

- [22] S. Xu, J. Lam, and D. W. C. Ho, "A new LMI condition for delay-dependent asymptotic stability of delayed Hopfield neural networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 3, pp. 230–234, Mar. 2006.
- [23] C.-J. Cheng, T.-L. Liao, J.-J. Yan, and C.-C. Hwang, "Globally asymptotic stability of a class of neutral-type neural networks with delays," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1191–1195, Oct. 2006.
- [24] K. Yuan, J. Cao, and H.-X. Li, "Robust stability of switched Cohen-Grossberg neural networks with mixed time-varying delays," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1356–1363, Dec. 2006.
- [25] C. C. Hua, C. N. Long, and X. P. Guan, "New results on stability analysis of neural networks with time-varying delays," *Phys. Lett. A*, vol. 352, pp. 335–340, 2006.
- [26] Y. He, G. P. Liu, and D. Rees, "New delay-dependent stability criteria for neural networks with time-varying delay," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 310–314, Jan. 2007.
- [27] Y. He, M. Wu, J. H. She, and G. P. Liu, "Delay-dependent robust stability criteria for uncertain neutral systems with mixed delays," *Syst. Control Lett.*, vol. 51, no. 1, pp. 57–65, 2004.
- [28] Y. He, M. Wu, J. H. She, and G. P. Liu, "Parameter-dependent Lyapunov functional for stability of time-delay systems with polytopic-type uncertainties," *IEEE Trans. Autom. Control*, vol. 49, no. 5, pp. 828–832, May 2004.
- [29] M. Wu, Y. He, J. H. She, and G. P. Liu, "Delay-dependent criteria for robust stability of time-varying delay systems," *Automatica*, vol. 40, no. 8, pp. 1435–1439, 2004.
- [30] X. J. Jing, D. L. Tan, and Y. C. Wang, "An LMI approach to stability of systems with severe time-delay," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1192–1195, Jul. 2004.

Neural Networks for Improved Tracking

Leonid I. Perlovsky and Ross W. Deming

Abstract—In this letter, we have developed a neural network (NN) based upon modeling fields for improved object tracking. Models for ground moving target indicator (GMTI) tracks have been developed as well as neural architecture incorporating these models. The neural tracker overcomes combinatorial complexity of tracking in highly cluttered scenarios and results in about 20-dB (two orders of magnitude) improvement in signal-to-clutter ratio.

Index Terms—Combinatorial complexity, ground moving target indicator (GMTI) radar, multitarget tracking, neural networks (NNs).

I. INTRODUCTION

Ground moving target indication (GMTI) radar signals have been used for tracking since the 1990s [1]. When clutter is strong, so that

Manuscript received September 29, 2006; revised January 25, 2007 and April 17, 2007; accepted May 1, 2007. This work was supported by the Lab. Task 05SN02COR, PM Dr. J. Sjogren.

L. I. Perlovsky is with the Harvard University, Cambridge, MA 02138 USA and with the Air Force Research Laboratory, Sensors Directorate (AFRL/SNHE), Hanscom AFB, MA 01731 USA (e-mail: leonid@seas.harvard.edu; Leonid.Perlovsky@hanscom.af.mil).

R. W. Deming is with the General Dynamics IT, Inc., Air Force Research Laboratory, Sensors Directorate, Hanscom AFB, MA 01731 USA (e-mail: ross.deming@hanscom.af.mil).

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2007.903143

signals are below clutter, detection and tracking are difficult. The Cramér–Rao bound for tracking [2] indicates that performance of the state-of-the-art algorithms is significantly below the information-theoretic limit.

The current GMTI detection and tracking subsystem operates in a two-step process. First, Doppler peaks are detected that exceed a pre-determined threshold. Second, these potential target peaks are used to initiate tracks. This two-step procedure is a state-of-the-art approach which is currently used by most tracking systems. The limitation of this procedure is determined by the detection threshold. If the threshold is reduced, the number of detected peaks grows quickly. Increased computer power does not help because the processing requirements are combinatorial in terms of the number of peaks, so that a tenfold increase in the number of peaks results in a million fold increase in the required computer power.

