Correspondence

Computational-Complexity Reduction for Neural Network Algorithms

ALLON GUEZ, MOSHE KAM, MEMBER, IEEE, AND JAMES L. EILBERT, MEMBER, IEEE

Abstract — **An important class of neural models is described as a set of coupled nonlinear differential equations with state variables corresponding to the axon hillock potential of neurons. Through a nonlinear transformation, these models can be converted to an equivalent system of differential equations whose state variables correspond to firing rates. The new firing rate formulation has certain computational advantages over the potential formulation of the model. The computational and storage burdens per cycle in simulations are reduced, and the resulting equations become quasi-linear in a large significant subset of the state space. Moreover, the dynamic range of the state space is bounded, alleviating the numerical stability problems in network simulation. These advantages are demonstrated through an example, using our model for the "neural" solution to the traveling salesman problem proposed by Hopfield and Tank.**

I. Introduction

A neural network is a set of N simple "artificial neurons" that are connected to each other and pass simple messages. Information processing by an artificial neuron or node follows the same sequence of steps in essentially all of the models found in the literature (e.g., [2], [4], [7], [8], [11], [13], [19]). Starting at the input or "dendrite" end of the node, the steps are as follows.

1) Each input to the node is gated by a permanent or slowly varying connection strength, leading to a post-synaptic potential (PSP).

2) The PSP's are combined, usually by simple summation, resulting in an "axon hillock potential" or simply the node's potential. In some cases, the sum of PSP's is gated by a normalization factor. There may also be a decay rate that affects the variation of the potential.

3) The potential is converted, through a sigmoidal or probablistic transfer function [10], leading to an "instantaneous firing rate."

4) The firing rate is delivered to the nodes connected to the output end of the original node, supplying these targets with input.

In characterizing the state of a neural network, either the *potential* or the *firing rate* of each node must be specified. The conversion steps 1–4 lead directly to an equation of motion for the network. The variation in potential or firing rate is either described by a differential equation, a difference equation, or as a probability of firing at any particular time [1]. In the present correspondence, we use the differential equation formulation for the equations of motion (e.g., (1) below, which uses potential as the state variable).

In the literature, both potential and firing rates have been used as state variables. Many of the early neural models, such as Adaline [18] and perceptrons [16], used firing rates as the state variables. The reason for this choice was probably the biological studies in the 1950's by Hodgkin and Huxley [10] elucidating the mechanism of the action potential, and the work of Eccles [14] on the role of the synapse. However, several authors, notably Pribram and Grossberg, have later made specific biological and computational arguments in favor of using the potential as the state variable. Pribram's interest in holography as an analogy to visual recognition [15] led him to consider state variables that could reflect the interaction of neural signals—it is the potentials that combine the effects of inputs from different neural sources. Another advantage of the potential representation is that potentials can be positive or negative, and can therefore completely determine the evolution of the system based on a single time point. In addition, it is the potentials that decay over time, so decay rates have a simple linear form. Finally, potentials lead to simpler governing equations for the system, since the effects of internal input, external inputs and decay rate simply add [5]. Grossberg [3] also points out that a sigmoidal function applied to the potential leads to noise suppression.

While the potential representation has its advantages, there are a number of advantages to working with firing rates. One advantage is that the firing rates contain the information that can be communicated to other components of the neural network. The potential can only be communicated after it is transformed. In this correspondence, we will show that working with the firing rate has a major computational advantage over use of the potential, as in the popular models of Cohen–Grossberg [4] and Hopfield [7]. In spite of the popularity of these models and their usefulness for some novel applications (e.g., [8], [17]) analysis is difficult. Numerical methods are needed to get solutions, and due to the complexity of the equations and the coupling between them, simulations are slow to converge, very sensitive to initial conditions, and often numerically unstable.

The neural network model is presented in Section II. In Section III, we perform the transformation that replaces the potentials by the firing rates as state variables. We then discuss the computational advantages of the resulting model. In Section IV we exploit these advantages to facilitate a better computation of the recently proposed "neural" solution for the traveling salesman problem [8].

