

TABLE I
WEIGHTS VALUES IN CONSEQUENCE

weights\Output#	Output 1	Output 2	Output 3	Output 4
wa0	-0.58262	10.09521	7.031277	-0.03806
wa1	1.474861	2.859392	-0.15656	-0.19239
wa2	0.501382	0.723931	7.654014	0.447555
wa3	0.333573	0.627317	5.326648	5.083142
wa4	1.104205	2.492778	0.121761	-9.02884
wa5	0.871362	2.052076	-1.17841	-7.96553
wa6	-2.66596	0.705041	15.33132	0.950651
wa7	-0.73439	-0.16488	10.31283	-0.49057
wa8	1.477351	5.902972	-0.53434	16.44383
wa9	0.097167	4.534635	2.792938	10.05856
wa10	3.101285	0.487503	1.870658	-1.33512

time period for one walking step reduces to 0.52 s. The final values of w_{ai} are listed in Table I.

VI. CONCLUSION

Fuzzy neural network approaches for robotic gait synthesis are presented in this paper. The suggested scheme uses a fuzzy modeling neural network controller with the BTT algorithm in the gait synthesis of a walking robot. The uncertainty of the network size in the conventional neural network learning scheme has been overcome by the use of fuzzy modeling network. The fuzzy controller can generate control sequences and drive the biped along a desired pattern of a walking gait. The desired pattern is used only as a reference trajectory. The proposed learning scheme trains the controller to follow this given pattern as closely as possible. Different pruning algorithms, membership functions, and network structures are investigated. Simulation results demonstrate that the desired goals—crossing over a specific clearance, having a desired step length, and walking at a certain speed—were achieved.

REFERENCES

- [1] S. Murakami, E. Yamamoto, and K. Fujimoto, "Fuzzy control of dynamic biped walking robot," *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 1, no. 4, pp. 77–82, 1995.
- [2] C. L. Shih, W. A. Gruver, and Y. Zhu, "Fuzzy logic force control for a biped robot," *Proc. IEEE Int. Symp. Intelligent Control*, pp. 269–274, 1991.
- [3] L. Magdalena and F. Monasterio, "Fuzzy controlled gait synthesis of a biped walking machine," *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 2, pp. 1334–1339, 1993.
- [4] —, "Learning gait patterns for the fuzzy synthesis of biped walking," in *Proc. Int. Joint Conf. NAFIPS, IFIS and NASA*, 1994, pp. 248–250.
- [5] —, "Evolutionary-based learning applied to fuzzy controllers," *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 3, no. 4, pp. 1111–1118, 1995.
- [6] C. Zhou and K. Jagannathan, "Adaptive network based fuzzy control of a dynamic biped walking robot," *Proc. IEEE Int. Joint Symp. Intelligence and Systems*, pp. 109–116, 1996.
- [7] J.-G. Juang and C.-S. Lin, "Gait synthesis of a biped robot using back-propagation through time algorithm," *Proc. IEEE Int. Conf. Neural Networks*, vol. 3, pp. 1710–1715, 1996.
- [8] J.-G. Juang, "Intelligent path training of a five-link walking robot," *Proc. IEEE Int. Symp. Intelligent Control*, pp. 1–6, 1996.
- [9] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
- [10] J. Furusho and M. Masubuchi, "Control of a dynamical biped locomotion system for steady walking," *J. Dyn. Syst., Meas., Contr.*, vol. 108, pp. 111–118, June 1986.
- [11] Y. F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment," *J. Robot. Syst.*, vol. 2, no. 3, pp. 289–307, 1985.
- [12] M. Raibert, S. Tzafestas, and C. Tzafestas, "Comparative simulation study of three control techniques applied to a biped robot," *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 1, pp. 494–502, 1993.

- [13] L. A. Pars, *A Treatise on Analytical Dynamics*. London, U.K.: Heinemann, 1965, ch. 14.
- [14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [15] J. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. Neural Networks*, vol. 3, pp. 714–723, Sept. 1992.
- [16] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, pp. 801–806, Sept. 1992.
- [17] S. Horikawa, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Composition methods of fuzzy neural networks," *Proc. IEEE Int. Conf. Industrial Electronics, Control and Instrumentation*, pp. 1253–1258, 1990.
- [18] S. Horikawa, T. Furuhashi, S. Okuma, and Y. Uchikawa, "A fuzzy controller using a neural network and its capability to learn experts control rules," in *Proc. Int. Conf. Fuzzy Logic and Neural Networks*, 1990, pp. 103–106.
- [19] C. T. Chen, *Linear System Theory and Design*. New York: Rinehart, 1990.
- [20] J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [21] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, no. 2, pp. 135–156, 1992.

Constructing Hysteretic Memory in Neural Networks

Jyh-Da Wei and Chuen-Tsai Sun

Abstract—Hysteresis is a unique type of dynamic, which contains an important property, *rate-independent memory*. In addition to other memory-related studies such as time delay neural networks, recurrent networks, and reinforcement learning, rate-independent memory deserves further attention owing to its potential applications. In this work, we attempt to define hysteretic memory (rate-independent memory) and examine whether or not it could be modeled in neural networks. Our analysis results demonstrate that other memory-related mechanisms are not hysteresis systems. A novel neural cell, referred to herein as the propulsive neural unit, is then proposed. The proposed cell is based on a notion related the submemory pool, which accumulates the stimulus and ultimately assists neural networks to achieve model hysteresis. In addition to training by backpropagation, a combination of such cells can simulate given hysteresis trajectories.

Index Terms—Hysteresis, hysteretic memory, rate independence, recurrent network, reinforcement learning, time delay neural network.

I. INTRODUCTION

In addition to its simple and practical nature, network computation is adaptive. Network approximation is thus used in many control systems [31], [52]. Neural networks with a short-term memory, occasionally referred to as belonging to dynamic systems, are also highly attractive for processing time varying signals [32], [56]. In this work, we study another type of dynamic, i.e., hysteresis, which is a unique case of memory. Whether or not neural networks can model hysteresis behaviors is discussed. Network computation can be more fully implemented to achieve a better performance from those systems that are

Manuscript received June 25, 1998; revised June 20, 1999 and March 22, 2000. This work was supported in part by the National Science Council, R.O.C., under Contract NSC87-2213-E009-004. This paper was recommended by Associate Editor W. Pedrycz.

The authors are with the Institute of Computer and Information Science, National Chiao Tung University, Hsin-Chu, Taiwan 300, R.O.C. (e-mail: jdwei@cindy.cis.nctu.edu.tw).

Publisher Item Identifier S 1083-4419(00)06722-4.

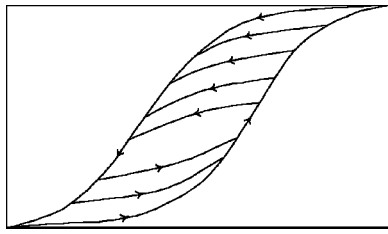


Fig. 1. Typical branches of hysteresis. The outcome of hysteresis is not only based on the current input value, but is also related to the previous history.