The limiting factor for existing tracking algorithms is combinatorial complexity (CC) and not the absence of information in the data. CC has been encountered since the 1950s in various applications and was manifest in a variety of mathematical techniques. An early case was Bellman's discussion of the "the curse of dimensionality" [3]. The following 30 years of developing self-learning mathematical techniques led to a conclusion that these approaches often encountered CC of learning requirements. The required examples had to account for all possible variations of "an object," in all possible combinations with other objects, conditions, etc. The number of combinations quickly grows with the number of objects and conditions; say, a medium complexity problem of recognizing 100 different objects in various combinations might lead to a need to learn 100^{100} combinations; this number is larger than the number of all elementary particle interactions in the entire life of the Universe [4].

Initially, tracking algorithms did not face combinatorial complexity [5] and simple track models were used. Kalman filters revolutionized tracking [6] by providing the possibility to use complex track models. However, Kalman filters were designed for tracking single targets. For multiple targets in clutter, radar returns had to be associated either with clutter, or with a track, and, therefore, association algorithms had to be developed. Multiple hypothesis tracking (MHT) [7] is a widely used algorithm which evaluates multiple hypotheses about how the set of radar returns are associated with each track or with clutter. It is well known, however, to face CC [8]. To alleviate the CC of MHT, the probabilistic data association (PDA) algorithm was developed [9]. However, it can only maintain tracks—it cannot be used to initiate tracks in clutter.

Neural algorithms for tracking were developed based on Widrow's adaline [10], a pioneering neural network (NN) combining neural architecture and a model-based structure suitable for tracking a single target. In this letter, a neural architecture is developed for tracking multiple targets in clutter, while avoiding combinatorial complexity. The tracker is tailored for GMTI data, and it follows the neural modeling fields architecture described in [8].

II. NEURAL MODELING FIELDS FOR GMTI TRACKING

For GMTI tracking, in neural modeling fields (NMF) theory [8], each input neuron $n = 1, \dots, N$, encodes four values: range and cross-range positions of cell $n(x_n, y_n)$ and radar return parameters amplitude and Doppler (a_n, D_n); we denote $\mathbf{X}(n) = (x_n, y_n, a_n, D_n)$. Every radar return is also characterized by time t_n . We call an input neural field a set of $\mathbf{X}(n)$; it is a set of bottom-up input signals. Top-down, or priming signals to these neurons, are generated by models, clutter and track models $\mathbf{M}_h(S_h, n)$ enumerated by index $h = 1, \dots, H$. Each

model is characterized by its parameters S_h . In this letter, we consider targets moving with constant velocities (v_{x_h}, v_{y_h})

$$\mathbf{M}_h(S_h, n) = (x0_h + v_{x_h}t_n, y0_h + v_{y_h}t_n, a_h, D_h). \quad (1)$$

These models predict expected positions of targets. The NN compares these predictions to the data and generates parameter update signals as described later. Model parameters $S_h = (x0_h, v_{x_h}, y0_h, v_{y_h}, a_h, D_h)$ characterize position, velocity, average amplitude, and Doppler of the target, also $D_h = v_{x_h}$. These parameters are not known and are to be estimated through the internal dynamics of the NMF NN.

Interaction between bottom-up and top-down signals is determined by neural weights. First, define similarity measures between bottom-up signals $\mathbf{X}(n)$ and top-down signals \mathbf{M}_h

$$l(n|h) = (2\pi)^{-2} (\det C_h)^{-0.5} \cdot \exp\{-0.5(\mathbf{X}(n) - \mathbf{M}_h^T)C_h^{-1}(\mathbf{X}(n) - \mathbf{M}_h)\}. \quad (2)$$

Here, we use a Gaussian function with the mean \mathbf{M}_h and a diagonal covariance $C_h = \text{diag}(\sigma_{x_h}^2, \sigma_{y_h}^2, \sigma_{a_h}^2, \sigma_{D_h}^2)$. Second, we compute neural weights

$$f(h|n) = \frac{r(h)l(n|h)}{\sum_{h' \in H} r(h')l(n|h')}. \quad (3)$$

Here, rates $r(h)$ are the relative numbers of radar returns associated with model h . For object models, we assume that a single radar return is received from each scan; this is compatible with a low setting of threshold for radar signal detections. The number of radar scans k is known, therefore, $r(h) = k$ are known quantities. We denote the clutter model by $h = 1$; it is characterized by average amplitude, Doppler, and rate

$$M_1(S_1, n) = (a_1, D_1) \quad r(1) = \frac{(N - H)k}{N}. \quad (4)$$

Model predictions are computed by modeling neurons, therefore, the total number of neurons is the number of data points N plus the number of models H .