II. The Neural Network Model

We refer to a popular model of continuous, interconnected N nodes (e.g., [4], [7]). Each node is assigned a potential, $x_i(t)$, $i=1,\cdots,N$ as its state variable. Each node receives external input I_i, and internal inputs from other nodes in the form of a normalized weighted sum of firing rates $b(x_i)[\Sigma T_{ij} g_j(x_j)]$ where $g_i(\cdot)$ is a monotonically increasing bounded function converting potential to firing rate). The equations of motion are

$$\frac{dx(t)}{dt} = -Ax(t) + b\{g[x(t)]\}Tg[x(t)] + I \qquad (1a)$$

or

$$\frac{dx_i}{dt} = \left(-a_i x_i + b_{ii}\sum_{j=1}^{N} T_{ij} g_j(x_j) + I_i\right) \qquad (1b)$$

where

$$x(t) = [x_1(t), x_2(t), \cdots, x_N(t)]^T \in R^N \quad \text{(soma potentials)}$$

$$g[x(t)] = [g_1(x_1(t)), g_2(x_2(t)), \cdots, g_N(x_N(t))]^T,$$

Manuscript received August 10, 1987; revised November 11, 1988.

A. Guez and M. Kam are with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104.

J. L. Eilbert was with the Department of Mathematics and Computer Science, Drexel University, Philadelphia PA and is now with the Grumman Corporation, Bethpage, Long Island, NY 11714-3586. This work was supported in part by the National Science Foundation under Grant DMCE 8505235 and IRI 8810186.

IEEE Log Number 8825967.

0018-9472/89/0300-0409$01.00 ©1989 IEEE

where

$g_i(x_i(t))$	differentiable monotonically increasing bounded function, representing the short-term average firing rate;
$b\{g[x(t)]\}$	diagonal matrix $b_{ii} = b_{ii}[g_i(x_i(t)]$ representing normalization of the input with respect to the firing rate of the receiving neuron;
I	$(I_1, I_2, \cdots, I_N)^T \in R^N$, a constant vector of external inputs;
A	constant $N \times N$ diagonal matrix: $\text{diag}(a_1, a_2, \cdots, a_n)$;
a_i	positive real number, representing the natural frequency of the potential decay of the ith node;
T	constant $N \times N$ matrix (not necessarily symmetric);
T_{ij}	synaptic efficacy of the jth node potential, transmitted to the ith node.

The dynamic model (1) is highly nonlinear, coupled and of a high dimension for a large number of nodes. As such it is difficult to analyze, simulate and synthesize. In addition, its stability is not guaranteed. Due to the saturation of $g(x_i)$, an acceptable solution for the firing rates may involve very high values for the potentials x_i implying a large computational dynamic range.

If the firing rate, $y_i = g_i(x_i)$, is chosen as the basis for assigning the state variables in a neural network, then the analysis above leads to the following observations.

1) The PSP that a firing rate causes at a synapse is $T_{ij}y_j$.

2) The contribution from inputs reaching the axon hillock is $\sum_{j=1}^{N} T_{ij}y_j + I_i$.

3) The actual rate of change in potential at the axon hillock (when normalization and decay are taken into account) is

$$-a_i g_i^{-1}(y_i) + b(y_i)\left[\sum_{j=1}^{N} T_{ij}y_j\right] + I_i.$$

4) Thus the governing equation for a neural network characterized by firing rate is

$$\frac{dy_i(t)}{dt} = g_i\left\{-a_i g_i^{-1}(y_i) + b(y_i)\left[\sum_{j=1}^{N} T_{ij}y_j\right] + I_i\right\}. \quad (2)$$

III. Isomorphic Transformation on the Neural Network Model

Rather than using (2) to describe the equations of motion in terms of firing rates, we define a nonlinear transformation and apply it to (1). Consider the nonlinear transformation on the state space x,

$$y = \varphi(x) \in B \subset R^N \quad (3)$$

where B is a bounded, connected region contained in R^N. The ith term in y is assumed to be a function of x_i only,

$$y_i = \varphi_i(x) = g_i(x_i), \qquad i = 1, \cdots, N. \quad (4)$$

We now transform the original model (1):

$$\frac{dy(t)}{dt} = \frac{d\varphi(x)}{dt} = \frac{\partial g(x)}{\partial x}\frac{dx(t)}{dt} = \frac{\partial g(x)}{\partial x} \cdot \left[-Ax(t) + b\{g[x(t)]\}Tg[x(t)] + I\right] \quad (5a)$$

or

$$\frac{dy_i}{dt} = \frac{\partial g_i(x_i)}{\partial x_i}\frac{dx_i}{dt} = \frac{\partial g_i(x_i)}{\partial x_i}\left(-a_i x_i + b_{ii}\sum_{j=1}^{N} T_{ij}g_j(x_j) + I_i\right) \quad (5b)$$

and

$$\frac{\partial g(x)}{\partial x} = \text{diag}\left\{\frac{\partial g_i(x_i)}{\partial x_i}\right\} = \text{diag}[\lambda_i(y_i)] = L(Y). \quad (5c)$$