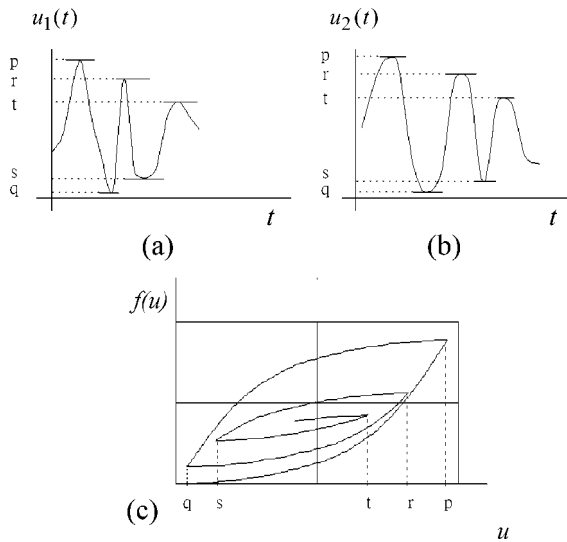


Fig. 2. Hysteresis as a rate-independent memory effect (RIME), where (a) and (b) plot the two different inputs, which have the same successive extreme inputs. These input plots result in the same f - u effect diagram of hysteresis.

hysteresis embedded if neural networks could be constructed as hysteresis simulators.

Hysteresis is a memory effect, with its literal meaning implying to “lag behind.” Fig. 1 depicts the typical shape of a hysteresis diagram. This figure indicates the characteristic behavior of a hysteresis system, a lag in evocation, and perseverance in recovery. When the input value alternates between increasing and decreasing, the response curve does not continue to follow the original path; instead, it draws a new effect delayed curve. In other words, the outcome of a hysteretic mechanism is not only based on the current input value, but is also related to the previous history. Therefore, hysteresis behavior can be easily distinguished from conventional static mathematical functions.

Hysteresis also heavily concentrates on an important property, *rate independence*. Generally speaking, in the discussion of hysteresis non-linearity, only the previous extreme input values determine the hysteresis branches. The speed of input variations is not an influential factor. Fig. 2 illustrates this property. Fig. 2(a) and (b) plot two different inputs, $u_1(t)$ and $u_2(t)$, whose successive extreme values are the same. These inputs initially rise to p , fall to q , and are then followed by values such as r , s , t . Let the output of the hysteresis system be $f(u)$. Fig. 2(c) summarizes the results in the same f - u diagram. This property is known as rate independence. Restated, hysteresis is considered as the rate-independent memory effect (RIME) [47].

As an extreme example of rate independence, we track the output in response to input sequence $\{1, 2, 5, 8, 8, 8, 8, 8, 9, 12, \dots\}$. Imagine that the input sequence is adopted in a hysteresis system. The response is varied with 1, 2, 5, 8 sequentially input, and remains on the same measure during the run of unchanged 8's. To emphasize the difference from other memory-related schemes, we first consider the time delay

neural network (TDNN) [25], [51]. A TDNN uses delay kernels with weight factors to ensure short-term memory. Therefore, the above inputs cause a TDNN to alter its output even with the same input values, i.e., unless the memory depth is full. According to an earlier investigation on recurrent networks [14], [30], [53], [54], the response cannot remain on the same value either, because the feedback requires some iteration to converge (if at all possible). From another aspect, delayed rewards in reinforcement learning (RL) may be considered as a case of memory. Herein, we focus primarily on delayed rewards in RL by employing methods such as temporal difference learning [TD(λ)] [39] and Q -learning [49]. The notion applied in TD(λ) is credit assignment toward temporal patterns, while Q -learning is based on Markov chains. These two concepts are both related to the time factor and, thus, differ from hysteresis.

Consider another example, one involving the concept of rate independence mentioned earlier. Assume not only that the sequence $\{1, 2, 5, 30, 8, 9, 12, \dots\}$ is the input, but also that the extreme value 30 is not attributed to noise. According to our results, the value 30 impacts the hysteresis system for an extended period. Notably, presenting the value 30 changes the path in effect diagram [f - u diagram in Fig. 2(c)]. Such an influence can be eliminated only after the coming input is less than a certain scale, e.g., 0 or -30 in some cases. Our results further indicate that for the same input sent into some memory-related schemes (other than the hysteresis system): a) time plays an important role and b) a certain long time or a sufficiently long sequence of common inputs can neglect a circumstance in which the value 30 occurred. In summing up these two examples, hysteresis is characterized by rate-independent memory (also referred to as “*hysteretic memory*”). Regardless of how slow or fast the input values appear, only the previous extreme values determine the response, even if the inputs remain for a long time or merely flash at once. This property makes hysteresis markedly different from other memory-related studies, thereby meriting its thorough investigation.

The rest of this paper is organized as follows. Section II reviews hysteresis related literature, indicating its prominence in diverse fields. This section also briefly introduces the Preisach model, which is the most instructive and important model in study of hysteresis. Section III defines the rate-independent memory by formulation. Section IV describes the structure of our approach to modeling hysteresis and derives a backpropagation procedure to train this model. Concluding remarks and areas for future research are finally made in Section V.

II. HYSTERESIS RELATED LITERATURE

A. Review of Related Studies

The term hysteresis originates from ancient Greek and is first used while describing ferromagnetism [see Visintin ([47], p. 9)]. In fact, many useful models have emerged from this domain [17]–[19]. Another cradle for early hysteresis models to develop is plasticity. After Tresca introduced the maximum shear stress yield criterion in 1864, successive investigations increasingly emphasized this criterion [23], [47]. In addition to these two areas, many studies subsequently followed. According to these studies, hysteresis can also be found in various fields, including microelectronics (ferroelectricity [15], [27]), thermodynamics (thermostat [6], [8], thermal relaxation [9]), and in some recently developed materials (the shape memory effects [1], [2], [28]) and mechanics [5], [44].

Besides physical engineering, this phenomenon also occurs in cognitive engineering. Hysteresis can be observed in spatiotemporal pattern recognition (nonstationary noise clearing [21], [42]), time varying signal processing (phase transitions [4]), and cybernetics (control of plants [29], [41]). Moreover, it is becoming increasingly important in the fields of psychology (long-term memory and painful

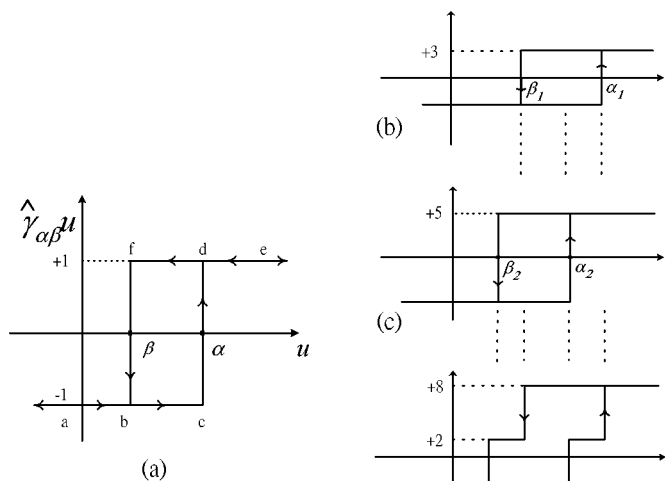


Fig. 3. Preliminary hysteresis operators in the Preisach model. The Preisach model of hysteresis is based on the combination of $\hat{\gamma}_{\alpha\beta}$ operators.

experience [38]) and economics [10], [11], [45], [48]. There are still some unresolved questions in all of the above fields, and rate-independent memory can be a hint. Why is the wage rigidity [7], [22] considered in the Keynesian model? Does the stock market rise and become different after the indices have risen higher than some level [16]? These are related, to certain extents, to rate-independent memory. As generally known, reversing a situation (or even forgetting it) after it occurs is extremely difficult. Such a situation can persist for a long time, ultimately reducing the influence, not because time has passed but because another new scene (extreme value) occurs.

B. Mathematical Models

As mentioned above, the study of hysteresis begins with ferromagnetism. Pertinent literature regarding ferromagnetic hysteresis, which can be found in [35], indicates that Lord Rayleigh [34] proposed the first model in 1887, and the most important model, the Preisach model, was proposed in 1935 [24], [33].