NMF NN iterations [8] start with an initial guess using known minimum and maximum values of coordinates X_{\min} , X_{\max} , Y_{\min} , and Y_{\max} . The maximum absolute values of velocities in either direction (x, y) are 100 mi/h, or 44.7 m/s, $V_{\min} = -44.7$ m/s, and $V_{\max} = 44.7$ m/s. Amplitude is normalized to $A_{\min} = 0$ and $A_{\max} = 1$. Target initial parameter values are taken randomly between the minimum and maximum values. Clutter initial values are taken as zero for Doppler and 0.5 for amplitude. Initial values of the standard deviations σ_{initial} are estimated using all radar returns, using the same values for all models. Next, NMF computes neural weights (3) and then computes new values of the model parameters. Model parameter equations are written using the following notation for a standard neural operation of computing a weighted sum:

$$\langle \dots \rangle_h \sum_{n \in N} f(h|n) (\dots)_n. \quad (5)$$

Using these notations, parameters are computed at each iteration as

$$a_h = \langle a_n \rangle_h \quad (6)$$

$$y0_h \langle 1 \rangle_h + v_{y_h} \langle t_n \rangle_h = \langle y_n \rangle_h \quad (7)$$

$$y0_h \langle t_n \rangle_h + v_{y_h} \langle t_n^2 \rangle_h = \langle y_n t_n \rangle_h$$

$$x0_h \langle 1 \rangle_h + v_{x_h} \langle t_n \rangle_h = \langle x_n \rangle_h$$

$$x0_h \langle t_n \rangle_h + v_{x_h} \langle t_n^2 \rangle_h + c \langle 1 \rangle_h = \langle x_n t_n \rangle_h + c \langle D_n \rangle_h. \quad (8)$$

Here, $c = \sigma_{x_h}^2 / \sigma_{D_h}^2$. For the unknown parameters $y0_h$ and v_{y_h} , (7) is a linear system of equations; similarly, (8) is a 2-D linear system of equations for $x0_h$ and v_{x_h} . After parameters are computed as previously described, standard deviations are computed as follows:

$$\sigma = \sigma_{\text{initial}} \exp\left\{\frac{-7i}{IT}\right\} + \sigma_{\text{final}}. \quad (9)$$

Here, σ_{initial} and σ_{final} are initial and final values of standard deviation. The initial value is defined using standard statistical procedure [11] from all available data. The final value is defined by the radar measurement errors. The exponential factor is defined so that at the end of iterations the first item is small; i is the iteration number and IT is the total number of iterations; it is set from experience. Usually, it is between 10 and 100; results are not very sensitive and few trials are sufficient. Number of tracks is also an unknown parameter estimated from the data. Here, we describe a simple procedure that we used in the example in Section III. We start with a larger number of track models than is actually expected. After every five iterations, if estimated tracks come closer to each other than two standard deviations, one of them is removed and reinitialized randomly. After convergence ($i = IT$), for each track, we compute detection measures defined as a local log-likelihood ratio (LLR) computed using returns within two standard deviations $\{n'\}$ of each track

$$\text{LLR}(h) = \sum_{n' \in N'} [\log l(n'|h) - \log l(n'|1)]. \quad (10)$$

Convergence of this iterative procedure to a local maximum of similarity measure (2) was proven in [8]. Such local convergence usually occurs within relatively few iterations; a typical example in Section III took 20 iterations. Since similarity is a highly nonlinear function, regular convergence to the global maximum cannot be expected. The local rather than global convergence sometimes presents an irresolvable difficulty in many applications. In the presented method, this problem is resolved in three ways. First, the large initial standard deviation of the similarity measure smooths local maxima. Second, according to the previous description [before (10)], a large number of tracks is used initially; many of them are reinitiated several times. Therefore, if a particular real track is not "captured" after few iterations, it will be captured at a later iteration, after track reinitialization. Third, some of the initiated track models converge to spurious events not corresponding to real tracks; say, they will come nearby only two data points. In these cases, local LLR (10) will be low and spurious events are discarded. A detailed characterization of performance usually requires operating curves [12], plots of probability of detection versus probability of false alarm, computed for various signal-to-clutter ratios, densities of targets, target velocities, and other scenario parameters. Such detailed characterization is beyond the scope of this letter. We would just add that the NMF tracker performance came close to the Cramér–Rao bound for tracking [2] in several cases, when we performed such an investigation; these studies will be published separately. Finally, in practical applications, detecting a track is only a part of the overall tracking procedure. The detection procedure described in this section and illustrated in Section III is performed many times, as new data are acquired. Detected short-term tracks (tracklets) are connected into long-term tracks. In this process, spurious events are discarded, and tracks not detected initially are detected at a later time. This procedure, however, is not a subject of this letter. Graphical representation of the described NMF architecture is shown in Fig. 1.

