INTELLIGENT CONTROLLER FOR MULTIPLE-INPUT MULTIPLE-OUTPUT SYSTEMS – PART I

JEEN LİN¹ AND RUEY-JING LIAN²,*

¹Department of Mechanical Engineering
National Taipei University of Technology
No. 1, Sec. 3, Jhongsiao E. Rd., Taipei City 10608, Taiwan
jlin@ntut.edu.tw

²Department of Management and Information Technology
Vanung University
No. 1, Wanneng Rd., Jhongli City, Toayuan County 32061, Taiwan
*Corresponding author: rjlian@vnu.edu.tw

Received February 2010; revised June 2010

ABSTRACT. A self-organizing fuzzy controller (SOFC) has been developed for control engineering applications. However, in practical applications, it is difficult to choose the values of the SOFC’s learning rate and weighting distribution appropriately to achieve reasonable control performance. In addition, the SOFC is mainly used to control single-input single-output systems. When the SOFC is applied to manipulating multiple-input multiple-output (MIMO) systems, it is hard to eliminate the dynamic coupling effects between the degrees of freedom (DOFs) of the MIMO system. To address the problems, this study developed a hybrid self-organizing fuzzy and radial basis-function neural-network controller (HSFRBNC), which applies a radial basis function neural-network (RBFN) to regulate the learning rate and weighting distribution of the SOFC to optimal values in real time, to solve the problems faced when the SOFC was applied to controlling MIMO systems. The HSFRBNC can compensate for the dynamic coupling effects between the DOFs of the MIMO system control because its learning rate and weighting distribution are adjusted by the RBFN which has a coupling weighting regulation ability of the neural-network. Stability and robustness of the HSFRBNC have been demonstrated using a state-space approach. From the simulation results of the 2-link robotic manipulator application and the experimental results of the 6-DOF robot tests, the HSFRBNC demonstrated better control performance than the SOFC.

Keywords: Self-organizing fuzzy controller, Radial basis-function neural-network, Stability and robustness, State-space approach

1. Introduction. Multiple-input multiple-output (MIMO) systems typically have dynamic coupling characteristics with complexities and nonlinearities. It has been difficult to identify a dynamic model of an MIMO system accurately and to decouple it for controller design. The model-based control strategy requires extensive computation that renders it nearly impossible to adapt correctly to the complexities of MIMO systems. Therefore, a model-free, intelligent control strategy for MIMO systems is slowly attracting interest. Fuzzy control theory has been successfully applied to many control engineering fields [1-3], however, fuzzy control strategies are designed primarily for single-input single-output (SISO) systems. Although fuzzy control strategies for manipulating MIMO systems have been developed, these strategies have not focused on overcoming the dynamic coupling effects between the degrees of freedom (DOFs) of MIMO systems. Moreover, the number of control rules and the computational burden of a fuzzy logic controller (FLC) grow exponentially with the number of variables considered.
Neural-networks have been used to control MIMO systems [4-6]; however, neural-networks’ convergence rates are too slow to compensate for the coupling variations of MIMO systems in real time. The difficulty of controlling MIMO systems lies in eliminating the dynamic coupling effects between the DOFs. Accordingly, the use of the individual fuzzy control strategy, which is designed from the viewpoint of an SISO system control, or the use of the neural-network control strategy to handle complicated and nonlinear MIMO systems in practical applications may require considerable effort or may be impossible to achieve satisfactory control performance. However, both strategies have their own advantages. Fuzzy control provides the benefit of facilitating the implementation of each DOF of a MIMO system and the neural-network learns in a superior way when an unknown dynamics model of the system is used. Thus, combining a neural-network with an FLC for manipulating MIMO systems is an interesting area of research worth investigating.

The automation which introduces learning capacities of neural-networks into fuzzy logic control systems has become a very active area of research in recent years [7-9]. Such a synergism between neural-networks and fuzzy logic control systems, integrated together into a functional system, has provided a new means of realizing intelligent systems for various applications. Gao and Er [10] designed a robust fuzzy neural-network controller for identifying and manipulating MIMO systems with uncertain dynamics characteristics. Lin et al. [11] developed a method which integrated a feedback controller and an adaptive recurrent fuzzy neural-network controller to control an F-16 flight MIMO system. Dinh and Afzulpurkar [12] presented a neuro-fuzzy inference system for modeling a complicated MIMO temperature process of a roller kiln used in ceramic tile manufacturing line. Tian et al. [13] suggested a self-organizing fuzzy neural-network controller for modulating the evaporation pressure and the superheater in an MIMO refrigeration system to improve the control performance of the system.