The Preisach model has received extensive attention [37], [43], [46] since it was published in 1935. This model contains the notion that a complicated system can be constructed as a superposition of simplest operators (Fig. 3). The operator $\hat{\gamma}_{\alpha\beta}$ is the unit of the Preisach model. Footnotes α and β denote the operating limitation of each operator. Each operator works with the current input value, u , and results in bi-valued hysteresis (returns -1 or 1) owing to that the ascending paths and the descending paths are switched. While combining operators, such as $\hat{\gamma}_{\alpha_1\beta_1}$, $\hat{\gamma}_{\alpha_2\beta_2}$, etc., an entire system can be expressed by the following equation:

$$f(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) [\hat{\gamma}_{\alpha\beta} u(t)] d\alpha d\beta \quad (1)$$

where $\mu(\alpha, \beta)$ is the weight to each $\alpha - \beta$ operator $\hat{\gamma}_{\alpha\beta}$.

The Preisach model is known as a kind of “nonlocal memory hysteresis model” [26] since it responds to the outputs while thoroughly considering the previous inputs. Another kind of hysteresis model is called “local memory hysteresis model.” These models generate their next output by consulting the local memory they have produced in the current state. In this case, at most two curves pass through each point in the input-output (I/O) matching diagram, say, the $f-u$ diagram. For

an increasing input $u(t)$, the rising curve is followed, causing the response $f(u)$ to rise with u . If the input is decreased, then the falling curve is traced.

The Hysteresy model [42] is an instance of local memory models. Both the upper and lower branches of a hysteresis loop in this model are described by hyperbolic tangent functions.

III. DEFINITION OF HYSTERESIS SYSTEM

In the previous section, we introduced rate-independent memory and briefly mentioned that hysteresis is frequently observed in many fields. However, other memory related mechanisms are not rate-independent and, thus, may not function properly in a hysteresis system. In this section, we repeat these instances by a formulation. An attempt is initially made to properly define a rate-independent system in terms of digital data processing. In doing so, all of the schemes, such as time delay neural networks, recurrent networks, and some reinforcement learning methods, are distinguished from hysteresis. In the next section, we propose the propulsive model to model the hysteresis behavior in neural networks.

A. Rate-Independent Systems

Initially, we denote a system T that transduces an input sequence with values $x[n]$ into an output sequence with values $y[n]$ as the following form:

$$y[n] = T\{x[n]\}. \quad (2)$$

Then, a rate-independent system is defined as a system having the following two properties.

Property 1—Ineffective Insertion of Mean Values (IIMV): Consider an input sequence $x = \{\dots, x[k], x[k+1], \dots\}$ and two contiguous input values in it, say $x[k]$ and $x[k+1]$. We insert a mean value $x_q = px[k] + (1-p)x[k+1]$ with $0 \leq p \leq 1$ between $x[k]$ and $x[k+1]$. In doing so, a new sequence w is obtained as $w[i] = x[i]$ for $i \leq k$, $w[k+1] = x_q$, and $w[j] = x[j-1]$ for $j > k+1$. Thereafter, x and w are sent to two systems, which are both duplicated from system T . By assuming that the corresponding outputs are sequences y and z , we state that the system T has ineffective insertion of mean values if $z[j] = y[j-1]$ with $j > k+1$ for any input sequence x and any interpolating index k . (In addition, it is always true that $z[i] = y[i]$ for $i \leq k$ in any system. The output in response to x_q , $z[k+1]$, is not important here.)

Property 2—Ineffective Removal of Mean Values (IRMV): For an input sequence x , we select any three contiguous values $x[k-1]$, $x[k]$ and $x[k+1]$ under the condition $x[k] = px[k-1] + (1-p)x[k+1]$ where p satisfies the condition that $0 \leq p \leq 1$. A new sequence w is obtained after we omit $x[k]$ such that $w[i] = x[i]$ for $i < k$ and $w[j] = x[j+1]$ for $j \geq k$. By sending x and w to two duplicate systems both referring to system T , let the corresponding output sequences be y and z . We state that the system T has ineffective removal of mean values if $z[j] = y[j+1]$ with $j \geq k$ for any input sequence x and any omitting of index k . (Also, it is obvious that $z[i] = y[i]$ for $i < k$ in any system.)

Definition 3—Rate-Independent System: A rate-independent system contains both of the above IIMV and IRMV properties.

The inserted mean values could turn to be a *subsequence* by iteratively applying the IIMV property on the same interpolating index. Similarly, continuous removed mean values can form a subsequence as the IRMV property is applied repeatedly. That is, insertion (or removal) of an input subsequence is also ineffective for a rate-independent system as long as this subsequence has no extreme value according to the newly generated input sequence.

B. Hysteresis Systems

Consider that a system is memoryless iff the output $y[n]$ at every value of n depends only on the input $x[n]$ on the same value of n . That is, $y[n] = f(x[n])$ for each value of n , where $f(\cdot)$ is one of any functions. A memoryless system is surely a rate-independent system. More specifically, a hysteresis system is defined herein as a rate-independent system with a causal memory.

Definition 4—Hysteresis System: A hysteresis system is rate independent, causal, and not memoryless.

Restated, a hysteresis system has rate-independent causal memory. It is also referred to as “*hysteretic memory*.”

A necessary condition for a hysteresis system can be derived from its causality and rate independence.

Lemma 5—Stayed Input Stayed Output (SINSOUT): For a rate-independent system, an unvaried input subsequence causes an output subsequence that also remains unchanged.

Proof: By applying the IIMV property on interpolating index k with $p = 0$, the inserted mean value is $x_q = x[k + 1]$, which is also viewed as the unvaried input value. Let us focus on the key I/O subsequences. The original I/O pair is $\{x[k], x[k + 1]\}$ and $\{y[k], y[k + 1]\}$. By rate independence, the inserted I/O pair is $\{w[k] = x[k], w[k + 1] = x_q = x[k + 1], w[k + 2] = x[k + 1]\}$ and $\{z[k] = y[k], z[k + 1] = T\{x_q\}, z[k + 2] = y[k + 1]\}$. Because of causality, $T\{x_q\}$ with $x_q = x[k + 1]$ is only based on all the input values $x[i]$ for $i \leq k + 1$. Therefore, $T\{x_q\} = T\{x[k + 1]\} = y[k + 1]$. The unchanged output value $z[k + 2] = y[k + 1] = z[k + 1]$ is shown here. The proof is completed while we repeatedly apply the IIMV property as above. \square

C. Multilayer Feedforward Neural Networks versus Hysteresis

The preliminary neural network model is the multilayer feedforward proceptron [52]. The activation of each node, x_i , in this preliminary model can be represented as

$$x_i = \sigma(\text{net}_i) + I_i \quad (3)$$

where

$$\text{net}_i = \sum_{j < i} w_{ij} x_j \quad (4)$$

w_{ij} denotes the weight factor connecting node j to node i , $\sigma(\cdot)$ is a squashing function, and I_i represents the bias of node i . To show the activation of this system in the form of a sequence, (3) and (4) can be rewritten as

$$x_i[n] = \sigma(\text{net}_i[n]) + I_i \quad (5)$$

and

$$\text{net}_i[n] = \sum_{j < i} w_{ij} x_j[n]. \quad (6)$$

According to the above equations, this is a memoryless system and obviously not a hysteresis system by definition.