The computational complexity of the procedure described in this section is proportional to the number of data points and the number of tracks $\text{const} \times N \times H$. The const here accounts for the number of iterations, and for complexity of procedures described by (1)–(9). Typical numbers are discussed in Section III. The principal theoretical moment

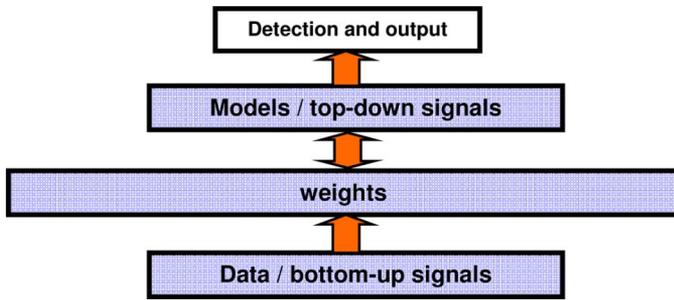


Fig. 1. Graphical representation of the NMF architecture.

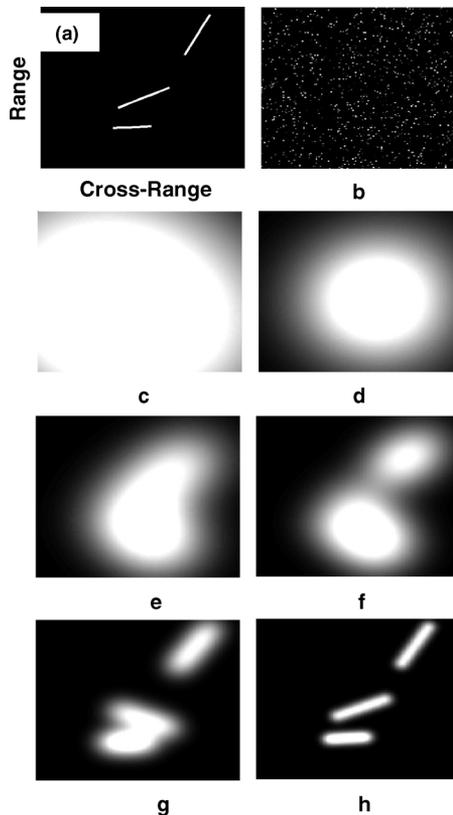


Fig. 2. Detection and tracking objects below clutter using NMF: (a) true track positions in $0.5 \text{ km} \times 0.5 \text{ km}$ data set; (b) actual data available for detection and tracking. Evolution of the NMF NN is illustrated in (c)–(h), where (c) shows the initial, uncertain model and (h) shows the model upon convergence after 20 iterations. Note the close agreement between the converged model (h) and the truth (a).

is that this number is linear in N and in H , rather than combinatorial $\sim H^N$ like in MHT.

III. TRACKING EXAMPLE

An application example of the NMF technique described previously is illustrated in Fig. 2, where detection and tracking are performed on targets below the clutter level. Fig. 2(a) shows true track positions in a $0.5 \text{ km} \times 0.5 \text{ km}$ data set, while Fig. 2(b) shows the actual data available for detection and tracking. In this data, the target returns are buried in the clutter, with signal-to-clutter ratio of about -2 dB for amplitude and -3 dB for Doppler. Here, the data is displayed such that all six revisit scans are shown superimposed in the $0.5 \text{ km} \times 0.5 \text{ km}$ area, 500 predicted signals per scan, and the brightness of each data

sample is proportional to its measured Doppler value. Fig. 2(c)–(h) illustrates the evolution of the NMF model as it adapts during increasing iterations. Here, Fig. 2(c) shows the initial vague-fuzzy model, while Fig. 2(h) shows the model upon convergence at 20 iterations. Between Fig. 2(c) and (h), the NMF NN automatically decides how many model components are needed to fit the data, and simultaneously adapts the model parameters, including target track parameters. There are two types of models: one uniform model describing clutter (it is not shown), and linear track models with large uncertainty; the number of track models is determined from data as described in Section II. In Fig. 2(c) and (d), the NMF NN fits the data with one model, and uncertainty is somewhat reduced. Between Fig. 2(d) and (e), NMF uses more than one track model and decides that it needs two models to “understand” the content of the data. Fitting with two tracks continues until Fig. 2(f); between Fig. 2(f) and (g), a third track is added. Iterations stop at Fig. 2(h), when similarity stops increasing. Detected tracks closely correspond to the truth [Fig. 2(a)].