Most of the hybrid fuzzy-logic and neural-network control strategies make use of neural-networks to determine the membership functions which are used to design appropriate fuzzy rules of an FLC for control systems and the design of these control strategies is very complicated. In general, the problem encountered in controlling MIMO systems is that the dynamic coupling effects between the DOFs, which existed in MIMO control systems, are difficult to eliminate. To solve the problem, Li et al. [14] developed an online neural-network decoupling controller for MIMO systems. This method adopted an elitist genetic algorithm and a hybrid genetic algorithm to train the neural-network to compensate for the dynamic coupling effects between the DOFs of the MIMO system. Lygouras et al. [15] suggested a mixed fuzzy controller (MFC) for MIMO systems. The design procedure of the MFC is that an FLC is first used to manipulate each DOF of an MIMO system. Subsequently, a coupling fuzzy controller is introduced into the FLC to compensate for the dynamic coupling effects between the DOFs so as to improve the control performance of the system.

Most of the aforementioned studies, in overcoming the dynamic coupling effects between the DOFs of an MIMO system, were performed in simulation tests. These algorithms are very complex and are difficult to implement in practical applications. Although the practical application of the MFC to manipulate an MIMO system has reasonable control performance [15], it is more suitable for using in 2-DOF systems and may be unsuitable for use in a system with three or more DOFs. This is because a coupling fuzzy controller, for a control system with three or more DOFs, is arduous to design. Furthermore, it is still to difficult to design suitable fuzzy rules of the MFC for MIMO systems.

The main problem in the design of an FLC is that both the inference table and the knowledge base of the FLC, which are constructed by using an expert’s knowledge or
the experience of a skilled operator, are fixed after selection. Procyk and Mamdani [16] first proposed a self-organizing fuzzy controller (SOFC) to solve this problem. Instead of using human thinking, this strategy established fuzzy control rules by learning a system’s dynamics behavior, which simplified the procedures for the design of an FLC. Shao [17], and Zhang and Edmunds [18] further improved learning methods to simplify the design of the SOFC. The improved learning scheme was based on a performance decision table proposed by Procyk and Mamdani [16]. However, the determination of the performance decision table is as difficult as the design of a fuzzy rule table. To overcome the problem, Yang [19], Huang and Lee [20], Lian et al. [21] and Lin and Lian [3] applied the output error and the error change of the system to establish a learning algorithm that can adjust the SOFC’s linguistic fuzzy rule table directly, so that such can be generated without any initial fuzzy rules. The SOFC eliminates the difficulty of designing an FLC with appropriate membership functions and fuzzy rules.

Clearly, the SOFC [3,19-21] has superior learning ability for handling complicated and nonlinear systems. However, both the learning rate and weighting distribution in the SOFC must be carefully chosen and are fixed once these parameters have been selected. Unfortunately, inappropriate selection of either the weighting distribution or the learning rate (or both) in the SOFC will substantially affect the system’s output response, and may result in the system becoming unstable. In addition, the design of the SOFC involves in controlling SISO systems, so it cannot eliminate the dynamic coupling effects between the DOFs of MIMO systems. Therefore, this study develops a hybrid self-organizing fuzzy and radial basis-function neural-network controller (HSFRBNC) for MIMO systems to solve the problems faced when the SOFC is applied. The HSFRBNC uses a radial basis function neural-network (RBFN) algorithm [22-24] to regulate the parameters (the learning rate and the weighting distribution) of the SOFC in real time to suitable values instead of using results obtained by trial and error from experiments [3,20,21]. Since the regulation of the parameters of the SOFC are dependent on the system’s output error and error change for each DOF in an MIMO system, the HSFRBNC introduces the RBFN, which has a coupling weighting regulation capability of the neural-network, to adjust these parameters to perfect values so that it can compensate for the dynamic coupling effects between the DOFs of the MIMO system control.

The stability of a control system is an important factor that must be addressed in determining the availability of a controller. Fuzzy control systems are essentially nonlinear systems, so their stability analyses are based primarily on classical nonlinear dynamics theory [25]. In a stability analysis, the control plant is a non-fuzzy system that is controlled by an FLC. The FLC is regarded as a special case of the general nonlinear controllers. The stability of a closed-loop non-fuzzy system, with an FLC, can then be analyzed by applying classical nonlinear dynamics theory.

Garcia-Cerezo et al. [26] were the first to employ a nonlinear function to represent an FLC. In their work, the ideas of Tong [27] were applied to characterizing a closed-loop fuzzy control system. The nonlinear function of an FLC can be determined from the relation matrix of the discrete form of a fuzzy relationship rule. Tong [28] demonstrated that the relation matrix depends only on the set of fuzzy rules. Therefore, the control performance of a closed-loop system can be improved by modifying these fuzzy rules.

