipulators: a comparative study, *Automatica, 31*(7), 1995, 941-956.

[7] S. S. Ge, T. H. Lee, & C. J. Harris, *Adaptive neural network control of robotic manipulators* (Singapore: World Scientific Publishing, 1998).

[8] A. C. Huang & Y. C. Chen, Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties, *IEEE Trans. Contr. Syst. Technol., 12*(5), 2004, 770-775.

[9] C. W. Park, Robust stable fuzzy control via fuzzy modeling and feedback linearization with its applications to controlling uncertain single-link flexible joint manipulators, *Journal of Intelligent and Robotic Systems, 39*, 2004, 131-147.

[10] H. C. Nho, An experimental and theoretical study of various control approaches to flexible-joint robot manipulator undergoing payload changes, Ph.D. dissertation, Dept. Mech. Eng., Purdue Univ., West Lafayette, IN, 2004.

[11] J. T. Spooner, M. Maggiore, R. Ordonez, & K. M. Passino, *Stable adaptive control and estimation for nonlinear systems* (New York: Wiley Interscience, 2002).

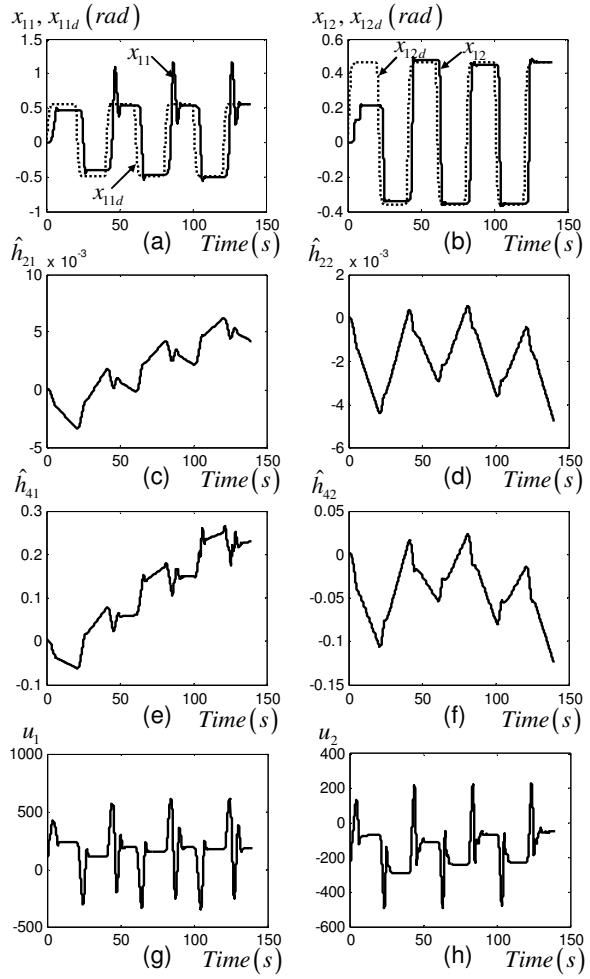

Fig. 4. Experimental results in 140 seconds. (a) θ_1 versus its desired trajectory θ_{1d}. (b) θ_2 versus its desired trajectory θ_{2d}. (c) Estimated function $\hat{h}_{21}$. (d) Estimated function $\hat{h}_{22}$. (e) Estimated function $\hat{h}_{41}$. (f) Estimated function $\hat{h}_{42}$. (g) Control input u_1. (h) Control input u_2.