Also,

$$x_i = g_i^{-1}(y_i) \quad (5d)$$

and therefore, when (5c) and (5d) are substituted into (5a), we get

$$\frac{dy_i(t)}{dt} = \lambda_i(y_i)\left\{-a_i g_i^{-1}(y_i) + b_{ii}(y_i)\left[\sum_{j=1}^{N} T_{ij}y_j\right] + I_i\right\}, \quad i = 1, \cdots, N \quad (6a)$$

and, in matrix form,

$$\frac{dy(t)}{dt} = \Lambda(y)\left[-Ag^{-1}(y) + b(y)Ty + I\right]. \quad (6b)$$

The model (6b) is isomorphic to (1) since the mapping $g_i(x_i)$ is one-to-one and onto for all $i = 1, \cdots, N$. However, it suggests the description of information flow in terms of the firing rates and not in terms of potentials. The rate of change of the firing rate of the ith node is proportional to its external input I_i, to a linear combination of the firing rates of the other nodes ($\sum_{j=1}^{N} T_{ij}y_j$), and its potential, $g^{-1}(y_i)$. The summed input is shunted by $\lambda_i(y_i)$, which indicates the sensitivity of the ith node to its input. Figs. 1(a) and (b) describe a typical sigmoid function $g_i(x_i)$ and its partial derivative $\lambda_i(y) = \partial g_i(x_i)/\partial x_i$. $\lambda_i(y)$ is a monotonically decreasing function of $|x_i|$; therefore, an increase in the node's potential implies a monotonic loss of sensitivity to the input (saturation).

Computational Advantages of the Firing-Rate Representation

Computationally, (6a) and (6b) have several advantages over the original model (1).

Quasilinearity: When the network operates far away from the state space's origin ($|y_i| \gg 0$, $i = 1, \cdots, N$), the Jacobian $\Lambda(y)$ is almost constant, and the shunting function reaches its saturation value b_{ii}. Equation (6a) becomes

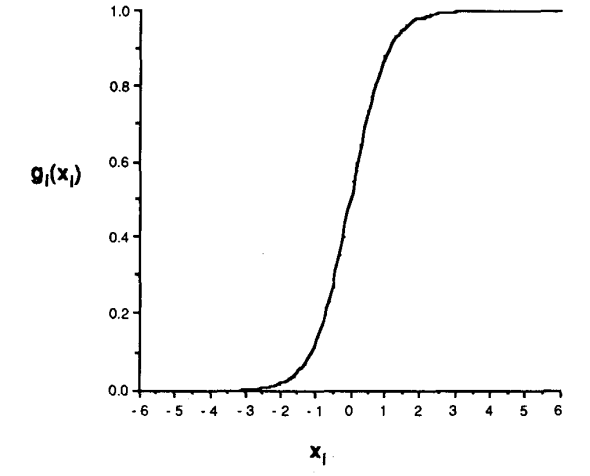
$$\frac{dy_i(t)}{dt} = -\lambda_i a_i g_i^{-1}(y_i) + \left[\sum_{j=1}^{N} \lambda_i b_{ii} T_{ij}y_j\right] + \lambda_i I_i, \quad i = 1, \cdots, N \quad (7)$$

which is almost linear in y_i, since $\lambda_i g^{-1}(y_i) = (\partial g_i/\partial x_i)x_i$ goes toward a constant as y_i grows.

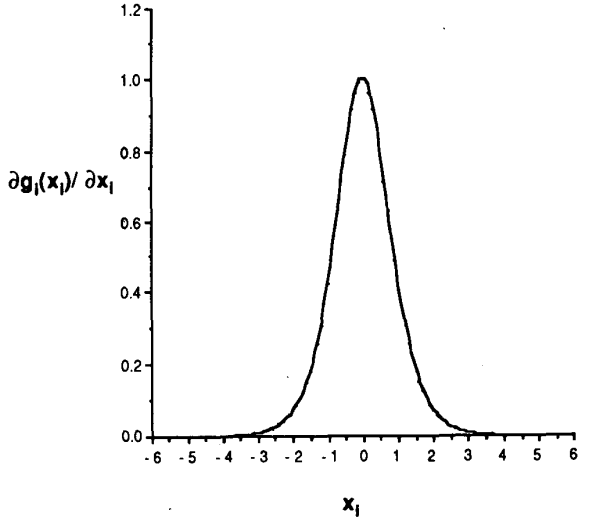
Powerful methods used in linear system analysis may be applied to the solution and study of (7), for the quasilinear hyperactive mode. For most problems of interest the condition $|y_i| \gg 0$, $i = 1, \cdots, N$, represents the region of interest for the network, which contains the stable solution [7].

