

Linear control cannot meet the ever increasing demands for better control performance, faster operation speed from industry. Different nonlinear control techniques have been investigated. Computed Torque Control is one of the most intuitive controllers, which relies on the exact cancellation of all the nonlinear dynamics of the system. The disadvantages of such a controller include: (i) exact knowledge of the dynamics is required, (ii) robust closed-loop stability is difficult to guarantee if accelerations are not measurable, which make the scheme not very attractive to industry.

Robots may have to manipulate loads of different weights, sizes and mass distributions, the dynamics of the robots are also different. Controllers designed for a particular nominal payload may not be able to control the system properly for all changes in the parameters. Adaptive control systems can be designed for this situation. In general, an adaptive controller is different from an ordinary controller in that the parameters of an adaptive controller are variable and adjusted by an adaptation mechanism. The so-called "direct" adaptation mechanism has been frequently used in the robotics literature, which computes the controller parameters directly from the tracking error and the states of the system.

It is well known that the dynamics of robots are linear-in-the-parameters. Some prior art adaptive controllers have been proposed based on the linear-in-the-parameters models. The controllers are both stable and robust under certain conditions. Despite the fact that there are several softwares available to help to deduce the dynamics of robots, it is still very tedious and time-consuming to obtain the dynamic equations of the robots except for the very simple ones. It is also extremely error prone to express the dynamics as a product of the regressor and the parameter vector.

Recently, some prior art suggestions have been made to use a neural network to overcome the need for a linear-parametrization model. In many schemes, the neural networks are first "trained" off-line to obtain sufficiently accurate approximations of the systems, and then the networks can be appropriately used to construct controllers. This scheme suffers from the following disadvantages. Since there is

no underlying proof of closed-loop stabilities, systems which performed well in the laboratory might become unstable in real-world operations. Further, these design techniques did not consider the effects of changing dynamics and operations outside a well defined operating range, the resulting system would not be able to handle the disturbances effectively. Other prior art attempts were made to train the neural network on-line during normal system operations. However, because of the underlying approximation nature of the neural networks, closed-loop stability were guaranteed only if the approximation errors of the neural networks corresponding to different nonlinear functions were taken into account. A sliding mode control was often used to suppress the errors and provide closed-loop robustness.

The Invention

Accordingly, it is an object of the present invention to provide general linear-in-the-parameters models for dynamical plants by simply constructing structural networks of known functions of the plants and unknown network parameters in order to eliminate the problems associated with the error prone and time consuming conventional dynamic modeling processes.

It is another object of the present invention to provide an adaptive control system based on the structural networks so as to eliminate the tedious conventional modeling processes, and also eliminate the problems associated with neural network parametrization such as the usual approximation errors, the choices of number of layers in a neural network, and the number of nodes necessary in order to guarantee desired closed-loop performance.

It is still another object of the present invention to provide general linear-in-the-parameters models for a robot by simply constructing structural networks of known functions of position (and velocity) variables and unknown network parameters in order to eliminate the problems associated with the error prone and time consuming dynamic modeling, even though the exact mechanical structure may be not known.

It is yet another object of the present invention to provide an adaptive control system for robots which does not require the tedious, time consuming and error prone processes of dynamic modeling for different robots, which starts from the assignments of the inertial reference coordinate system and the moving relative coordinate systems, formulations of transformation matrices, through the Lagrangian functions and Lagrange-Euler equations, and ends at the problem of parameter identification.

It is a further object of the present invention to provide an adaptive control system for robots which does not have the problems associated with neural network parametrization such as the usual approximation errors, the choices of number of layers in a neural network, and the number of nodes necessary in order to guarantee desired closed-loop performance.

It is still a further object of the present invention to provide a general adaptive control system which is stable and independent of the actual physical structures of robots except for the number of degrees-of-freedom. Therefore, an adaptive controller designed for a n degrees-of-freedom robot can be used for any other kinds of mechanical structural robots as long as the number of degrees-of-freedom is n .

It is yet still a further object of the present invention to provide a general adaptive controller which requires minimum knowledge about the controlled robots, provides user friendly tuning gains for different control performances, and shortens the development period of the control system.

3 Summary of the Invention

The foregoing objects are achieved and the foregoing problems are solved in one illustrative embodiment of the invention. Dynamic modeling of robots becomes simply the construction of Structural Networks based on the systems' own known functions. As long as the size of the network is large enough, the system is exactly parametrized. The approximation error of neural network modeling is removed. A general adaptive controller is presented based on the resulting network models of



robots. It can be shown that the closed-loop system is asymptotically stable and the parameters are all bounded even though the system may be over parametrized. Various controllers can be designed based on the inventive modeling approach presented here using various control techniques.