Fig. 2 illustrations are directly related to values of the model parameters described in Section II. The brightness of the figure is proportional to neural weights (3). Each track begins at $(x0_h, y0_h)$ and extends to $(x0_h + vx_h t_N, y0_h + vy_h t_N)$. Thickness of tracks is determined by standard deviation σ .

The number of neurons is dominated by the number of data neurons; their total number is $500 \times 6 = 3000$. The number of modeling neurons varied during iterations with the number of models, most of the time; as we can see in Fig. 2 frames, there were four modeling neurons, one for clutter and three for track models.

In this example, target signals are below clutter. A single scan does not contain enough information for detection. Detection should be performed concurrently with tracking, using several radar scans, and six scans are used. In this case, a standard multiple hypothesis tracking, evaluating all tracking association hypothesis, would require about 10^{1500} operations, a number too large for computation. Therefore, tracking requires strong signals, with about 15-dB signal-to-clutter ratio [1]. NMF successfully detected and tracked all three targets and required only 10^6 operations, achieving about 18-dB improvement in signal-to-clutter sensitivity.

In case of low clutter (signal-to-clutter ratio above 15 dB), MHT can successfully detect tracks and in this case accuracy of MHT tracks is expected to be comparable to NMF, when tracks are well separated. If probability densities of radar measurements are known, such estimates will be unbiased. When tracks overlap, MHT is expected to produce biased results. The reason is that NMF estimation is not true maximum likelihood, but conditional on hypotheses about correct data-to-track assignments. For close tracks, assignments are biased, therefore, estimations are biased. If probability densities are used for similarities, MHT produces true maximum-likelihood estimation, including maximum-likelihood assignments, and therefore, they are expected to be unbiased (for many shapes of probability densities). However, this bias is a relatively small problem. The real advantage of NMF is its ability to track in heavy clutter, improving upon other tracking algorithms by orders of magnitude; in other words, solving previously unsolvable problems.

IV. FUTURE DIRECTIONS

Future research will include feature-added tracking when—in addition to amplitude, position, and velocity—other characteristics of received signals are also used for improved associations between signals and track models. The NMF NN can naturally incorporate this additional information. Since association neural weights in NMF are functions of object models (1) any object feature can be included into the models and will be used for signal-model associations.

Other sources of information can be included. For example, coordinates of roads can be easily incorporated into the NMF procedure. For this purpose, road positions should be characterized by a probability density, depending on the known coordinates and expected errors. Then, similarities (2) can be modified by multiplying them by the probability densities of roads.

REFERENCES

- [1] J. Jones, J. Bradstreet, M. Kozak, T. Hughes, and M. Blount, "Ground moving target tracking and exploitation performance measures," Pentagon Rep. A269234, 2004, approved for public release.
- [2] L. I. Perlovsky, "Cramer-Rao bound for tracking in clutter and tracking multiple objects," *Pattern Recognit. Lett.*, vol. 18, pp. 283–288, 1997.
- [3] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ: Princeton Univ. Press, 1961.
- [4] L. I. Perlovsky, "Conundrum of computational complexity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 666–670, Jun. 1998.
- [5] N. Wiener, *Cybernetics*. New York: Wiley, 1948.
- [6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. D82, pp. 35–45, 1960.
- [7] R. A. Singer, R. G. Sea, and R. B. Housewright, "Derivation and evaluation of improved tracking filters for use in dense multitarget environments," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 4, pp. 423–432, Jul. 1974.
- [8] L. I. Perlovsky, *Neural Networks and Intellect*. New York: Oxford Univ. Press, 2001.
- [9] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, pp. 451–460, 1975.
- [10] B. Widrow, "Adaptive sample-data systems – A statistical theory of adaptation," in *1959 WESCON Convention Record*, 1959, pp. 74–85.
- [11] Anderson, Sweeney, and Williams, *Statistics for Business and Economics*, 9th ed. New York: Academic, 2006.
- [12] [Online]. Available: http://en.wikipedia.org/wiki/Receiver_operating_characteristic. Jun. 2007

Noise-Induced Stabilization of the Recurrent Neural Networks With Mixed Time-Varying Delays and Markovian-Switching Parameters

Yi Shen and Jun Wang

Abstract—The stabilization of recurrent neural networks with mixed time-varying delays and Markovian-switching parameters by noise is discussed. First, a new result is given for the existence of unique states of recurrent neural networks (NNs) with mixed time-varying delays and Markovian-switching parameters in the presence of noise, without the need to satisfy the linear growth conditions required by general stochastic

Manuscript received February 28, 2007; revised May 7, 2007; accepted May 21, 2007. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CUHK4165/03E and the Natural Science Foundation of China under Grant 60574025.