Bounded Dynamic Range: While the potentials x_i in (1) are not bounded, the firing rates experience saturation for both high and low potentials. This saturation prevents overflows, as the numerical values of y are always bounded between zero and a finite maximum rate. In addition, we can use the boundedness for improving resolution and computational accuracy, with finite precision hardware (e.g., dynamic scaling with integer arithmetic could be used, rather than floating point arithmetic).

Reduced Requirements on Calculations and Storage: At each simulation step, (1) requires N^2 evaluations of the nonlinear



(a)



(b)

Fig. 1. (a) Typical sigmoid function $g_i(x_i)$. (b) Its partial derivative $\lambda_i(y_i) = \partial g_i(x_i)/\partial x_i$.

terms $g_i(x_i)$, or N evaluations along with the storage of N real values of the function. The formulation (6), on the other hand, requires only $2N$ evaluations (N for $\Lambda(y)$ and N for $g^{-1}(y)$).

IV. Firing Rate versus Potential Formulation in the Neural Solution to the Traveling Salesman Problem

Hopfield and Tank [8] have recently suggested a "neural" solution to the traveling salesman problem. For a tour of N cities, N^2 graded-response nodes are used. The nodes are displayed in an $N \times N$ square array like the one in (8), pertaining to the steady-state firing rates of 25 nodes for a five-city problem. The interpretation of the particular solution in (8) is the tour $C-A-E-B-D$

City	1	2	3	4	5
A	0	1	0	0	0
B	0	0	0	1	0
C	1	0	0	0	0
D	0	0	0	0	1
E	0	0	1	0	0

(8)

The energy function minimized by the neural network is

$$E = \frac{A}{2}\sum_X \sum_i \sum_{j\neq i} V_{x_i}V_{x_j} + \frac{B}{2}\sum_i \sum_X \sum_{X\neq Y} V_{x_i}V_{y_i} + \frac{C}{2}\sum_X \sum_i \left(V_{x_i} - N\right)^2 + \frac{D}{2}\sum_X \sum_{Y\neq X} \sum_i d_{XY}V_{x_i}\left(V_{Y_{i+1}} + V_{Y_{i-1}}\right) \quad (9)$$

where

A, B, C, D	positive constants,
X, Y	city names,
i, j	indices $1, 2, \cdots, n$,
V_{Xi}	$= g(u_{Xi})$, the firing rate of the node X_i, where u_{Xi} is the potential of the node x_i, corresponding to the ith column in the row for city X,
$d_{x,y}$	distance between city X and city Y.

The terms in (9) have the following interpretation:

- the first penalizes for having more than one 1 in every city row;
- the second penalizes for having more than one 1 in every position column;
- the third penalizes for having a number of 1's in the array which does not equal N;
- the last represents the total length of the tour which is to be minimized.

Tank and Hopfield have shown that the following N^2 equations, expressed in terms of potentials, need to be simulated for the minimization of (9):

$$\frac{du_{X_i}}{dt} = -a_i u_{X_i} - A\sum_{j\neq i} V_{X_j} - B\sum_{Y\neq X} V_{Y_i} - C\left(\sum_X \sum_j V_{X_j} - N\right) - D\sum_Y d_{XY}\left(V_{Y_{i+1}} + V_{Y_{i-1}}\right) \quad (10)$$

where the sigmoidal relation between the potential and the firing rate was chosen to be

$$V_{X_i} = g(u_{X_i}) = \frac{1}{2}\left(1 + \tanh\left(\frac{u_{X_i}}{u_0}\right)\right), \qquad \text{for all } X, i \quad (11)$$

and u_0 is a normalizing constant.