Brief Description of the Drawing

The following detained description of the invention will be more readily understood when considered in conjunction with the accompanying drawings, in which:

Fig. 1 is a block schematic diagram of a prior art model-based adaptive control system.

Fig. 2 is a block schematic diagram of a prior art neural network based adaptive control system.

Fig. 3 is a block schematic diagram of an illustrative control system constructed in accordance with the principles of the present invention.

Fig. 4 is an expended block schematic diagram of the signal processing block 301 as shown in Fig. 3.

Fig. 5 is an expended block schematic diagram of the i th element, $\hat{g}_i$, of the block 303-1 and the ij th element, $\hat{d}_{ij}$, of the block 303-2 as shown in Fig. 3, respectively.

Fig. 6 is an expended block schematic diagram of the ij th element, $\hat{c}_{ij}$, of the block 303-3 as shown in Fig. 3.

Fig. 8 is a graph showing position tracking versus time of a first simulation.

Fig. 9 is a graph showing control signals provided to the robot versus time of a first simulation.

Fig. 10, 11 and 12 are graphs showing adapted parameters of the networks versus time of a first simulation.

Fig. 13 is an network representation for the ij th element, $d_{ij}(q)$, of $D(q)$ and the i th element, $g_i(q)$ of $G(q)$.

Fig. 14 is an network representation for the ij th element, $c_{ij}(q, \dot{q})$ of $C(q, \dot{q})$.

Fig. 15 is an network block diagram representation for $D(q)$, $C(q, \dot{q})$ and $G(q)$.

In these diagrams, *plant* represents any type of dynamical plants being controlled (e.g. robots and multi-degrees-of-freedom positioning or tracking systems, etc.), and x represents measured or estimated feedback information of the plant such as the states, the outputs and other auxiliary signals.

4 Detailed Description of a Preferred Embodiment

In the ensuing discussion, the structural network modeling underlying the present invention will be firstly discussed. Subsequently, an illustrative control system using the underlying network models will be presented in detail. The following discussion is a simplified summary of a rigorous mathematical theory presented in detail in the report entitled Structural Network Modeling and Adaptive Control of Robots by Shuzhi S. Ge and C.C. Hang, a copy is included in the patent file. As such, the detailed analysis presented in the report should be understood as the proper mathematical expression of the underlying theory and some inaccuracies may be introduced in this summary.

4.1 Structural Network Modeling

A mechanical manipulator consists of a sequence of rigid links connected by either revolute or prismatic joints. For an n degrees-of-freedom (DOF) robot, there are n joint-link pairs with link 0 attached to a supporting base where an inertial coordinate frame is usually established for the dynamic system. The joints and links are numbered outwardly from the base; thus, joint i is the point of connection between link i and link $i - 1$ [1].

Without losing generality, assume that the first n_1 joints are revolute and the rest $n_2 = n - n_1$ joints are prismatic. It can be shown that the ik th element, d_{ik} , of the inertia matrix $D(q) \in R^{n \times n}$ can be exactly parametrized by the following general form:

$$d_{ik} = S_{ik}^T P_{ik} \quad (1)$$

where P_{ik} is the parameter vector and

$$S_{ik} = [1, s_1^T, s_2^T, \dots, s_{2n}^T]^T \quad (2)$$

with the sub-vectors, s_m , $m = 1, 2, \dots, 2n$, being defined by stacking the elements from the set:

$$\begin{aligned} \bar{\mathcal{H}}_m := & \{ \sin^{r_1} q_1 \dots \sin^{r_{n_1}} q_{n_1} \cos^{k_1} q_1 \dots \cos^{k_{n_1}} q_{n_1} q_{n_1+1}^{l_{n_1+1}} \dots q_n^{l_n}, \\ & \forall r_i, k_i, l_j \in [0 \leq r_i, k_i, l_j \leq 2, \sum_{i=1}^{n_1} (r_i + k_i) + \sum_{j=n_1+1}^n l_j = m, \\ & i = 1, \dots, n_1, j = n_1 + 1, \dots, n] \} \end{aligned} \quad (3)$$

The element, 1, in S_{ik} is to take care of the constant parameter of d_{ik} . For clarity, let the set $\bar{\mathcal{H}}_0 := \{1\}$. Then, S_{ik} can be constructed by the elements in the set:

$$\bar{\mathcal{H}}_{ik} := \{\bar{\mathcal{H}}_m, 0 \leq m \leq 2n\} \quad (4)$$

A reduction in the number of elements in $\bar{\mathcal{H}}_{ik}$ is possible by using the fact that $\sin^2 q_i + \cos^2 q_i = 1$ and the definition of d_{ik} . It is not going to be elaborated further since it is not the main contribution here. Let $\underline{\mathcal{H}}_{ik}$ be the minimum subset of $\bar{\mathcal{H}}_{ik}$ such that d_{ik} is exactly parametrized. Then the function d_{ik} is over parametrized (the number of parameters is more than necessary) by $\mathcal{H}_{ik}$ and P_{ik} with some redundant zero elements, where $\bar{\mathcal{H}}_{ik} \supseteq \mathcal{H}_{ik} \supseteq \underline{\mathcal{H}}_{ik}$.