Braae and Rutherford [29,30] incorporated the linguistic trajectory concept into the stability analysis of a fuzzy control system and applied the technique of phase plane analysis to investigate the stability problems of an FLC. Aracil et al. [31] examined the stability of a knowledge-based expert control system, the relative stability of the equilibrium point at the origin, and whether the system existed at other globally stable points. According to the aforementioned studies, Aracil et al. [31] developed two stability
indices for designing and analyzing fuzzy control systems. Wang and Liu [32] proposed a homogenous polynomial matrix function to design a suitable controller and select an appropriate Lyapunov function to guarantee the stabilities of the discrete-time and the continues-time fuzzy system. Zhang and Hu [1], Wang et al. [2] and Mendez-Monroy and Benitez-Perez [33] employed the Lyapunov theorem for analyzing the stabilities of their respective proposed fuzzy control systems. Lee and Pan [34] and Shoorehahdeli et al. [35] also used the Lyapunov theorem to demonstrate the stabilities of an adaptive network-based fuzzy inference system and an interval-valued neural fuzzy inference system with asymmetric membership functions, respectively.

Historically, the stability analysis of an FLC mainly focuses on the control of SISO systems. This study develops an HSFRBNC, which differs from the above-mentioned control strategy, for MIMO systems. A two-link robot with a complex dynamics model is presented to determine the applicability and stability of the proposed HSFRBNC in controlling MIMO systems. In evaluating the presented system, this study demonstrates the implementation and stability of the HSFRBNC by numerical simulations. To further evaluate the feasible of the proposed HSFRBNC in practical applications, this study applies the HSFRBNC to control a 6-DOF robotic manipulator in real time to determine experimentally the performance of the joint trajectory tracing for a square-path control of the robot.

2. Controller Design. Since an MIMO system may have many inputs and outputs, its input and output vectors generally have different dimensions, so that the controller design becomes very complicated. To simplify the design of the proposed controller, this study develops an HSFRBNC for MIMO systems in which the dimensions of the input vector and the output vector are regarded as the same. Figure 1 shows the control block diagram of the HSFRBNC for an MIMO system.

2.1. Self-organizing fuzzy algorithm. An SOFC has been developed for many applications [3,19-21]. It has superior learning capacity in real time, starting from an empty fuzzy rule table and without the need of a mathematical model of the system for the controller development to control complicated and nonlinear systems. The self-organizing part is introduced into an FLC to constitute an SOFC, as depicted in Figure 1, and consists of three steps: system performance measure, model estimation and rule modification. The system performance measure is important for achieving successful learning.
algorithms. Two physical features, such as the output error and the error change of the system, are measured as performance indices in a decision table. The measurement of such indices is similar to establishing a fuzzy rule table. A system model is estimated to determine the relationship between the output response and the control input of the system. The performance measure is then applied to determining the correction value of each fuzzy rule by applying the estimation model of the system. However, a suitable performance decision table is arduous to organize for each control system. Thus, this study developed a real-time linguistic SOFC, in which two parameters (learning rate and weighting distribution) are applied to establishing a performance measure function, instead of a performance decision table.

The rule table of the proposed SOFC includes only a modification of the original fuzzy rules; the correction value of each fuzzy rule is introduced into each original fuzzy rule to generate a new control rule. This approach overcomes the expansive defect of the database in the scheme of Procyk and Mamdani [16] and reduces time consumption. Furthermore, the dynamic system output may be specified by precise design parameters. The dynamic behavior of the system can be represented using an autoregressive moving average model [36]:

\[ y(k) = A(z)y(k-1) + Mu(k-d) + B(z)u(k-d-1) \]  

where

\[ A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n} \]
\[ B(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-(m-d)} \]

and \( z \) represents a forward shift operator; \( d \) is the system delay and \( M \) is the direct forward system gain in the control system. The values of \( n \) and \( m \) depend on the dynamic characteristics of the system. It is arduous to estimate the aforementioned parameters for a dynamic system with complexity and nonlinearity. To eliminate the problem, an FLC, which is one of the control methods for manipulating complex and nonlinear systems, has been proposed to control the above-mentioned system. It neither requires a system model nor depends on either a precise mathematical model or system parameter. Hence, the following controller design includes none of these. If the system is excited with a different control input, \( u'(k-d) \), at a \( k-d \)-step sampling interval while the system’s past state is unchanged, then the system output, \( y'(k) \), at a \( k \)-step sampling interval can be described as

\[ y'(k) = A(z)y(k-1) + Mu'(k-d) + B(z)u(k-d-1) \]  

The correction of the control input, \( \Delta \pi(k) \), will yield a system’s output deviation, \( \Delta y(k) \). If \( \Delta y(k) \) is small, then the relationship between the control input’s correction and the corresponding system’s output deviation is

\[ \Delta y(k) = y'(k) - y(k) = M [u'(k-d) - u(k-d)] = M \Delta \pi(k) \]  