The model (10) can now be converted via the isomorphic transformation in Section III to yield

$$\frac{dV_{X_i}}{dt} = \frac{2V_{X_i}}{u_0}\left(1 - V_{X_i}\right)\left[-\frac{a_i u_0}{2}\ln\left(\frac{V_{X_i}}{1 - V_{X_i}}\right) - A\sum_{j\neq i} V_{X_j} - B\sum_{Y\neq X} V_{Y_i} - C\left(\sum_X \sum_j V_{X_j} - N\right) - D\left(\sum_Y d_{XY}\left(V_{Y_{i-1}} + V_{Y_{i-1}}\right)\right)\right]. \quad (12)$$

and hence *as the firing rate approaches the saturation limits, the equation simply becomes* $dV_{Xi}/dt = 0$.

Simulation results

We have simulated (10) and (12), using the simultaneous differential equation solver (DVERK in the IMSL [9]). This routine solves a set of nonlinear differential equations using the Runge Kutta-Verner fifth- and sixth-order method. In Fig. 2 we show the city configurations studied through our simulation (5, 7, 10, and 30 cities). The results are summarized in Table I and in

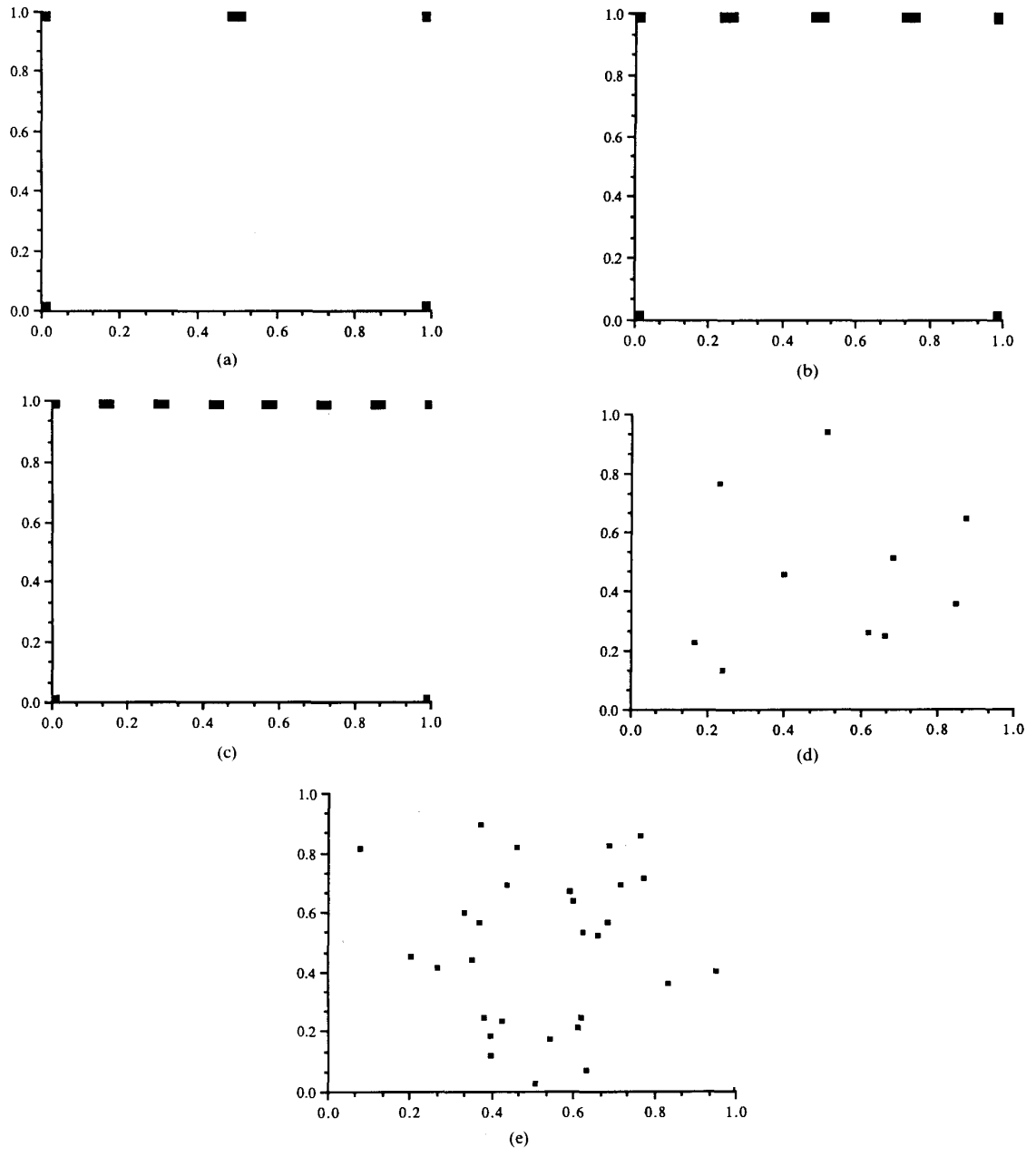


Fig. 2. City arrangements used in simulations of traveling salesman problem. (a) Five cities, square arrangement. (b) Seven cities, square arrangement. (c) Ten cities, square arrangement. (d) Ten cities, arrangement used in [8]. (e) Thirty cities, arrangement used in [8] and [12].