D. Convolution Neural Models versus Hysteresis

A time delay neural network (TDNN) is one of the so-called “convolution models” [13]. The activation of the nodes in convolution models is improved from (6). Herein, we let

$$\begin{aligned} \text{net}_i[n] &= \sum_{j < i} \sum_{k=0}^n w_{ij}[n-k] x_j[k] \\ &= \sum_{j < i} \sum_{k=0}^n w_{ij}[k] x_j[n-k]. \end{aligned} \quad (7)$$

For a TDNN, the weight factors $w_{ij}[k]$ are only valid in a bounded range, referred to as “memory depth.” If k exceeds the memory depth, the weight factor equals zero. For some advanced versions of convolution models, e.g., the concentration-in-time net (CITN) [40] and the gamma neural model [13], there is no limitation for memory depth. In the following, we demonstrate that convolution models cannot be hysteresis systems.

The following discussion focuses on the first node, which operates with memory mechanism as above. This event implies that other nodes earlier than this node are all memoryless. Assume that the node is node i , and its response on some input $x[n_i]$ is concerned with

$$\text{net}_i[n_i] = \sum_{j < i} \sum_{k=0}^{n_i} w_{ij}[n_i - k] x_j[k]. \quad (8)$$

When the next input $x[n_t + 1]$ is accepted

$$\begin{aligned} \text{net}_i[n_t + 1] &= \sum_{j < i} \sum_{k=0}^{n_t+1} w_{ij}[n_t + 1 - k] x_j[k] \\ &= \sum_{j < i} \sum_{k=0}^{n_t} w_{ij}[n_t - k + 1] x_j[k] \\ &\quad + \sum_{j < i} w_{ij}[0] x_j[n_t + 1]. \end{aligned} \quad (9)$$

The above equations reveal that $\text{net}_i[n_t + 1]$ does not equal $\text{net}_i[n_t]$, although $x_j[n_t + 1] = x_j[n_t]$ for all $j < i$. As the signal feeds forward, the output of this network also performs the inequality. This performance conflicts with the SINSOUT lemma and, therefore, convolution models cannot establish hysteresis systems.

E. Recurrent Networks versus Hysteresis

In contrast to feedforward networks, a latter node in a recurrent network may have backlinks connected to earlier ones [31]. Therefore, the signal could feed back to join the computation in the next step, thereby constructing the memory effect. The equation can be expressed as

$$\text{net}_i[n] = \sum_{j < i} w_{ij} x_j[n] + \sum_{j \geq i} w_{ij} x_j[n - 1]. \quad (10)$$

Among the unresolved issues surrounding, recurrent networks include such questions as “Is it stable, and inner stable?” “Is it convergent?” and “How fast does it converge?”

Next, whether or not it is rate independent is explored.

Consider the node that is initially linked by some latter nodes. Assume that it is node i , and one of the latter nodes is node j . Notably, the network structure linked before node i is feedforward and, thus, memoryless. Assume that the previous input is x_c , the current input is x_a ; without loss of generality, we denote that the activation of node j on x_c , and x_a is not equal. If x_a is input again, the feedforward part sends the same signal to node i , while node j feedback to node i differs from that activated on x_c inputs. The inequality occurs and the entire network structure has the starting point to generate a different output.

Again, this performance conflicts with the SINSOUT lemma. We can conclude that recurrent networks cannot construct hysteresis systems.

F. Reinforcement Learning versus Hysteresis

Reinforcement learning is applied to discover how to yield the highest reward, and is characteristic of a trial-and-error and delayed reward. A well-trained mechanism is not discussed herein; in fact, it is usually a memoryless controller. However, our primary concern

is the learning algorithm itself, thus raising the following question: Does it offer rate-independent information during learning? Learning algorithms also impact the system response during the training stage.

Sutton proposed temporal difference learning, TD(λ), in 1988 [39]. This incremental real-time algorithm adjusts the weights in a connectionist network. The learning process follows the form

$$\Delta w_t = \alpha(P_{t+1} - P_t) \sum_{k=1}^t \lambda^{t-k} \nabla_w P_k \quad (11)$$

where P_t denotes the network's output upon the input pattern x_t at time t , w represents the vector of weights that parametrizes the network, and $\nabla_w P_k$ is the gradient of network output with respect to weights. This method provides a feasible means for supervised learning procedures to solve temporal credit assignment problems. However, it is not rate independent because it may not correspond to the principle of mean value ineffectiveness.

Other reinforcement learning algorithms [50] are not rate independent either. In fact, some of these algorithms, such as Q -learning [49], function in a discrete finite world, not a numerical world in which hysteresis has been discussed herein so far. Frequency is what the learning algorithms function to predict, thereby making time a major factor. The learning process operates at every time step by the Markov process. Hysteretic memory is unnecessary for these applications.

Nevertheless, we believe that learning depends, to varying extents, on memory. Exactly what role hysteretic memory may operate in reinforcement learning is explored later on.

IV. HYSTERETIC MEMORY VIA THE PROPULSIVE NEURAL UNIT

The previous section has defined hysteresis system. According to this definition, a system is observably not hysteresis if it is sensitive about input speed. Conventional network strategies, either in reference to unselected and limited historic inputs or using fixed recurrent links, are closely associated with data input rate. In sum, networks based on computational nodes and links cannot purely function as a hysteresis simulator. In the following, we present a propulsive neural cell to assist hysteresis simulation. This neural cell is also trained by backpropagation.

A. Using the Delta as the Input

As far as the SINSOUT lemma is concerned, we speculate that the difference between two contiguous inputs may be the essence of the hysteresis system. This difference is referred to herein as the "delta." A circumstance in which the network accepts the delta instead of the real input and only action at nonzero delta input could avert the opposite situation to the SINSOUT lemma.

Although consulting with the delta is preferred, directly using the delta for computation is not an appropriate design. Obviously, a *delta-in-delta-out* network, which computes the movement of output instead of the real response value, does not fully consist of the IIMV property. Alternatively, some mechanisms may need to be designed in a neural unit and the delta adopted as well to modify its state. Then, allow the response to refer to the state of this mechanism. Restated, hysteretic memory thus resides in the inner state of a neuron.

B. Propulsive Neural Unit

This study constructs hysteretic memory in a neural unit based on a hypothesis involving the accumulation of the stimulus. Assume herein that a neuron accepts delta signals and pushes them into a *submemory pool*. The process of push is named "*propulsion*" or "*propulsive process*." By propulsion, if all the deltas are positive, combining all of the deltas becomes a longer and deeper potency carrier. Otherwise,

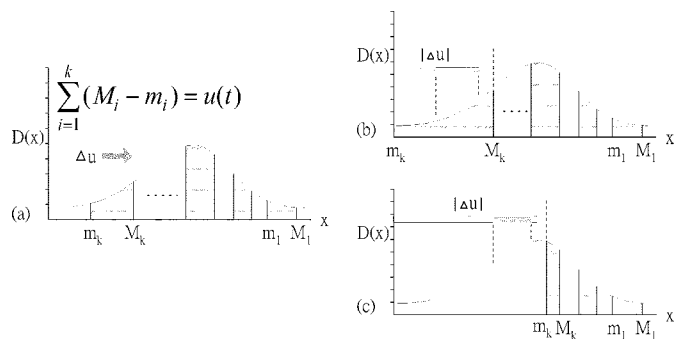


Fig. 4. Propulsive neural models accessing delta inputs. (a) Current state of the neural input, where k is a variable indicating the number of record couples. (b) Propulsion process skips a carried interval and does not stop until the increased delta is exactly pushed. (c) Decreased delta performs the same algorithm while empty intervals are skipped now. Variable k decreases in (b) and (c).

such a combination would be with some empty areas, and indicates a weaker carrier.