From equation (1), we obtain

$$\frac{\partial d_{ij}}{\partial q_k} = w_{ijk}^T P_{ij} \quad (5)$$

where

$$\begin{aligned} w_{ijk} &= \left[\frac{\partial S_{ij}}{\partial q_k} \right] = [0, w_{s1}^T, w_{s2}^T, \dots, w_{s(2n)}^T]^T \\ w_{sm} &= \frac{\partial s_m}{\partial q_k}, \quad m = 1, 2, \dots, 2n \end{aligned}$$

Therefore, we have

$$\begin{aligned} c_{ijk} &= \frac{1}{2} \left[\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right] \\ &= \frac{1}{2} [w_{kji}^T P_{kj} + w_{kij}^T P_{ki} - w_{ijk}^T P_{ij}] \end{aligned}$$

The same applies to the potential energy, $V(q)$, which can be expressed as

$$V(q) = S_0^T P_0 \quad (6)$$

where

$$S_0 = [1, s_1^T, s_2^T, \dots, s_n^T]^T \quad (7)$$

where $s_m \in \bar{\mathcal{H}}_m$, $m = 1, \dots, n$. Similarly, we have

$$g_k = \frac{\partial V(q)}{\partial q_k} = w_{0k}^T P_0 \quad (8)$$

with

$$w_{0k} = \left[\frac{\partial S_0}{\partial q_k} \right] = [0, w_{s1}^T, w_{s2}^T, \dots, w_{sn}^T]^T, \quad w_{si} = \frac{\partial s_i}{\partial q_k} \quad (9)$$

Because w_{0k} is the partial derivatives of S_0 , the constant parameter P_{01} is nullified by the zero element of w_{0k} . Therefore, we can simply redefine S_0 and w_{0k} as

$$S_0 = [s_1^T, s_2^T, \dots, s_n^T]^T \quad (10)$$

$$w_{0k} = \left[\frac{\partial S_0}{\partial q_k} \right] = [w_{s1}^T, w_{s2}^T, \dots, w_{sn}^T]^T \quad (11)$$

and change the dimension for P_0 accordingly.

Once S_{ik} and S_0 are pre-determined to parametrize all possible variations of $D(q)$ and $V(q)$, the corresponding network based dynamics for robots can be described by the **Network Model I**:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (12)$$

where

$$d_{kj}(q) = S_{kj}^T P_{kj} \quad (13)$$

$$c_{kj}(q, \dot{q}) = \sum_{i=1}^n c_{ijk} \dot{q}_i = \frac{1}{2} \sum_{i=1}^n [w_{kji}^T P_{kj} + w_{kij}^T P_{ki} - w_{ijk}^T P_{ij}] \dot{q}_i \quad (14)$$

$$g_k(q) = w_{0k}^T P_0 \quad (15)$$

It can easily be seen that the full dynamics of robots can be constructed by using the parameters for $D(q)$ and $V(q)$, because $C(q, \dot{q})$ can be constructed based on the parameters of $D(q)$. Note that since the $D(q)$ and $G(q)$ are functions of q only, the sub-networks are static instead of dynamic.

The above parametric description for $G(q)$ is deduced from the network description for the potential energy $V(q)$. Actually, the proposed network modeling approach can be applied to $G(q)$ directly. Let S_{0k} be formed by all the elements in the set $H_{0k} \subseteq \bar{\mathcal{H}}$, and P_{0k} be the corresponding parameter vector such that $g_k = S_{0k}^T P_{0k}$, then we have the Network Model II:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (16)$$

where

$$d_{kj}(q) = S_{kj}^T P_{kj} \quad (17)$$

$$c_{kj}(q, \dot{q}) = \sum_{i=1}^n c_{ijk} \dot{q}_i = \frac{1}{2} \sum_{i=1}^n [w_{kji}^T P_{kj} + w_{kij}^T P_{ki} - w_{ijk}^T P_{ij}] \dot{q}_i \quad (18)$$

$$g_k(q) = S_{0k}^T P_{0k} \quad (19)$$

The networks for $d_{ij}(q)$ and $g_i(q)$ are shown in Fig. 13, and the network for $c_{ij}(q, \dot{q})$ is shown in Fig. 14. The network representation of the Network Model I is shown in Fig. 15. By changing W_0 to S_0 in Fig.15, we have the corresponding network representation of Network Model II.