Therefore, the difference of the system’s output deviation, \( \Delta \hat{y}(k) \), can be described as

\[ \Delta \hat{y}(k) = \frac{\Delta y(k)}{T} = \frac{M}{T} \Delta \pi(k) \]  

where \( T \) represents a sampling interval that is small. If a system at \( k \)-step sampling interval has a system’s output deviation, \( \Delta y(k) \), and its difference, \( \Delta \hat{y}(k) \), then the corrections of the corresponding theoretical control input are \( \Delta \pi_c(k) \) and \( \Delta \pi_{cc}(k) \), respectively. Referring to Equation (4), these corrections can be determined as

\[ \Delta \pi_c(k) = \frac{\Delta y(k)}{M} \quad \text{and} \quad \Delta \pi_{cc}(k) = \frac{T \Delta \hat{y}(k)}{M} \]  

\[ \Delta \pi_{cc}(k) = \frac{T \Delta \hat{y}(k)}{M} \]
The system has only one control input, $u(k)$, so the correction of the control input must be an appropriate combination of the aforementioned two terms. The following form for general cases [19] is used:

$$\Delta u(k) = (1 - \varsigma)\Delta u_e(k) + \varsigma \Delta u_{ec}(k)$$

(6)

where $\varsigma$ is a design parameter, which is restricted to lie in (0, 1), and represents the weighting distribution between $\Delta u_e(k)$ and $\Delta u_{ec}(k)$. If the system output, $y(k)$, differs greatly from the desired output, $y_d(k)$, then a value of $y'(k)$ between $y(k)$ and $y_d(k)$ should be chosen in the design. The system output, $y(k)$, will then slowly approach $y_d(k)$ at a learning rate ($\gamma$) [19]:

$$y'(k) = (1 - \gamma)y(k) + \gamma y_d(k), \quad 0 < \gamma < 1$$

(7)

The system’s output deviation, $\Delta y(k)$, and its difference, $\Delta \dot{y}(k)$, thus become

$$\Delta y(k) = y'(k) - y(k) = \gamma [y_d(k) - y(k)] = \gamma e(k)$$

(8)

$$\Delta \dot{y}(k) = \gamma \dot{e}(k) = \gamma \left[ \frac{e(k) - e(k - 1)}{T} \right] = \frac{\gamma}{T} e(k)$$

(9)

Equations (5), (8) and (9) are introduced into Equation (6), which yields the correction of the control input,

$$\Delta u(k) = \frac{\gamma}{M} [(1 - \varsigma)e(k) + \varsigma ec(k)]$$

(10)

The system’s output error, $e(k)$, and error change, $ec(k)$, are divided into seven fuzzy subsets with integer values from −6 to +6. In each control step, the fuzzy input variables, namely the system’s output error and error change, excite two fuzzy subsets of the $E$ and $EC$ universe of discourse, respectively. The control input, $u(k)$, is derived from the fuzzy rule inference, so the four fuzzy rules will be modified in each control step. The correction value of each fuzzy rule is proportional to the excitation strength of each fuzzy rule, $w$. The $w$ is represented as a triangular membership function and is calculated using a linear interpolation algorithm [19]. Then, the control input of the $i$th fuzzy rule is

$$\bar{u}_i(k + 1) = \bar{u}_i(k) + \Delta \bar{u}_i(k)$$

$$= \bar{u}_i(k) + w_{ei} w_{eic} \frac{\gamma}{M} \times [(1 - \varsigma)e(k) + \varsigma ec(k)]$$

(11)

For a multiple DOF self-organizing fuzzy control system, Equation (11) can be modified as

$$\bar{u}_{li}(k + 1) = \bar{u}_{li}(k) + \Delta \bar{u}_{li}(k)$$

$$= \bar{u}_{li}(k) + w_{ei} w_{eic} \frac{\gamma_i}{M} \times [(1 - \varsigma_i)e_i(k) + \varsigma_i ec_i(k)]$$

(12)

where $i$ indicates the DOF of the control system.

The term, $\frac{\gamma_i}{M}$, in the aforementioned equation can be regarded as the correction weighting for the control input of the $i$th fuzzy rule of each DOF of the control system. This study sets $M$ to 1 to eliminate the identification procedure and reduce the computational time required during implementation. The correction weighting is determined only by the learning rate, $\gamma_i$. A large value of $\gamma_i$ will introduce a large rule correction and result in the system’s output oscillation. This parameter for the control system influences only the transient response and does not affect the steady-state performance. This was demonstrated by Huang and Lee [20] through their experiments.
2.2. RBFN algorithm. The RBFN has been extensively applied to representing the nonlinear mappings between inputs and outputs of nonlinear control systems [22-24]. Lu and Basar [22] designed an RBFN algorithm for identifying the system model. Huang et al. [23] suggested an RBFN sliding mode controller for a dynamic absorber system to suppress the vibration amplitude of the system due to external disturbances. Most of the RBFNs applied to applications of model identification for dynamic systems with complexities and nonlinearities. This study used an RBFN algorithm to adjust, in real time, both the learning rates and the weighting distributions of the SOFC to appropriate values for MIMO system control.