Y. Shen is with the Department of Control Science and Engineering and the Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China (e-mail: yishen64@163.com).

J. Wang is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (e-mail: jwang@mae.cuhk.edu.hk).

Digital Object Identifier 10.1109/TNN.2007.903159

Markovian-switching systems. Next, a delay-dependent condition for stabilization of concerned recurrent NNs is derived by applying the Itô formula, the Gronwall inequality, the law of large numbers, and the ergodic property of Markovian chain. The results show that there always exists an appropriate white noise such that any recurrent NNs with mixed time-varying delays and Markovian-switching parameters can be exponentially stabilized by noise if the delays are sufficiently small.

Index Terms—Markovian chain, noise, recurrent neural networks (NNs), stabilization, time-varying delay.

I. INTRODUCTION

Time delays in neural signal processing are often unavoidable in the hardware implementation or sometimes desirable in engineering applications of neural networks (NNs). As time delays may cause undesirable instability and the stability of recurrent NNs is usually a prerequisite for successful applications, stability analysis of recurrent NNs with time delays is deemed necessary. In recent years, many new results on the stability analysis of various recurrent NNs with time delays have been reported; e.g., see [1]–[6].

In addition to time delays, the parameters of recurrent NNs are also subject to random and abrupt changes; i.e., the connection weights and biases of recurrent NNs change from one set to another randomly. Besides, in some applications such as real-time control and pattern recognition [7], more than one NNs work cooperatively by switching from one to another. In these cases, NNs have to be modeled as switching systems. The stability of recurrent NNs with switching parameters also deserves in-depth analysis. For example, the stability of a switching Cohen–Grossberg NN [8] with mixed time-varying delays was recently analyzed in [6].

Noise-induced stabilization is an effective approach by which a dynamic system can be stabilized by using a proper white noise [9], [10]. Generally, it is thought that adding noise to an originally stable NN would cause the NN unstable. However, it will be shown in this letter that recurrent NNs with mixed time-varying delays and Markovian-switching parameters can be stabilized by using a suitable white noise, no matter whether it originally is stable or not, provided that the time delays are sufficiently small.

In this letter, we present new results on the stabilization of a general class of NNs by noise. A delay-dependent algebraic criterion on noise will be shown to stabilize the Cohen–Grossberg NN model [8] with mixed time-varying delays and Markovian-switching parameters (e.g., connection weights and biases are subject to changes according to a Markovian chain).

II. PRELIMINARIES

Throughout this letter, unless otherwise specified, let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t \geq 0}$ satisfying the usual conditions (i.e., it is right continuous and \mathcal{F} contains all P -null sets). Let $\omega(t)$ be a scalar Brownian motion defined on the probability space. Let $C([-\tau, 0], \mathbb{R}^n)$ denote the family of continuous functions φ from $[-\tau, 0]$ to \mathbb{R}^n with the norm $\|\varphi\| = \sup_{-\tau \leq \theta \leq 0} |\varphi(\theta)|$, where $\tau > 0$, $|\cdot|$ is the Euclidean norm in \mathbb{R}^n . Let $\mathbb{R}_+ = [0, \infty)$ and $\tau_k : \mathbb{R}_+ \rightarrow [0, \tau]$ ($k = 1, 2$) be a continuous function which stands for the time delay. The operator norm of matrix A is denoted by $\|A\| = \sup\{|Ax| : |x| = 1\}$ (without any confusion with $\|\varphi\|$). Denote $C_{\mathcal{F}_0}^b([-\tau, 0], \mathbb{R}^n)$ as the family of all bounded, \mathcal{F}_0 -measurable, $C([-\tau, 0], \mathbb{R}^n)$ -valued random variables. Let $x(t)$ be a continuous \mathbb{R} -valued stochastic process on $t \in [-\tau, \infty)$, and let $x_t = \{x(t + \theta) : -\tau \leq \theta \leq 0\}$ for $t \geq 0$ which is regarded as a $C([-\tau, 0], \mathbb{R}^n)$ -valued stochastic process.