From the above analysis, it can be seen that

1. Dynamic modeling of robots becomes simply the construction of structural networks based on their own known functions, and the dynamics of a robot can be exactly parametrized by augmenting its parametric space and the corresponding known regressor functions, since all the elements of the inertia matrix $D(q)$ and the potential energy $V(q)$ (or the gravitational forces $G(q)$) can be precisely parametrized by all possible combinations of known functions of positions, such as $\cos q_i$, $\sin q_i$ and q_i (or by the network of the above functions).

2. The present inventive modeling approach is an exact parametrization method if the size of the network is large enough (less than or equal to the maximum). The approximation error of neural network modeling is removed. Modeling errors do exist, however, if network models of smaller sizes are used to capture only the main features of the functions.
3. The present inventive modeling approach is independent of the physical structures, i.e., it can be the same for different structures of robotic manipulators as long as the numbers of degrees-of-freedom are the same.
4. The present inventive modeling approach is applicable to any multi-degrees-of-freedom mechanical systems which are similar to robotic manipulators mathematically.
5. The present inventive modeling approach is applicable to any dynamical systems which can be expressed as networks of known functions of the plants.
6. The present inventive modeling approach can also be used to solve the nonlinear friction modeling problems by simply constructing a function space large enough to include all nonlinear possibilities as for $G(q)$ in the Network Model II (16).

Different identification techniques can be use to estimate the nominal values of the parameters of robots. The nominal models can then be used to design controllers using the various control design techniques. In the next subsection, an adaptive controller is presented based on the above structural network models. Parameter identification is not an essential step for the adaptive controller because it can "learn" and adapt the above parameters accordingly.

4.2 Structural Network Model-Based Adaptive Control

In this section, an adaptive controller based on the network parametric model I (12) is presented. The controller is a simplified version of the more complicated controller

given in the enclosed report. Adaptive Controllers based on the network parametric models II (16) can be obtained by simply expanding the same techniques presented here.

Lemma 4.1 [6]

Let $e(t) = h * r$, where $h = L^{-1}(H(s))$ and $H(s)$ is an $n \times n$ strictly proper, exponentially stable transfer function. Then $r \in L_n^2 \implies e \in L_n^2 \cap L_n^\infty$, $\dot{e} \in L_n^2$, e is continuous and $e \rightarrow 0$ as $t \rightarrow \infty$. If, in addition, $r \rightarrow 0$ as $t \rightarrow \infty$, then $\dot{e} \rightarrow 0$.

Let q_d be the desired position trajectory, and $\dot{q}_d$ and $\ddot{q}_d$ be the desired velocity and acceleration of the desired trajectory. Define the tracking errors as $e = q_d - q$, $\dot{e} = \dot{q}_d - \dot{q}$ and other tracking variables as

$$\dot{q}_r = \dot{q}_d + \Lambda e \quad (20)$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda \dot{e} \quad (21)$$

$$r := \dot{q}_r - \dot{q} = \dot{e} + \Lambda e \quad (22)$$

where $\Lambda = \text{diag}[\lambda_i]$, $\lambda_i > 0$. In this case, $H(s) = \text{diag}[1/(1 + \lambda_i s)]$. According to Lemma 4.1, to study the stability of e and $\dot{e}$, we only need to study the properties of r . Actually, other forms of $H(s)$ can be chosen as well, see [7] for detail.

Let $(\hat{\cdot})$ be the estimate of $(\cdot)$, and define $(\tilde{\cdot}) = (\cdot) - (\hat{\cdot})$. Further, let $\hat{D}(q)$, $\hat{C}(q, \dot{q})$ and $\hat{G}(q)$ correspond to the estimates of $D(q)$, $C(q, \dot{q})$ and $G(q)$, respectively, obtained by substituting the true parameter vector P and P_0 by the estimated parameter vector $\hat{P}$ and $\hat{P}_0$. Accordingly, let $\hat{d}_{ij}(q)$, $\hat{c}_{ij}(q, \dot{q})$ and $\hat{g}_i(q)$ correspond to the estimates of $d_{ij}(q)$, $c_{ij}(q, \dot{q})$ and $g_i(q)$, respectively, obtained by substituting the true parameter vector P_{ij} and P_0 by the estimated parameter vector $\hat{P}_{ij}$ and $\hat{P}_0$, i.e.,

$$\hat{d}_{kj}(q) = S_{kj}^T \hat{P}_{kj} \quad (23)$$

$$\hat{c}_{kj}(q, \dot{q}) = \frac{1}{2} \sum_{i=1}^n [w_{kji}^T \hat{P}_{kj} + w_{kij}^T \hat{P}_{ki} - w_{ijk}^T \hat{P}_{ij}] \dot{q}_i \quad (24)$$

$$\hat{g}_k(q) = w_{0k}^T \hat{P}_0 \quad (25)$$

Accordingly, the networks for $\hat{d}_{ij}(q)$ and $\hat{g}_i(q)$ can be represented by Fig. 5, and

the network for $\hat{c}_{ij}(q, \dot{q})$ is shown in Fig. 6. The network implementation of $\hat{D}(q)$, $\hat{C}(q, \dot{q})$ and $\hat{G}(q)$ as a whole is shown in Fig. 7.