The RBFN has a feed forward structure consisting of a single hidden of $s$ locally tuned neurons, which are fully interconnected to an output layer of $n$ linear neurons. All hidden neurons simultaneously receive the $m$-dimensional input vector $X$ (Figure 2). After the hidden layer receives the data from the input layer, the Gaussian basis function in the hidden layer is used to transform the data nonlinearly, and then the function responses are linearly combined to construct the data of the output layer. Figure 2 shows a structural diagram of the RBFN. The algorithm of the RBFN can be described as

$$ O_j = \sum_{i=1}^{s} w_{ij} h_i = \sum_{i=1}^{s} w_{ij} R(\|X - Z_i\|), \quad j = 1, 2, \ldots, n \quad (13) $$

and

$$ h_i = R(\|X - Z_i\|), \quad i = 1, 2, \ldots, s $$

$$ R(\|X - Z_i\|) = \exp \left( -\frac{\|X - Z_i\|^2}{2\sigma_i^2} \right), \quad i = 1, 2, \ldots, s $$

$$ X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}^T $$

$$ Z_i = \begin{bmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{im} \end{bmatrix}^T, \quad i = 1, 2, \ldots, s $$

where $w_{ij}$ is a weighting, representing the strength of the connection from the neuron $h_i$ to the output neuron $O_j$; the $j$th output neuron of the output layer is a nonlinear function of the output value of the hidden layer neurons. The index $i$ runs over all connections to the $j$th neuron. $Z_i$ and $\sigma_i$ are the central position and the standard deviation of the $i$th neuron receptive field, respectively. The norm, $\| \cdot \|$, is the 2-dimensional Euclidean space; $R(\cdot)$ is a Gaussian function.

![Figure 2. Radial basis function neural-network](image-url)
The desired output is $T^*_j$, and the corresponding output of the output layer is $O_j$. Then, a cost function of the RBFN is defined as

$$ E = \frac{1}{2} \sum_{j=1}^{n} (T^*_j - O_j)^2 = \frac{1}{2} \sum_{j=1}^{n} v^2_j \quad (14) $$

Most of the neural-networks, such as a back-propagation neural-network and an RBFN applied by Huang et al. [23], used the steepest descent method to minimize their cost functions so as to determine the correction values of the weightings in the neural-networks. However, the convergence rate for the correction value of the weighting in the neural-network is slow when this method is applied. To solve the problem, this study proposed a Levenberg-Marquardt (LM) algorithm [37-39], instead of the steepest descent method, to minimize the cost function of the RBFN. The LM algorithm will accelerate the convergence rate for the correction value of the weighting in the RBFN.

The LM algorithm, for a neural-network, is one of the most widely used methods for optimization. It outperforms the simple gradient descent and other conjugate gradient methods in a wide variety of applications. The LM method combines the gradient descent with the Gauss-Newton iteration in a learning algorithm, thereby providing faster convergence than the steepest descent method alone in determining the correction value of the weighting. This study used the LM method to minimize the cost function of the RBFN so as to calculate the correction value of the weighting in the RBFN.

A generalized form for the LM algorithm has been proposed [37-39] as

$$ \Delta x_k = -[J^T(x_k)J(x_k) + \mu_k I]^{-1}J^T(x_k)v(x_k) \quad (15) $$

where $\nabla E(x)|_{x=x_k} = J^T(x_k)v(x_k)$. $J(x_k)$ is the Jacobian matrix of $v(x)$ at $x_k$; $E(x)$ is an objective function, $v(x_k)$ is a residual error, $\mu_k$ is a parameter, and $I$ is an identity matrix.

The LM algorithm has a very useful feature. As $\mu_k$ is increased, it approaches the steepest descent algorithm with small learning rate.

$$ x_{k+1} = x_k - \frac{1}{\mu_k}J^T(x_k)v(x_k) = x_k - \frac{1}{\mu_k}\nabla E(x) \quad \text{for large } \mu_k \quad (16) $$

when $\mu_k$ is decreased to zero, the LM algorithm becomes a Gauss-Newton method.

The LM algorithm begins from a small $\mu_k$ value (such as $\mu_k = 0.01$). If a step does not yield a smaller value for $E(x)$, the step will be repeated with $\mu_k$ multiplied by some factor $\eta > 1$ (such as $\eta = 10$). Ultimately, $E(x)$ should be reduced because a small step is taken in the direction of the steepest descent. If a step does not produce a smaller value for $E(x)$, $\mu_k$ will be divided by $\eta$ for the next step. This is done so that the LM algorithm will approach the Gauss-Newton method which should provide faster convergence [37].