For simplicity, this study implements the submemory as an axis, X , which is occupied by *carried intervals* and *empty intervals*. Allow the total length of carried intervals to equal the current input. The first carried interval thus occurs for the first signal inputs (assuming it is positive). Initially, it starts from zero and ends on the value of the first input because the first delta equals the first input. This carried interval sequentially grows during the run of positive delta inputs. Before it stops, it actually occupies a length of size equal to the first extreme input value. Then, a negative delta inputs and uses its absolute value to generate an empty interval also from zero. After the duration of negative delta inputs, the first carried interval turns from the right end of the first empty interval. Thereafter, the second carried interval grows, and so on. In doing so, any mean value inputs do not alter the state of carried intervals and, thus, could not impact the responding outputs. Regarding the neuron outputs, the response to the current input, $u(t)$, can be expressed by integrating a "reaction function," $D(x)$, on the union of all carried intervals, A_t . It is positive and written as

$$f(t) = \int_{A_t} D(x) dx |_{A_t} = \bigcup_{\{\text{carried intervals}\}} \text{on } u(t). \quad (12)$$

Fig. 4 illustrates how this model works. Two sequences, M 's and m 's, are made to record the distribution of intervals. The extreme rising signals are finally recorded by the M 's, while the extreme falling signals are recorded by the m 's. Both M and m sequences are in a monotonically descending order, appearing in couples (M_i, m_i) 's with $M_i > m_i$ except in the initial condition when $M_1 = m_1 = 0$. Restated, these two sequences concern themselves with $M_1 > m_1 > M_2 > m_2 > M_3 > \dots$. Actually, the j th carried interval is recorded by $[m_j, M_j]$, while the j th empty interval is recorded by $[M_{j+1}, m_j]$ or $[0, m_j]$. Therefore, by allowing k to be the total number of record couples, the following equation is valid:

$$u(t) = \sum_{j=1}^k (M_j - m_j). \quad (13)$$

When an increasing signal arrives, the present record M_k is generally shifted from a lower position to a higher one. If the shift meets a carried interval $[m_j, M_j]$, M_j replaces the current record. Restated, this stage must be skipped, and proceed to its right end to continue the propulsive process. The model continues to perform this process until the increased delta is exactly pushed into the carried interval. While a decreasing signal comes, we shift m_k in the same manner except that the skipped stages are now $[M_{j+1}, m_j]$'s.

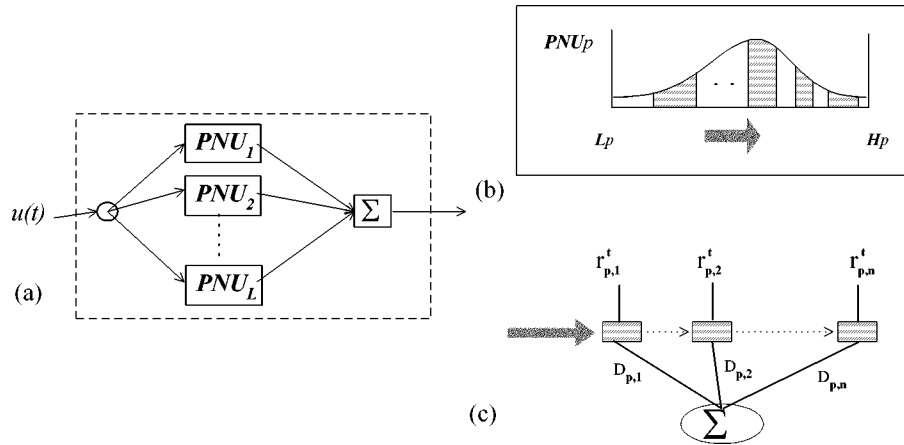


Fig. 5. System established on a basic PNU organization. (a) We treat each PNU as the basic unit in the structure. (b) Picture zooms in on the p th PNU, which has a sensible range $[L_p, H_p]$. (c) Each PNU functions similarly to a one-layered perceptron by selecting a step function as the reaction function.

Consider the following example of tracing a series $\{25, 15, 20, 18\}$. The delta sequence is thus $\{25, -10, +5, -2\}$, and the M and m 's arise as $(M_1, m_1) = (25, 0)$ when the value 25 inputs, $(M_1, m_1) = (25, 10)$ when the value 15 inputs. When the value 20 inputs, it turns out to be $(M_1, m_1, M_2, m_2) = (25, 10, 5, 0)$. Finally, it is $(M_1, m_1, M_2, m_2) = (25, 10, 5, 2)$ when the value 18 inputs. Now, allowing a new signal to arrive as 24, a new M_2 must occur by propulsion and the records list becomes $(M_1, m_1, M_2, m_2) = (25, 10, 9, 0)$.

To summarize, this model allows us to give the response formula again:

$$\begin{aligned} \text{Input: } u(t) &= \sum_{j=1}^k (M_j - m_j) \\ \text{Output: } f(t) &= \int_{A_t} D(x) dx, \quad \text{where } A_t = \bigcup_{j=1}^k [m_j, M_j]. \end{aligned} \quad (14)$$

C. System Structure

A propulsive neural unit (PNU) is a neuron that accepts the delta and operate propulsive process as designed above. In this section, PNUs are used to construct a simple structure. Before doing so, two questions must be addressed: "How can the PNUs be organized?" and "What is chosen to be the reaction function, $D(x)$?"

The *sensible range* for a PNU is initially provided. According to this term, a neuron has its range to accept a signal. Too large or too small a signal must be filtered into a value between a higher bound and a lower one. A PNU operates like the plot in Fig. 2(c) when it accepts a signal with value in the sensible range. An overflow signal makes the submemory pool full, while a deficient signal makes it empty. These two cases cause the PNU to respond to constant values. Cumulatively, a PNU takes the filtered value, counts the delta and integrates the response after propulsion. Now, (13) and (14) must be adjusted with an offset of the lower bound.

As mentioned earlier, each PNU has its own sensible (working) range and its own effective trajectory (of the shape like Fig. 2(c) shown). Collecting some PNUs is an effective means of simulating various types of hysteresis. Thus, as expected, a network can model hysteresis behaviors if it contains some nodes built as PNUs with distinct sensible ranges. For simplicity, the proposed system is basic and only organized into several parallel PNUs. Fig. 5(a) reveals that

the input signal is simply sent to all L PNUs and the response to the summation of their output is obtained as well. Fig. 5(b) displays the p th PNU, whose sensible range is $[L_p, H_p]$.

Regarding the decision of the reaction function, we select step functions for convenience, although their parameters are numerous. A step function $D_p(x)$ is taken onto the p th PNU. The value of the i th scale, $D_{p,i}$, is an adaptable parameter. The collaborative input $r_{p,i}^t$ is calculated as the size of a conjunction of the i th stage and the current carried intervals, where the index t is responding to the input u_t at time t , p denotes that it belongs to the p th PNU and i represents that it should be integrated in the i th stage of the step function $D_p(x)$. That is, the effect corresponding to the i th stage is $D_{p,i} \cdot r_{p,i}^t$. Therefore, each PNU functions similarly to a one-layered perceptron. If there are a total of n stages in each $[L_p, H_p]$, the response value of input u_t can be expressed as

$$f(u_t) = \sum_{p=1}^L \sum_{i=1}^n D_{p,i} \cdot r_{p,i}^t. \quad (15)$$

The series input u_t must be diffused to each PNU, filtered into the sensible range, changed into the delta value, through the propulsive process and, finally, divided into each respective input $r_{p,i}^t$. The total operation design herein to construct hysteretic memory is termed the *propulsive model*.