Consider the general controller introduced in [7] of the form

$$\tau = \hat{D}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{G}(q) + Kr + K_i \int_0^t r d\tau \quad (26)$$

The first four terms give PID type control, with K_i introduced to eliminate static tracking errors. Note that PD control is effectively introduced through the definition of r in (22).

A closed-loop error equation can then be obtained as

$$D(q)\dot{r} + C(q, \dot{q})r + K(t)r + K_i \int_0^t r(\tau) d\tau = \tilde{D}\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{G} \quad (27)$$

where

$$\begin{aligned} \tilde{d}_{kj}(q) &= S_{kj}^T \tilde{P}_{kj} \\ \tilde{c}_{kj}(q, \dot{q}) &= \frac{1}{2} \sum_{i=1}^n \left[w_{kji}^T \tilde{P}_{kj} + w_{kij}^T \tilde{P}_{ki} - w_{ijk}^T \tilde{P}_{ij} \right] \dot{q}_i \\ \tilde{g}_k(q) &= w_{0k}^T \tilde{P}_0 \end{aligned}$$

The closed-loop stability property of the system (27) is given by the following theorem (detailed proof can be found in the enclosed report):

Theorem 4.1 *If $K > 0$, $K_i = K_i^T \geq 0$, the closed-loop system (27) is asymptotically stable, i.e., e and $\dot{e} \rightarrow 0$ as $t \rightarrow \infty$ under the following parameter adaptation*

$$\begin{aligned} \dot{\tilde{P}}_{ij} &= \Gamma_{ij} S_{ij} \ddot{q}_{rj} r_i + [\Gamma_{ij} w_{ij1} \ \Gamma_{ij} w_{ij2} \ \dots \ \Gamma_{ij} w_{ijn}] \dot{q} \dot{q}_{rj} \frac{r_i}{2} \\ &\quad + \sum_{k=1}^n \Gamma_{ij} w_{ijk} \dot{q}_j \frac{\dot{q}_{rk} r_i - r_k \dot{q}_{ri}}{2} \end{aligned} \quad (28)$$

$$\dot{\tilde{P}}_0 = [\Gamma_0 w_{01} \ \Gamma_0 w_{02} \ \dots \ \Gamma_0 w_{0n}] r \quad (29)$$

where $\Gamma_{ij} = \Gamma_{ij}^T > 0$, $\Gamma_0 = \Gamma_0^T > 0$.

In some cases, a sliding mode control can be introduced for robust closed-loop stability in the presence of unmodeled dynamics such as frictions, and for different control performance as explained in [7].

Based on the above discussion, it can be seen that

1. Adaptive controllers designed based on the proposed network model is superior to the neural network based controllers because the approximation nature of neural networks is eliminated since the proposed network model uses its own known functions instead of pure "imaginary" basis functions [3][4].
2. Owing to the present invention, complicated dynamic modeling of mechanical systems now becomes simply a systematic construction of Structural Networks based on systems' own functions. Since the network models are valid for all kinds of robots having the same number of DOF, the new modeling approach is independent of the physical structures of the systems.
3. The presented adaptive controllers are based on the network models of robots, and the closed-loop system is asymptotically stable and the parameters are all bounded even though the system may be over parametrized. The controller design for a particular robot can be used to control other kind of robots having the same numbers of DOF.

Further, we have the following remarks to make:

1. If the same controller (26) is used but based on the Neural Network II (16), then it can be shown that the closed-loop system is stable if (i) the adaptation algorithm for $\hat{P}_{ij}$ is the same as (28), and (ii) the adaptation algorithm for $\hat{P}_{0i}$ is, for $\Gamma_{0i} > 0$, given by

$$\dot{\hat{P}}_{0i} = \Gamma_{0i} S_{0i} r_i \quad (30)$$

2. Similar adaptive controllers can be obtained by applying the same procedures to cases when friction modeling is included using a structural friction network to emulate the possible combinations of system frictions.
3. Various controllers can be obtained by (i) re-defining r , (ii) using different parameter adaptation algorithms, and (iii) using various control techniques for the Network Models presented here and their modifications or variations.