According to Equations (13)-(16), the weighting correction ($\Delta \overline{w}_{ij}$), central position correction ($\Delta Z_i$), and standard deviation correction ($\Delta \sigma_i$) of the RBFN could be separately determined as

$$ \Delta \overline{w}_{ij} = -\left[J(\overline{w}_{ij})^TJ(\overline{w}_{ij}) + \mu_k I\right]^{-1}J^T(\overline{w}_{ij})v_j(\overline{w}_{ij}) \quad (17) $$

$$ \Delta Z_i = -\left[J(Z_i)^TJ(Z_i) + \mu_k I\right]^{-1}J^T(Z_i)v_j(Z_i) \quad (18) $$

$$ \Delta \sigma_i = -\left[J(\sigma_i)^TJ(\sigma_i) + \mu_k I\right]^{-1}J^T(\sigma_i)v_j(\sigma_i) \quad (19) $$

The output value of the RBFN has to be maintained within an appropriate range for this control strategy implementation, so a nonlinear transformation layer is introduced between the hidden layer and the output layer to regulate the output value. The procedure for determining the weighting correction $\Delta \overline{w}_{ij}$ of the nonlinear transformation layer is
similar to the procedure for determining that of the RBFN between the hidden layer and the output layer.

To derive the weighting correction of the aforementioned RBFN, a batch learning method [40] was employed and an objective function of the RBFN for the step $p$ was defined as

$$E_p^* = \frac{1}{2} \sum_j (r_{pj} - y_{pj})^2 = \frac{1}{2} \sum_j e^2_{pj}$$

where $r_{pj}$ and $y_{pj}$ express the desired set-point and the system output in the step $p$, respectively. When $E_p^*$ approaches zero, the mapping between inputs and outputs of each step $p$ is realized. Similar to the derived process of Equations (17)-(19), the weighting correction, central position correction, and standard deviation correction of the RBFN in the step $p$ could be individually determined as

$$\Delta \bar{w}_{ij} = - \sum_p \left[ J_p(\bar{w}_{ij})^T J_p(\bar{w}_{ij}) + \mu_{kp} I \right]^{-1} J_p^T(\bar{w}_{ij}) e_{pj}(\bar{w}_{ij})$$

$$\Delta Z_i = - \sum_p \left[ J_p(Z_i)^T J_p(Z_i) + \mu_{kp} I \right]^{-1} J_p^T(Z_i) e_{pj}(Z_i)$$

$$\Delta \sigma_i = - \sum_p \left[ J_p(\sigma_i)^T J_p(\sigma_i) + \mu_{kp} I \right]^{-1} J_p^T(\sigma_i) e_{pj}(\sigma_i)$$

If the input data at $k$-step is $X(k)$, the updated rules for the aforementioned corrections could be separately described as

$$\bar{w}_{ij}(k + 1) = \bar{w}_{ij}(k) + \Delta \bar{w}_{ij}(k)$$

$$Z_i(k + 1) = Z_i(k) + \Delta Z_i(k)$$

$$\sigma_i(k + 1) = \sigma_i(k) + \Delta \sigma_i(k)$$

2.3. Design of an HSFRBNC. The HSFRBNC is composed of a self-organizing algorithm, an FLC, and an RBFN. The self-organizing learning algorithm and the RBFN were presented in the preceding section. Design procedures of the FLC and HSFRBNC are described here.

The structure of an FLC design consists of the following: the definition of input-output fuzzy variables, decision-making related to fuzzy control rules, fuzzy inference logic and defuzzification. Figure 1 presents an HSFRBNC for an MIMO system. From Figure 1, the control variables of the system were defined as

$$e_i(k) = r_i(k) - y_i(k)$$

$$ee_i(k) = e_i(k) - e_i(k - 1)$$

where $e_i(k)$ and $ee_i(k)$ are the system’s output error and error change on $k$-step sampling interval, respectively; $r_i(k)$ and $y_i(k)$ represent the system’s reference input and output response on $k$-step sampling interval, respectively. The subscript $i$ is employed to represent the DOF of the system. A triangular membership function, depicted in Figure 3, is employed to convert these input variables ($e_i(k)$ and $ee_i(k)$) and the output variable ($u_i(k)$) into linguistic control variables (NB, NM, . . . , PB), where $\beta^j_i$ is a scaling factor. The subscript $i$ has been described previously. The superscript ($j = 1, 2, 3$) is employed to express the system’s output error, error change and control input, respectively.

This study made use of the state evaluation fuzzy-control rules [41] for manipulating the complicated and nonlinear MIMO system. If the fuzzy-inference logic used the Max - Min product composition ($\land$) [41] to operate the fuzzy rules, the change of its antecedent suitability would not be smooth when $\land$ was applied. To address this problem, the $\land$
operation was replaced by an algebraic product [42]. Finally, this study used the height method [41] to defuzzify the output variables to gain accurate objectives for controlling the system. The aforementioned design process yields the following control input for the hybrid self-organizing fuzzy and radial basis-function neural-network control system,

\[ u_i(k) = u_i(k-1) + \Delta u_i(k) \]  

(29)

where \( \Delta u_i(k) \) indicates the control-input increment of the system on \( k \)-step sampling interval. \( u_i(k) \) and \( u_i(k-1) \) represent the system’s control inputs on \( k \)-step and \( k-1 \)-step sampling intervals, respectively.