D. Learning Method

The operation phase and the learning phase are separately discussed in the propulsive model. The operative phase of this model has been presented above. Herein, an adaptive method is proposed to determine the system parameters.

The learning phase of the PNU organization adopts the Widrow-Hoff back propagation rule. This rule is based on an iterative gradient descent algorithm designed to minimize the mse between the desired target values and the actual response values.

Although (15) has demonstrated how to calculate the response value with respect to the local input u_t , a slight transformation must occur on reaction function, $D_p(x)$, to denote each step and complete the adaptation. Let us take

$$f(u_t) = \sum_{p=1}^L \sum_{i=1}^n D_{p,i}^t \cdot r_{p,i}^t \quad (16)$$

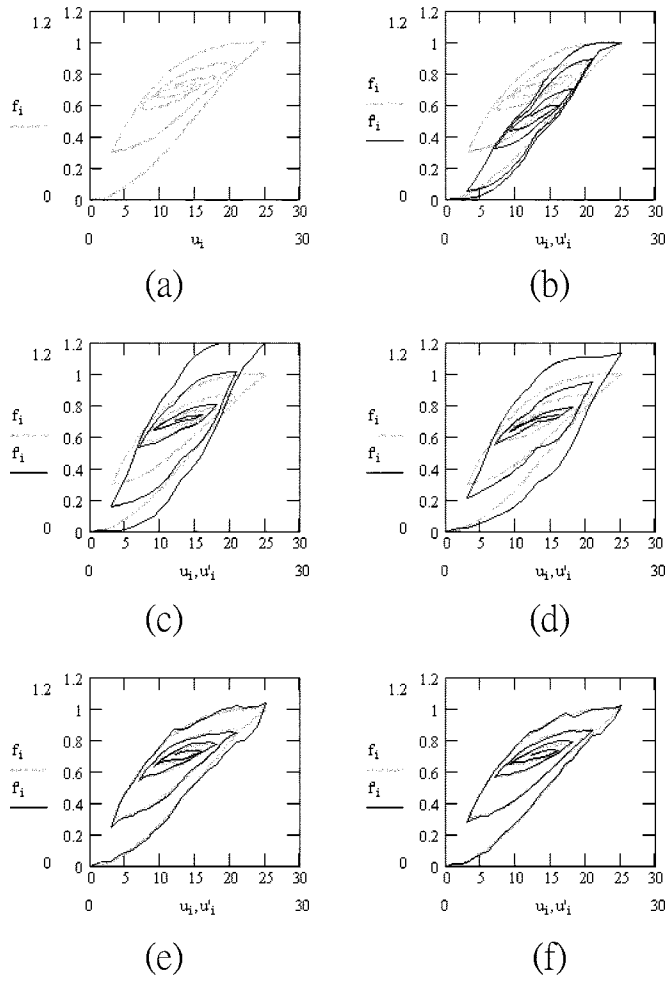


Fig. 6. Training the PNU organization to follow the loops given by (20) with $\mu(\alpha, \beta) = (4.5/25^2) - (3/25^3)\alpha - (1/25^3)\beta$. (a) Target diagram, with 111 points on it. The continuous four figures use four PNUs. (b) Initial trajectory versus target diagram. (c) After one epoch, the mse has the value 3.59×10^{-3} . (d) After two epochs, $\text{MSE} = 2.18 \times 10^{-3}$. (e) After 1528 epochs, $\text{MSE} = 2.82 \times 10^{-4}$. The remaining figure uses six PNUs. (f) After 814 epochs, $\text{MSE} = 1 \times 10^{-4}$.

and represent the target value as $G(u_t)$. The error is expressed as the following equation:

$$E = \frac{1}{2} \sum_{t=1}^N (G(u_t) - f(u_t))^2 \quad (17)$$

where N denotes that there are total N patterns.

The adaptation rule is formulated as

$$D_{p,i}^{t+1} = D_{p,i}^t + \Delta D_{p,i}^t \quad (18)$$

Thus, it is derived that

$$\begin{aligned} \Delta D_{p,i}^t &= -\eta \frac{\partial E}{\partial D_{p,i}^t} \\ &= \eta (G(u_t) - f(u_t)) \cdot \frac{f(u_t)}{\partial D_{p,i}^t} \\ &= \eta (G(u_t) - f(u_t)) \cdot \frac{\sum_{q=1}^L \sum_{j=1}^n r_{q,j}^t \cdot D_{q,j}^t}{\partial D_{p,i}^t} \\ &= \eta (G(u_t) - f(u_t)) \cdot r_{p,i}^t \end{aligned} \quad (19)$$

where the positive value η is the learning rate.

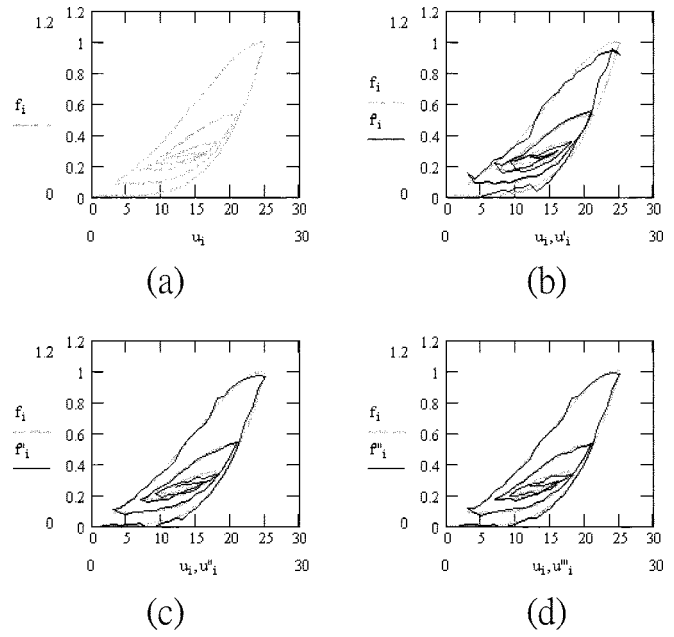


Fig. 7. Training the PNU organization to follow the loops given by (20) in a nonlinear form $\mu(\alpha, \beta) = (2/25^4)\alpha^2 + (6/25^4)\beta^2$. (a) Target diagram, with 111 points on it. The following figures are trained after 1000 epochs. According to our results, the more PNUs used implies a better performance that it converges. (b) Four PNUs used, $\text{MSE} = 8.73 \times 10^{-4}$. (c) Six PNUs used, $\text{MSE} = 2.33 \times 10^{-4}$. (d) Eight PNUs used, $\text{MSE} = 1.93 \times 10^{-4}$.

Equation (19) can be implemented as an iterative procedure to adapt the system parameters.

E. Adaptation to Given Behavior

As generally known, trajectory traversal is the foundation of system modeling. In this section, the PNU organizational ability is demonstrated to learn some loops. Desired tracks are generated by the Preisach model (see Section II). Initially, a system with four PNUs is trained to follow these trajectory samples. These four PNUs are bounded with distinct sensible ranges as $[0, 25]$, $[5, 20]$, $[3, 13]$, and $[12, 22]$. In addition, all stage lengths of their reaction functions, which are step functions, are set to be one equally. To normalize the response value in $[0, 1]$, we merely assign average weight parameters $(1/4 \cdot (25 - 0))$, $(1/4 \cdot (20 - 5))$, $(1/4 \cdot (13 - 3))$, and $(1/4 \cdot (22 - 12))$ to each stage of these four PNUs as the initial parameters. In doing so, the adaptation can be clearly observed.