Fig. 3. Table I presents the average tour length, as obtained by Monte Carlo simulations, using the "original" potential-based representation (10), and the "natural variable" representation (12). For all simulations we have used random starting points for the equations. We note the following properties of the solution.

1) The new formulation converges to the optimal solution for all but the 30-city problem, even when the original solution did not. The new formulation has a shorter average tour length than the original formulation.
2) Using DVERK, the original formulation does not converge for the 30-city problem, while the new formulation does.

Fig. 3 shows the average CPU time required to calculate a valid tour, as well as the standard error of the CPU time. The graphs are drawn for five, seven, and ten cities. In the ten-city case we show both the square arrangement and the arrangement used in [8].

3) The average CPU times calculated for both formulations indicates that the original formulation requires $\sim N^2$ calculations for the N-city problem, while the new formulation requires time that is linear in N.
4) The firing-rate-based formulation shows a clear advantage over the potential-based formulation in both average and standard error of the time required to calculate a valid tour.

TABLE I
AVERAGE TOUR LENGTH FOR THE TRAVELING SALESMAN PROBLEM

Number of Cities	Arrangement	Average Length of Valid Tours Potential-Based Representation	Firing-Rate-Based Representation	Optimal Tour Length
5	Square	5.63[a]	4.90[a]	4.00
7	Square	5.55[a]	5.60[a]	4.00
10	Square	6.37	5.96[a]	4.00
10	Hopfield and Tank [8]	5.03	4.17[a]	2.83
30	Lin and Kernighan [12]	no convergence	10.58	4.26[b]

[a] The optimal tour has been discovered by the algorithm.
[b] Best known tour (not necessarily optimal).

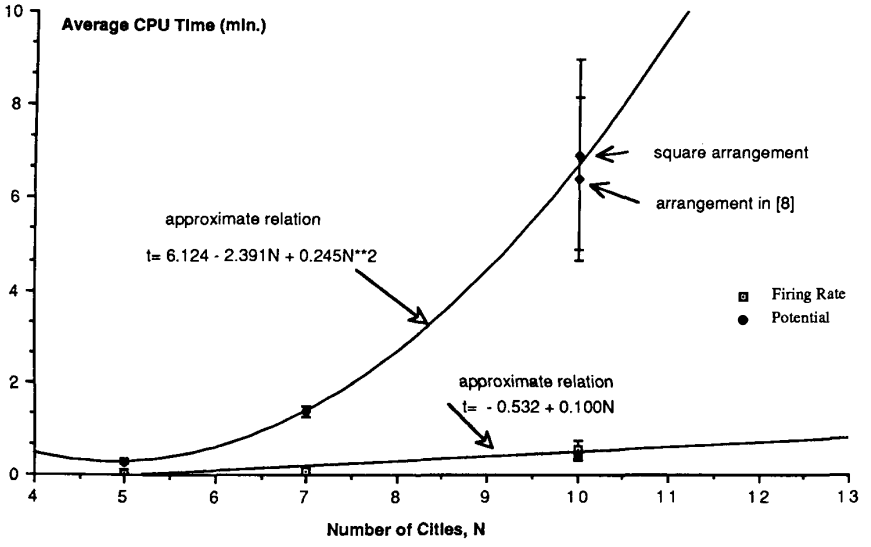


Fig. 3. Traveling salesman problem: average CPU time for discovering valid tour.

V. CONCLUSION

The simple transformation proposed in this study converts the popular model of neural-network behavior from a potential to a firing-rate representation. The new formulation is physiologically sound, as it emphasizes the centrality of the firing rate in the understanding of neural network behavior. In addition, it manifests a clear computational advantage over the original model: it has a limited dynamic range, the number of calculations is significantly reduced and the model becomes quasi-linear in an important subspace of the solution space. The example of the traveling salesman problem demonstrates that our model is capable of increasing the dimensionality of problems that lend themselves to a neural solution. In addition, the proposed transformation generalizes to discrete-time continuous-variable models without further analysis.