Although an SOFC has learning capability in real time, it still cannot solve the problem of selecting its appropriate learning rate and weighting distribution for the control of complex and nonlinear MIMO systems. The SOFC can modify an inappropriate fuzzy rule table continuously into a satisfactory fuzzy rule table by using the aforementioned designed learning algorithm. However, the weighting distribution, \( \xi_i \), and the learning rate, \( \gamma_i \), affect the dynamic behavior of the system response in the design of the SOFC. In general, the weighting distribution of the SOFC for controlling each DOF of the MIMO system is chosen to be 0.5 to avoid bias toward \( \Delta p_{ii}(k) \) or \( \Delta p_{ee}(k) \). This choice does not guarantee that such a selected parameter is appropriate in the design of an SOFC for controlling each DOF of the MIMO system. Additionally, choosing a suitable learning rate of the SOFC for controlling each DOF of the MIMO system is intricate because if a high learning rate exists (in the regulation of the self-organizing fuzzy learning algorithm), then the fuzzy rule table will be modified excessively. Moreover, the system’s control command will vary over such a large range that the system’s output response yields oscillatory phenomena. At this point, the system is likely to become unstable. In contrast, if the learning rate of the SOFC for controlling each DOF of the MIMO system is small, the system’s output response will be slow. Even though learning time is increased, the system’s control performance is unlikely to show improvement.

An appropriate learning rate of the SOFC for controlling each DOF of the MIMO system can be obtained experimentally by trial and error, but such a learning rate remains fixed after it has been selected. Therefore, this method creates a problem of limiting the dynamic behavior of the system output. This is because the SOFC is unable to specify an appropriate parameter for controlling each DOF of the MIMO system, in real time, according to the system’s output response. To eliminate the problem faced by the selection of appropriate parameters (\( \gamma_i \) and \( \xi_i \)), this study employed an RBFN to determine appropriate parameters of the SOFC in real time, instead of using the fixed parameters in the case where the SOFC is used, to improve the control performance of the MIMO system.
3. Stability Analysis of Fuzzy Control Systems. Although the state-space approach has been effectively employed to analyze the behavior of simple dynamic systems, it cannot be used directly to analyze the stability of fuzzy control systems. The geometric method [43] has been developed for analyzing the stability of a fuzzy control system, based on the relationship between the vector fields of the plant and controller. Dynamic system qualitative analysis [25] indicates that the stability and robustness indices of a system can be defined using a geometric method. These indices can be used not only to evaluate the stability and robustness of a closed-loop control system but also to describe dynamic behavior.

The stability analysis of an FLC on a fuzzy phase plane [29,30] requires that the relationship between the rules and the state-space concerning dynamic system behavior to be under control. This relationship is based on the relative influence of each fuzzy rule implemented during the control action of an FLC. Aracil et al. [43] developed a geometric interpretation of a state map based on the vector fields of the plant and the FLC rule base.

A closed-loop fuzzy control system can be described as

\[
\frac{dx}{dt} = f(x) + b \cdot u = \phi(x)
\]

where \(f(x)\) is a nonlinear function that represents the plant’s dynamics with \(f(0) = 0\); \(x\) and \(b\) are the state and control coefficient vectors of dimension \(n\), respectively; \(u\) is the scalar control variable, and \(\phi(x)\) is a nonlinear function that represents the FLC with \(\phi(0) = 0\) [41]. The dynamic behavior of the control system depends on the characteristics of \(f(x)\) and \(\phi(x)\). The direction of the vector field concerning the FLC is determined by the coefficients of \(b\), and the magnitude is defined as \(b \cdot \phi(x)\) [31].

According to the state-space approach, the vector field concerning \(b \cdot \phi(x)\) tries to turn the system’s trajectory into a switching curve \(\phi(x) = 0\). Then, the plant component of the vector field associated with a stable feedback control system takes over the control influence to bring the trajectories into an equilibrium point, which will guarantee that the system has the Lyapunov stability at an equilibrium point [41,43].

As a typical example, the case \(b = 1\) in Equation (30) is considered. Two stability indices [31] are introduced more completely to represent the stability of a one-dimensional fuzzy control system, based on the state-space approach.

\[
I_1 = -\left(\phi'(0) + f'(0)\right) \quad (31)
\]

\[
I_2 = \min_{C'} \left|\phi(x) + f(x)\right| \quad (32)
\]

The \(I_1\) is a stability index, which is a measure of the robustness of a system from instability at the origin. The \(I_2\) is a relative stability index, which is a measure of the relative stability of a control system. From Equation (32), the \(I_2\) could be measured using a geometric approach [43]. It is relative to a horizontal line passing through the origin of the system state to be measured. Since \(\left|\phi(x) + f(x)\right|\) has a zero minimum at the origin, a particular region \(C\) around the origin should be excluded [31,41]. Therefore, \(C'\) in Equation (32) is the complement of the particular region \(C\) around the origin.