Equation (1) displays how the Preisach model functions. However, an adjustment must be made such that whole positive trajectories can be generated for the PNU organization to simulate. Regarding (1)

$$f(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) [\hat{\gamma}_{\alpha\beta} u(t)] d\alpha d\beta$$

we redefine the target diagram by having the bi-valued operator $\hat{\gamma}_{\alpha\beta}$ return 0 instead of -1 while the descending paths are followed. Assume that this redefined operator is $\hat{\kappa}_{\alpha\beta}$. The target diagram now follows the equation

$$f(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) [\hat{\kappa}_{\alpha\beta} u(t)] d\alpha d\beta. \quad (20)$$

Next, consider the alternating series $\{0, 25, 3, 21, 7, 18, 9, 16, 12\}$ to perform the target trajectories. All distances are set to "1" sequentially between these extreme values. Therefore, there are cumulatively 111 points on the target sequences. In addition, allow the learning rate to be 0.015. The learning rate is set quite small with respect to the good

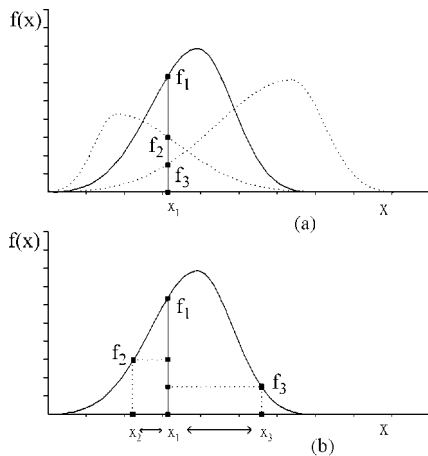


Fig. 8. Hysteresis embedded into a fuzzy membership function. We can take a hysteresis-embedded input instead for alternatively considering several membership functions during a memory-involved inference.

deal of set as weights in such a single layer. Later on, more experiments are performed as well in an attempt to accelerate the learning process.

Under the page constraint, only two experiments performed herein demonstrate the PNU organizational ability of adaptation. Fig. 6 demonstrates the learning capability of the trajectory with $\mu(\alpha, \beta) = (4.5/25^2) - (3/25^3)\alpha - (1/25^3)\beta$ in (20). The adaptive performance appears acceptable. This same figure indicates that using more PNUs carried facilitates a better performance. While two more PNUs are increased with sensible ranges of [17, 23], [19, 22], it converges faster and better, as indicated by Fig. 6(f). Fig. 7 reveals a similar observation when we trace nonlinear parameters, $\mu(\alpha, \beta) = (2/25^4)\alpha^2 + (6/25^4)\beta^2$. Fig. 7(b) is traced by the original four PNUs, system in Fig. 7(c) is joined with additional two units with sensible ranges of [17, 23], [19, 22]. In addition, the system in Fig. 7(d) adopts two more units bounded in [7, 15], [11, 16].

V. CONCLUSIONS

Four memory-related topics are frequently studied in engineering: memory kernels in convolution neural models [13], recursive efforts of recurrent networks [31], [52], delayed rewards in reinforcement learning [50], and rate-independent memory effect such as hysteresis. In this study, we closely examine the final one.

While focusing mainly on defining hysteretic memory, this study discusses whether or not network computation can be applied to hysteresis modeling. Analysis results indicate that networks with purely computational nodes and links cannot function as hysteresis simulators. A propulsive neural model is also proposed to construct hysteretic memory. The proposed model is based on a neural unit, the neuron. In addition to propulsive operation, a single neuron functions similarly to a one-layered perceptron. In addition, several propulsive neural units with distinct sensible ranges are used to organize a simple system. Finally, a learning method based on backpropagation is designed so that the system can automatically adapt its parameters.

Based on the results presented herein, the areas for future research are made as follows.

A. Hysteretic Memory-Related Applications

Hysteresis, although it is a memory effect whose output must consult with historic inputs, responds to the tendency in real time. A (local) highest value cannot be input and expected to receive the co-relative output after some time when the input value is decreasing. Restated, hysteretic memory would not incur aftereffects and delayed reactions.

On the other hand, a mechanism that is used to model delayed reactions may not adequately process hysteretic memory. In fact, the varied input speed is actually a challenge for all of those mechanisms, while it does not influence a hysteresis system owing to rate independence. Therefore, developing a feasible means of combining hysteretic memory with conventional time varying signal processors is of worthwhile interest.

Embedding hysteretic memory into a fuzzy membership description is another application. The fuzzy inference system must represent the membership of each condition before a decision can be reached. However, a static membership function may not function properly for a memory-involved inference. Alternatively considering several membership functions is an effective means of resolving this problem. A hysteresis embedded input can be developed instead, after we have developed a hysteresis model, as indicated by Fig. 8.

B. Improvement of System Performance

This study has presented a supervised learning method to adjust the system parameters. A future work should focus on how to self-organize the system structure so that the system identification task is completed. The genetic algorithm (GA) is a viable means of generating the structure.

Another point worth mentioning is that hysteresis does not only occur in one feature in a one-dimensional model. Hysteretic memory may appear in several coupled features with correlation. An interesting topic would be how we can more fully elucidate propulsion across these dimensions.

Particular attention should also be paid to the reaction function. This function is closely related to how the neuron operates and is adapted. We believe that the system performance could be markedly improved if a new reaction could be designed to function well. By doing so, construction of hysteretic memory would not be so complicated.

Notably, network approximation in other studies [3], [36] has functioned as a cooperator with particular hysteresis models. They also perform well in the aspect of engineering. In contrast, this work deals with hysteresis behavior in a unique manner. Results presented herein allow us to construct hysteretic memory in neural networks. This is also a novel means of imaging how a system generates hysteresis behaviors since the system is always a black box itself.

REFERENCES

- [1] K. Bhattacharya, R. D. James, and P. J. Swart, "Relaxation in shape-memory alloys. i. Mechanical model," *Acta. Mater.*, vol. 45, pp. 4547–4560, Nov. 1997.
- [2] —, "Relaxation in shape-memory alloys. ii. Thermo-mechanical model and proposed experiments," *Acta. Mater.*, vol. 45, pp. 4561–4568, Nov. 1997.
- [3] S. Bobbio, G. Miano, C. Serpico, and C. Visone, "Models of magnetic hysteresis based on play and stop hysteresis," *IEEE Trans. Magn.*, vol. 33, pp. 4417–4426, Nov. 1997.
- [4] M. Brokate, N. Kenmochi, I. Müller, J. F. Rodriguez, and C. Verdi, *Phase Transitions and Hysteresis*. New York: Springer-Verlag, 1993.
- [5] J. Cacko, "Simulation experiments in investigation of fatigue degradation effects of dynamic processes," in *Eur. Simulation Multiconf. 1995*, 1995, pp. 92–96.
- [6] H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*. New York: Wiley, 1985.
- [7] C. M. I. Campbell and K. S. Kamrani, "The reasons for wage rigidity: Evidence from a survey of firms," *Q. J. Econ.*, pp. 759–789, Aug. 1997.
- [8] S. Can, M. S. Richards, and R. A. Pease, "A 3 v thermostat circuit," in *Proc. 1996 BIPOLAR/BiCMOS Circuits Technol. Meeting*, 1996, pp. 105–108.
- [9] C. R. Chang and I. Klik, "Thermal relaxation in magnetic materials," *Proc. Natl. Sci. Council. Rep. China A, Phys. Sci. Eng.*, vol. 20, pp. 249–264, May 1996.
- [10] B. J. Chapman, "Long-term unemployment: The dimensions of the problem," *Aust. Econ. Review (1993 Second Quarter)*, pp. 22–25, 1993.