ACKNOWLEDGMENT

The authors thank F. Horan and M. J. O'Brien for assistance in simulating the traveling salesman problem algorithms. Helpful comments by an anonymous reviewer are gratefully acknowledged.

REFERENCES

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Sci.*, vol. 9, no. 1, pp. 147–169, 1985.
[2] J. A. Anderson, J. W. Silverstein, S. A. Ritz, and R. S. Jones, "Distinctive features, categorical perception, and probability learning: Some applications of a neural model," *Psychol. Rev.*, vol. 84, pp. 413–451, 1977.
[3] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision, Graphics, Image Processing*, vol. 37, pp. 54–115, 1987.
[4] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 5, pp. 815–825, 1983.
[5] S. Grossberg, "How does the brain build a cognitive code?" *Psychol. Rev.*, vol. 87, pp. 1–51, 1980.
[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554–2558, 1982.
[7] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088–3092, 1984.
[8] J. J. Hopfield and D. W. Tank, "'Neural' computation of decision in optimization problems," *Biol. Cybern.*, vol. 52, pp. 1–12, 1985.
[9] *IMSL — The IMSL Library — A Set of Fortran Subroutines for Mathematics and Statistics*. Houston, TX: 1980, vol. 1, ch. D.
[10] B. Katz, *Nerve, Muscle, Synapse*. New York: McGraw-Hill, 1966.
[11] T. Kohonen, *Associative Memory: A System Theoretic Approach*. New York: Springer, 1977.
[12] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operation Res.*, vol. 21, pp. 498–516, 1973.
[13] W. A. Little and G. L. Shaw, "Analytic study of the memory storage capacity of a neural network," *Math. Biosci.*, vol. 30, pp. 281–290, 1978.
[14] K. R. Popper and J. C. Eccles, *The Self and Its Brain*. Berlin: Springer-Verlag, 1977.
[15] K. H. Pribram, *Languages of the Brain: Experimental Paradoxes and Principles in Neuropsychology*. New York: Brandon House, 1971.
[16] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC: Spartan Books, 1961.
[17] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming

circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 5, pp. 533–541, May 1986.
[18] B. Widrow, "Generalization and information storage in networks of Adaline neurons," in *Self-Organizing Systems*, M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, Eds. Washington, DC: Spartan Books 1962, p. 435.
[19] H. R. Wilson and J. D. Cowan, "A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue," *Kybernetics*, vol. 13, pp. 55–80, 1973.

Decoupling Fuzzy Relational Systems—An Output Feedback Approach

CHEN WEI XU

Abstract—**The paper deals with output feedback decoupling problem for fuzzy relational dynamic systems. The structure of a decoupling controller is proposed, and some results concerning the solution of the decoupling problem are presented in terms of the proposed theorems. The essential difference between decoupling a fuzzy system and a nonfuzzy one is that in the former, all the variables are characterized by their membership functions.**

I. Introduction

Decoupling is a "classical" topic of both control engineering and control theory, which has been attracting the attentions from researchers and engineers working in the field of automatic control [1]–[3]. Even though fruitful results have been obtained in the theory of decoupling, applications of such results in industrial processes are relatively rare as compared to some other branches of control theory such as optimal control, adaptive control, etc. The reason for this may be found from the fact that "...decoupling is very sensitive to parameters and is expensive to implement" [2, p. 487]. In fact problems out of high sensitivity have prevented many theoretical results from possible applications in the area of automatic control.

One way to solve such problems is to develop less sensitive methods. This can be tried either within or outside the framework of the existing decoupling theory. It is proposed here that decoupling problem be treated upon within the framework of fuzzy relational systems [4]–[8], since insensitivity is an inherent property of fuzzy systems.

Fuzzy systems are those systems where part or all of the variables are fuzzy. Fuzzy relations [9] have been used to describe fuzzy dynamic systems by many researchers [4]–[7], [12], [13], [18]. Despite the fundamental work of Bellman and Zadeh, Zadeh, Tong, Pedrycz, and Michel, etc., fuzzy dynamic system theory is still a "developing" theory. Many problems in this area remain unsolved or even unobserved, one of which may be fuzzy system decoupling.