4. For those skilled in the art, various adaptive controllers (such as passive controllers, exponentially stable controllers and robust controller combined with other techniques such as sliding mode methods) can be constructed based on the network models presented in the previous section, or based on network models which are presented differently but use systems' own functions as the present invention.
5. For those skilled in the art, various controllers can also be obtained based on modified network models of reduced sizes and having parametrization errors.

4.3 Simulation Example

For the simulation studies, we consider a planar two-link manipulator, whose dynamics are described by

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (31)$$

where

$$\begin{aligned} D(q) &= \begin{bmatrix} m_1 + m_2 + 2m_3 \cos q_2 & m_2 + m_3 \cos q_2 \\ m_2 + m_3 \cos q_2 & m_2 \end{bmatrix} \\ C(q, \dot{q}) &= \begin{bmatrix} -m_3 \dot{q}_2 \sin q_2 & -m_3 (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_3 \dot{q}_1 \sin q_2 & 0.0 \end{bmatrix} \\ G(q) &= \begin{bmatrix} m_4 \cos q_1 + m_5 \cos(q_1 + q_2) \\ m_5 \cos(q_1 + q_2) \end{bmatrix} \end{aligned}$$

where the m_i 's are parameters of interest.

In order to construct the networks, define

$$\begin{aligned} S_{11} &= \begin{bmatrix} 1 \\ \cos q_1 \\ \cos q_2 \end{bmatrix}, \quad S_{12} = S_{21} = \begin{bmatrix} 1 \\ \sin q_1 \\ \cos q_2 \end{bmatrix}, \quad S_{22} = \begin{bmatrix} 1 \\ \cos q_2 \end{bmatrix} \\ S_0 &= [\sin q_1, \sin q_2, \sin q_1 \cos q_2, \cos q_1 \sin q_2]^T \end{aligned}$$

It can be seen that the dynamics of this robot is parametrized by

$$P_{11} = [m_1 + m_2, 0, 2m_2]^T$$

$$P_{12} = P_{21} = [m_2, 0, m_2]^T$$

$$P_{22} = [[m_2, 0]^T$$

$$P_0 = [m_4, 0, m_5, m_5]^T$$

In the simulation results, $\hat{P}_{ijk}$ and $\hat{P}_{0k}$ stand for the k th elements of the vectors $\hat{P}_i$ and $\hat{P}_0$. Suppose initially that we do not have any knowledge about the system, i.e., $\hat{P}_{11}(0) = \hat{P}_{12}(0) = \hat{P}_{21}(0) = 0$, $\hat{P}_{22}(0) = 0$, $\hat{P}_0(0) = 0$, let us study the performance of the proposed controller.

When the adaptation algorithms (28) and (29) are activated with $\Gamma_i = \text{diag}[1.0]$ and $\Gamma_0 = \text{diag}[10.0]$, the position tracking performance of the robot is as shown in Figure 8 and the control inputs are shown in Figure 9. The corresponding parameter adaptations are shown in Figures 10, 11 and 12. It can be seen that the tracking error converges because of the "learning" mechanism. The simulation results thus demonstrate that the proposed adaptive network control can effectively handle changes in the dynamics of the system.

Figure 1 shows a schematic block diagram of a conventional model-based adaptive control system of robots. In general, in order to use this approach with prior art theory, it is assumed that linear-in-the-parameter dynamics are already known. Dynamic modeling of robots is, however, very tedious, time consuming and error prone. The same modeling processes have to be repeated again for different robots of very few mechanical structural changes even though the numbers of DOF are the same.

Fig. 2 is a block schematic diagram of a prior art neural network based adaptive control system. Even though this control approach does not require the linear-in-the-parameters dynamics, approximation errors usually exist in emulating the nonlinear functions of robots, and other control methods, such as the sliding mode control, have to be used to guarantee robust closed-loop stabilities.

In accordance with the invention, the conventional adaptive controller can be im-

plemented in a very intelligent way so that dynamic modeling are not needed when constructing the controller. A comparison of Fig. 1 and Fig. 2 indicates that the corresponding elements are corresponding numerical designations. For example, the plant block 100 in Fig. 1 is labeled block 200 in Fig. 2. Similarly, the signal block 101 in Fig. 1 is labeled block 201 in Fig. 2. Different parts of the same block have the same block label followed by different identification numbers such as 303-1, 303-2 and 303-3 shown in Fig. 3.

Fig. 3 is a block schematic diagram of an illustrative control system constructed in accordance with the principles of the present invention based on the Network Model I. By substituting S_0 into W_0 in Fig. 3, we obtain a schematic block diagram of the control system in accordance with the invention based on the Network Model II.