In a two-dimensional case, the stability analysis must be performed to study the probability of static bifurcation and Hopf bifurcation [31,41]. Aracil et al. [31] developed two stability indices (\(I_1\) and \(I_1^*\)), which specify whether these bifurcations exist.

\[
I_1 = \det(J) \quad (33)
\]

\[
I_1^* = -\text{trace}(J)
\]
where $J$ is the Jacobian matrix of $f(x)$ at $x_0$. If the linearized system is stable, then the Lyapunov criterion will guarantee system stability at the origin of a nonlinear model [41]. If the $I_1$ is positive, then the system will not cause static bifurcation. If the $I^*_1$ is positive, then the control system will not exhibit Hopf bifurcation. Hence, when both the $I_1$ and $I^*_1$ are positive, the two-dimensional fuzzy control system is stable.

To evaluate the relative stability of a two-dimensional fuzzy control system, the auxiliary subspace [31] was defined to calculate the relative stability value.

$$\frac{f_1(x_1, x_2)}{b_1} = \frac{f_2(x_1, x_2)}{b_2}$$

(34)

The relative stability, $I_2$, could be defined as the minimum distance between components of the plant and the controller, and this distance is calculated in the auxiliary subspace except the particular region $C$ around the origin.

$$I_2 = \min_{C^*} |f(x) + b \cdot \phi(x)|$$

(35)

If $J$ is the Jacobian matrix of a nonlinear system around the origin and its corresponding characteristic polynomial is

$$P(s) = s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n$$

then the Hopf bifurcation unstable condition [44] will be equivalent to

$$\det(H_{n-1}) = 0$$

(37)

where $H_{n-1}$ is the minor principal of order $n - 1$ of the Hurwitz matrix $H_n$. The matrices $H_n$ and $H_{n-1}$ are defined as

$$H_n = \begin{bmatrix} a_1 & a_3 & a_5 & \cdots \\ 1 & a_2 & a_4 & \cdots \\ 0 & a_1 & a_3 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & a_n \end{bmatrix}, \quad H_{n-1} = \begin{bmatrix} a_1 & a_3 & a_5 & \cdots \\ 1 & a_2 & a_4 & \cdots \\ 0 & a_1 & a_3 & \cdots \\ \cdots & \cdots & \cdots & a_n \end{bmatrix}$$

(38)

Hence, the $I^*_1$ is defined as

$$I^*_1 = \det(H_{n-1})$$

(39)

and Ollero et al. [44] also showed that

$$I_1 I^*_1 = \det(H_n)$$

(40)

The generalization of the $I_2$ for the case $n > 2$ is straightforward. In this case, the one-dimensional auxiliary subspace will be given by

$$\frac{f_1(x_1, x_2, \ldots, x_n)}{b_1} = \frac{f_2(x_1, x_2, \ldots, x_n)}{b_2} = \cdots = \frac{f_n(x_1, x_2, \ldots, x_n)}{b_n}$$

(41)

The remarkable fact is that for every $n$, the auxiliary subspace is one-dimensional if the dimension of $u$ is one; thus, Equation (41) can apply only on an SISO control system to determine the suitable auxiliary subspace for evaluating the system’s relative stability.

For an MIMO control system, the state-space can be described as

$$\frac{dx}{dt} = f(x) + B \cdot U$$

$$U = \phi(x)$$

(42)
where \( \dim(x) = n \), \( \dim(B) = nm \), \( f(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \) and \( \phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_m(x)) \). The auxiliary subspace [31] can be defined by

\[
\frac{f_1(x)}{\sum_{j=1}^{m} b_{1j} \phi_j(x)} = \frac{f_2(x)}{\sum_{j=1}^{m} b_{2j} \phi_j(x)} = \ldots = \frac{f_n(x)}{\sum_{j=1}^{m} b_{nj} \phi_j(x)} \tag{43}
\]

For an MIMO system, the relative stability index, \( I_2 \), in the auxiliary subspace can be defined as

\[
I_2 = \min_{C'} g(x) + B \cdot \phi(x) \tag{44}
\]

where \( g(x) \) represents the vector field of the plant component in the auxiliary subspace [44]. According to the foregoing analysis, this state-space approach for analyzing the stability of a fuzzy control system can be expanded to every MIMO system.

4. Conclusion. This study has successfully developed an HSFRBNC for MIMO systems. Stability analysis of fuzzy control systems concerning the HSFRBNC has also been described. The HSFRBNC uses an RBFN, which has the coupling weighting regulation ability of the neural-network, to adjust both the learning rate and the weighting distribution of the SOFC in real time to perfect values for improving the control performance of the MIMO system. The HSFRBNC not only eliminates the need to establish proper parameters by trial and error which was done for the original design of using the SOFC, but also compensates for the dynamic coupling effects between the DOFs of the MIMO system. To demonstrate the applicability of the proposed HSFRBNC, simulation and experimental tests for the control of robotic systems have been performed in Part II of this paper [45].

REFERENCES