We discussed the cascade compensation approach to fuzzy dynamic systems decoupling in [12]. In this paper, we shall present a second approach: the output feedback approach.

The text is organized as follows. In Section II, a class of multiinput multioutput fuzzy relational dynamic systems is presented and the associated feedback decoupling problem is formally stated. Section III gives the solution (Theorem 2) to the problem. In Section IV, existence of such solutions is investigated and some beneficial results are derived. Finally, the concluding remarks.

Manuscript received March 3, 1987; revised April 15, 1988, and September 11, 1988.
The author is with the Department of Automatic Control, Kunming Institute of Technology, Kunming, People's Republic of China.
IEEE Log Number 8825073.

II. The Fuzzy Systems Under Consideration and the Decoupling Problem

First let us consider a multivariable (p-input/p-output) discrete time linear system in its input/output description,

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_p(k) \end{bmatrix} = \begin{bmatrix} g_{11}(z) & \cdots & g_{1p}(z) \\ \vdots & & \vdots \\ g_{p1}(z) & \cdots & g_{pp}(z) \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_p(k) \end{bmatrix} = G(z) \begin{bmatrix} u_1(k) \\ \vdots \\ u_p(k) \end{bmatrix} \tag{1}$$

where $y_1(k),\cdots,y_p(k)$ and $u_1(k),\cdots,u_p(k)$ are system outputs and inputs at time instant k respectively, and $g_{ij}(z)$, $i, j=1, p$, are proper rational polynomials in z-operator. $G(z)$ is known as the transfer function matrix of the system.

Our intention is to derive a fuzzy relational description of system (1) for the decoupling study. For this purpose, one may directly "translate" the linear system (1) into its fuzzy version, retaining all the original structural features such as orders of the $g_{ij}(z)$ and the time delays included [13]. For convenience, here we only consider first order fuzzy relational system, as most researchers did [4]–[7], [12], [14], [18]. If linear system (1) is a first order system, one has

$$g_{ij}(z) = \frac{b_{ij}}{z + a_{ij}}, \qquad i, j = 1, p \tag{2}$$

where a_{ij} and b_{ij} are constants for all i, j. Thus we can rewrite (1) as

$$\begin{cases} y_1(k+1) = f_1\big(y_1(k), u_1(k), \cdots, u_p(k)\big) \\ \quad\vdots \\ y_p(k+1) = f_p\big(y_p(k), u_1(k), \cdots, u_p(k)\big) \end{cases} \tag{3}$$

where $f_1,\cdots,f_p$ are all linear functions. Similarly, the first-order p-input p-output fuzzy relational dynamic system under consideration is assumed to be

$$\begin{cases} y_1(k+1) = y_1(k)\cdot u_1(k)\cdot\cdots\cdot u_p(k)\cdot R_1 \\ \quad\vdots \\ y_p(k+1) = y_p(k)\cdot u_1(k)\cdot\cdots\cdot u_p(k)\cdot R_p \end{cases} \tag{4}$$

where $u_1(k),\cdots,u_p(k)$ and $y_1(k),\cdots,y_p(k)$ are fuzzy inputs and outputs of the system at time instant k, respectively. Let $U_1,\cdots,U_p$ and $Y_1,\cdots,Y_p$ be the universes of discourse corresponding to $u_1,\cdots,u_p$ and $y_1,\cdots,y_p$ respectively, i.e., $u_1(\cdot)\in F(U_1),\cdots,y_p(\cdot)\in F(Y_p)$. $R_1,\cdots,R_p$ are p fuzzy relations with $R_i \in F(Y_i\times Y_i\times U_1\times\cdots\times U_p)$, $i=1, p$, where $\times$ stands for Cartesian product operation. The operator $\cdot$ in (4) is defined as maxmin composition. The reason why maxmin composition is adopted here is that such systems have been most widely used [1], [2], [4]–[7], [9] and most extensively studied [4]–[7], [12]–[18].

The open-loop system (4) is a coupled or interacted system since, in general, every input controls more than one output and every output is controlled by more than one input. Our goal is to cut off the undesirable couplings or interactions by means of a feedback controller, i.e. to decouple the system via feedback.

Although the essential concept of decoupling serves for both fuzzy systems and nonfuzzy systems, there is still a distinctness between their decouplings. The main point is that in a fuzzy system the variables are characterized by their membership functions, and the concepts of "coupling" and "decoupling" for fuzzy systems are consequently established based on the membership

0018-9472/89/0300-0414$01.00 ©1989 IEEE