The signal block 301 in Fig. 3 is expanded in Fig. 4. The signal block is used to provide all the signals needed by different blocks. For example, the tracking variable signal r is fed to the block 302 in Fig. 3, buses 3b1 and 3b2 are fed to adaptation mechanisms blocks 304-1 and 304-2, respectively.

Fig. 5 is an expanded block schematic diagram of the i th element, $\hat{g}_i$, of the block 303-1 and the ij th element, $\hat{d}_{ij}$, of the block 303-2 as shown in Fig. 3, respectively. Fig. 6 is an expanded block schematic diagram of the ij th element, $\hat{c}_{ij}$, of the block 303-3 as shown in Fig. 3. It should be understood that W_{0i} , W_{kji} , W_{kij} , W_{ijk} are vectors of nodes, and $\hat{P}_{0i}$, $\hat{P}_{kij}$, $\hat{P}_{kij}$, $\hat{P}_{ijk}$ are the corresponding vectors of estimated weights in Fig. 5 and Fig. 6.

Various modifications may be made to the preferred embodiments without departing from the spirit and scope of the invention as defined in the claims.



5 Claims

Although only a few embodiments of the present invention are disclosed above, other embodiments, changes and modifications will be immediately apparent to those skilled in the art. For example, the techniques discussed above can be extended to much more general classes of systems such as multi-degrees-of-freedom platforms, electromechanical systems and systems having structured frictions. The techniques discussed above can be further extended to dynamical systems having modeling uncertainties by combining with other control techniques for robust stability. For example, the blocks 203 and 204 in Fig. 2 can be replaced by the blocks 303 and 304 in Fig. 3 for different control performance.

What is claimed is:

1. A control system for controlling a nonlinear dynamic plant having a plurality of outputs so that each of said plurality of plant outputs approaches a corresponding desired plant output, said control system having means responsive to said plurality of plant outputs and to said plurality of desired plant outputs for generating a plurality of tracking error variables, said control system comprising:

means for generating a set of tracking variables;

a first controller which is a structural network controller having a plurality of variable adaptation parameters and being responsive to said plurality of tracking variables, to said plurality of plant outputs and to said plurality of desired plant outputs, and means for generating the first control outputs for controlling said plant;

an adaptation mechanism responsive to said plurality of tracking variables and to said plurality of plant outputs for adjusting said variable adaptation parameters;

a second controller which is a PID (proportional, integral and differential) controller responsive to said plurality of tracking error variables; and

means for summing said first control outputs and said second control outputs to generate a control sum, and means for applying said control sum to said plant.



2. A control system according to claim 1 wherein said means for generating a set of tracking variables comprises (i) means for summing said plurality of desired plant outputs and said plant outputs to generate said tracking error variables; (ii) means for multiplying each of said plurality of tracking error variables by a predetermined, fixed weight value to generate the weighted tracking variables; and (iii) means for further summing each of said plurality of weighted tracking variables, each of said plurality of desired plant outputs to generate other said tracking variables.

3. A control system according to claim 1 wherein said first controller comprises a plurality of structural networks.

4. A control system according to claim 3 wherein said each of structural networks comprises a plurality of nodes, each of said plurality of nodes comprising a predetermined interpolation function, said structural network further comprising means for applying said plant outputs to each of said nodes.

5. A control system according to claim 3 wherein said structural networks further comprise means for generating a set of weighted interpolated outputs and means for generating function estimators.

6. A control system according to claim 5 wherein said means for generating a set of weighted interpolated outputs comprises means for multiplying each of said plurality of interpolated outputs by said variable adaptation parameters.

7. A control system according to claim 5 wherein said means for generating function estimators comprises means for summing said set of weighted interpolated outputs to generate the most immediate said function estimators.

8. A control system according to claim 5 wherein said means for generating function estimators further comprises (i) means for summing said set of weighted interpolated outputs to generate the intermediate function estimators; (ii) means for multiplying each of said intermediate function estimators by said plurality of plant outputs to generate a set of weighted intermediate function estimators; and (iii) means for summing said weighted intermediate function estimators for generating final said function estimators.



9. A control system according to claim 4 wherein said interpolation function is a function of said plant's own known functions.

10. A control system according to claim 1 wherein said means for generating the first control outputs comprises means for multiplying said function estimators by said set of tracking variables to generate parts of said first control outputs; said means for generating the first control outputs further comprises means for summing said parts to generate said first control outputs.

11. A control system according to claim 1 wherein said second controller is a PID controller responsive to said plurality of tracking error variables for generating second control outputs.

12. A control system according to claim 1 further comprises means for summing said first control outputs and said second control outputs to generate a control sum and means for applying said control sum to said plant.