- [11] S. E. Christophe, "Hysteresis and the value of the U.S. multinational corporation," *J. Bus.*, pp. 435–462, July 1997.
- [12] J. M. Cruz-Hernandez and V. Hayward, "On the linear compensation of hysteresis," in *Proc. 36th IEEE Conf. Decision Contr.*, vol. 2, 1997, pp. 1956–1957.
- [13] B. de Vries and J. C. Principe, "The Gamma model—A new neural model for temporal processing," *Neural Networks*, vol. 5, pp. 565–576, 1992.
- [14] F. fur Informatik, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.
- [15] F. P. Gnadinger, "Ferroelectrics for nonvolatile RAMs," *IEEE Spectrum Mag.*, pp. 30–33, July 1989.
- [16] C. W. J. Granger and Z. Ding, "Varieties of long memory models," *J. Econometrics*, pp. 61–77, July 1996.
- [17] D. C. Jiles, "Frequency dependence of hysteresis curves in conducting magnetic materials," *J. Appl. Phys.*, vol. 76, pp. 5849–5855, Nov. 1994.
- [18] —, "Modeling the effects of eddy current losses on frequency dependent hysteresis in electrically conducting media," *IEEE Trans. Magn.*, vol. 30, pp. 4326–4328, Nov. 1994.
- [19] D. C. Jiles and D. L. Atherton, "Theory of ferromagnetic hysteresis (invited)," *J. Appl. Phys.*, vol. 55, pp. 2115–2120, Mar. 1984.
- [20] K. Jin'no and T. Saito, "Analysis and synthesis of continuous-time hysteretic neural networks," *Electron. Commun. Jpn.*, pt. 3, vol. 76, no. 4, 1993.
- [21] J. F. Jørgensen, K. Carneiro, and L. L. Madsen, "Hysteresis correction of scanning tunneling microscope images," *J. Vac. Sci. Technol. B.*, vol. 12, pp. 1702–1704, May/June 1994.
- [22] M. Kandil, "Sticky wage or sticky price? Analysis of the cyclical behavior of the real wage," *Southern Econ. J.*, pp. 440–459, Oct. 1996.
- [23] C. Koiter, "General theorems for elastic-plastic solids," in *Progress in Solid Mechanics*, I. N. Sneddon and R. Hill, Eds, Amsterdam, The Netherlands: North-Holland, 1960, vol. 1, pp. 165–221.
- [24] M. Krasnoselskii and A. Pokrovskii, *Systems with Hysteresis*, Moscow, Russia: Nauka, 1983.
- [25] D. T. Lin, J. E. Dayhoff, and P. A. Ligomenides, "Trajectory production with the adaptive time-delay neural network," *Neural Networks*, vol. 8, no. 3, pp. 447–461, 1995.
- [26] I. D. Mayergoyz, *Mathematical Models of Hysteresis*. New York: Springer-Verlag, 1991.
- [27] J. Millman and A. Grabel, *Microelectronics*, 2nd ed. New York: McGraw-Hill, 1987.
- [28] Y. Mualem, Theory of Universal Hysteretical Properties of Unsaturated Porous Media, Water Resour. Res. Public., 1979.
- [29] I. Nagy and Z. Suto, "Repetitive and chaotic processes in current controlled induction motor," in *ISIE'96. Proc. IEEE Int. Symp. Ind. Electron.*, vol. 2, 1996, pp. 946–951.
- [30] D. Obradovic, "On-line training of recurrent neural networks with continuous topology adaptation," *IEEE Trans. Neural Networks*, vol. 7, pp. 222–228, Jan. 1996.
- [31] O. Omidvar and D. L. Elliott, *Neural Systems for Control*. New York: Academic, 1997.
- [32] V. Petridis and A. Kehagias, "A recurrent network implementation of time series classification," *Neural Comput.*, pp. 357–372, Feb. 1996.
- [33] F. Preisach, "Über die magnetische nachwirkung," *Z. Physik*, vol. 94, pp. 277–302, 1935.
- [34] L. Rayleigh, "On the behavior of iron and steel under the operation of feeble magnetic forces," *Philos. Mag.*, vol. 1887, pp. 225–248, 1887.
- [35] G. Rieder, "Elastic obstacles for bloch walls," in *Models of Hysteresis*, A. Visintin, Ed. Harlow, London, U.K.: Longman, 1993, pp. 143–157.
- [36] C. Serpico and C. Visone, "Magnetic hysteresis modeling via feed-forward neural networks," *IEEE Trans. Magn.*, vol. 34, no. 3, pp. 623–628, May 1998.
- [37] R. C. Smith, "Hysteresis modeling in magnetostrictive materials via Preisach operators," *Inst. Comput. Applicat. Sci. Eng., Tech. Rep. TR-97-23*, May 1997.
- [38] N. L. Stein, P. A. Ornstein, B. Tversky, and C. Brainerd, *Memory for Everyday and Emotional Events*: Lawrence Erlbaum Associates, 1997.
- [39] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, pp. 9–44, 1988.
- [40] D. W. Tank and J. J. Hopfield, "Concentrating information in time: Analog neural networks with applications to speech recognition problems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 328–339, Mar. 1989.
- [41] G. Tao and P. V. Kokotovic, "Adaptive control of plants with unknown hysteresis," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 200–212, Feb. 1995.
- [42] M. D. Tom and M. F. Tenorio, "A neural computation model with short-term memory," *IEEE Trans. Neural Networks*, vol. 6, pp. 387–397, Mar. 1995.
- [43] E. D. Toore and F. Vajda, "Parameter identification of the complete-moving hysteresis model using major loop data," *IEEE Trans. Magn.*, vol. 30, pp. 4987–5000, Nov. 1994.
- [44] C. Truesdell and W. Noll, *The Nonlinear Field Theories of Mechanics. In Encyclopedia of Physics*, Heidelberg, Germany: Springer-Verlag, 1965, vol. III/3.
- [45] M. Uribe, "Hysteresis in a simple model of currency substitution," *J. Monetary Econ.*, pp. 185–202, Sept. 1997.
- [46] F. Vajda and E. D. Torre, "Remanence loop asymmetry and moving-type Preisach models," *IEEE Trans. Magn.*, vol. 30, pp. 3658–3660, Nov. 1994.
- [47] A. Visintin, *Differential Models of Hysteresis*. New York: Springer-Verlag, 1994.
- [48] S. Wadhwa, R. Caprihan, and S. Kumar, "Performance of a hysteresis based control strategy for a flexible machine operating under a periodic status monitoring policy," *Comput. Ind. Eng.*, pp. 557–574, July 1997.
- [49] C. J. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, pp. 9–44, May 1992.
- [50] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1990, ch. 3.
- [51] —, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
- [52] —, *The Roots of Backpropagation*. New York: Wiley-Interscience, 1994.
- [53] R. J. Williams, "Adaptive state representation and estimation using recurrent connectionist networks," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1990, ch. 4.
- [54] L. Wu and J. Moody, "A smoothing regularizer for feedforward and recurrent neural networks," *Neural Comput.*, vol. 8, pp. 461–489, Apr. 1996.
- [55] H. Yanai and Y. Sawada, "Associative memory network composed of neurons with hysteretic property," *Neural Networks*, vol. 3, pp. 223–228, 1990.
- [56] M. A. Zohdy, M. Karam, and H. S. A. A. Zohdy, "A recurrent dynamic neural network for noisy signal representation," *Neurocomputing*, vol. 17, pp. 77–97, Oct. 1997.