13. A control system for controlling a nonlinear dynamic plant having a plurality of outputs so that each of said plurality of plant outputs approaches a corresponding desired plant output, said control system having means responsive to said plurality of plant outputs and to said plurality of desired plant outputs for generating a plurality of tracking variables, a first structural network controller having a plurality of variable adaptation parameters and being responsive to said plurality of tracking variables, to said plurality of plant outputs and to said plurality of desired plant outputs, and means for generating the first control outputs for controlling said plant, and an adaptation mechanism responsive to said plurality of tracking variables and to said plurality of plant outputs for adjusting said variable adaptation parameters, the improvement wherein said first controller comprises:

means for generating a set of tracking variables;

a structural network having a plurality of nodes, each of said plurality of nodes comprising an interpolation function of said plant's own known functions;



means responsive to said variable adaptation parameters for generating a set of weighted interpolation outputs, and means for generating function estimators; an adaptive mechanism for updating said variable adaptation parameters; and means for generating the first control outputs for controlling said plant;

14. In a control system, the improvement according to claim 13 wherein said means for generating a set of tracking variables comprises (i) means for multiplying each of said plurality of tracking error variables by a predetermined, fixed weight value to generate weighted tracking variables; and (ii) means for further summing each of said plurality of weighted tracking variables, each of said plurality of desired plant outputs and each of said plurality of plant outputs to generate other said tracking variables.

15. In a control system, the improvement according to claim 13 wherein said structural network further comprises means for applying said plant outputs (states or other auxiliary variables) to each of said nodes.

16. In a control system, the improvement according to claim 13 wherein said means for generating a set of weighted interpolation outputs comprises means for multiplying each of the said plurality of interpolated outputs by said variable adaptation parameters.

17. In a control system, the improvement according to claim 13 wherein said means for generating function estimators comprises means for summing said set of weighted interpolated outputs to generate the most immediate said function estimators.

18. In a control system, the improvement according to claim 13 wherein said means for generating function estimators further comprises (i) means for summing said set of weighted interpolated outputs to generate the intermediate function estimators; (ii) means for multiplying each of said intermediate function estimators by said plurality of plant outputs (the states or other auxiliary variables) to generate a set of weighted intermediate function estimators; and (iii) means for summing said



weighted intermediate function estimators for generating final said function estimators.

19. In a control system, the improvement according to claim 13 further comprising a second controller which is a PID type controller responsive to said plurality of tracking error variables for generating second control outputs.

20. In a control system, the improvement according to claim 19 further comprising a third controller having a plurality of fixed parameters and being responsive to said plurality of tracking error variables, to said plurality of plant outputs and to said plurality of desired plant outputs for generating third control outputs; and means for applying said third control outputs to said plant.

21. In a control system, the improvement according to claim 20 wherein said third controller comprises a sliding controller.

22. In a control system, the improvement according to claim 19 further comprising means for summing said first control outputs, said second control outputs and said third control outputs to generate a control sum and means for applying said control sum to said plant.

23. A control system according to claims 1 and 13 wherein said dynamic plant includes a multi-degrees-of-freedom mechanical system which can be transferred from an initial configuration to desired final configurations by at least one actuator.

24. A control system according to claims 23 wherein said mechanical system includes a robotic manipulator having at least two links connected by movable joints and said at least one actuator is a joint actuator adapted to move said links from one manipulator configuration to desired final manipulator configurations.

25. A control system constructed and arranged according to Figure 3 of the accompanying drawings, and its modifications.

26. A control system incorporating a structural network according Figure 5 of the accompanying drawings, and its modifications.



27. A control system incorporating a structural network according Figure 6 of the accompanying drawings, and its modifications.

References

- [1] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics: Control, Sensing, Vision and Intelligence, *McGraw-Hill International Editions*, Industrial Engineering Series, New York , 1987.
- [2] J.J. E. Slotine and W. Li, Applied Nonlinear Control, *Prentice Hall*, Englewood Cliffs, 1991.
- [3] S.S. Ge, Z.-L. Wang and Z.-J. Chen, "Adaptive static neural network control of robots," *Proc. of IEEE International Conference on Industrial Technology*, Guangzhou, P.R. China, 5-9, December, 1994.
- [4] R.M. Sanner and J.J.E. Slotine, "Stable Adaptive Neural Network Controller," *U.S. Patent*, No. 5,268,834, 7 December, 1993.
- [5] R.M. Sanner and J.J.E. Slotine, "Gaussian networks for direct adaptive control," *Proc. of the 1991 ACC*, pp.2153 - 2159, 1991.
- [6] C. Desoer, and M. Vidyasagar, Feedback Systems: Input-Output Properties, *Academic Press*, New York, 1975.
- [7] S.S. Ge, J.C. Allwright and C.B. Besant, "A unified adaptive control law design for rigid body robots via a Hamiltonian property," *Proc. American Control Conference*, June 1992, pp. 3136 - 3140.