where $\theta > 0$ is the neuron's threshold, s_t is the bipolar input Bernoulli signal (with success probability $\frac{1}{2}$) with amplitude $A > 0$, and n_t is the additive white noise with probability density $p(n)$. Experiments with other success probabilities near $\frac{1}{2}$ did not produce substantially different simulation results.

C. Noisy Continuous Neuron

This additive neuron model is a bistable system with additive noise: [1], [10], [15], [37], [38], [43]

$$\dot{x} = -x + 2\tanh x + s(t) + n(t) \tag{7}$$

$$y(t) = \text{sgn}(x(t)). \tag{8}$$

where $y(t)$ is the binary output of the system. The neuron feeds its sigmoidal output signal $2\tanh x$ back to itself and emits the threshold bipolar signal $y(t)$ as output.

III. Mutual Information of the Threshold Neuron with Bipolar Input Signals

A. SR in Threshold Neuron

This section derives analytical SR results for the noisy threshold neuron based on the marginal probability density function of the output $P_Y(y)$ and the conditional density $P_{Y|S}(y|s)$. The system is the binary neuron with a fixed threshold θ. The bipolar (Bernoulli with success probability p) input signal s_t has amplitude A: $s_t \in \{-A, A\}$ with probability density $P_S(s)$. The noise n_t adds to the signal s_t before it enters the neuron. So the neuron's output y_t has the form (6). Figure 5 plots the mutual information $I(S,Y)$ for four standard closed-form noise probability density functions (15)-(34). The central result is a theorem that holds for almost all noise probability densities so long as the mean noise falls outside an interval that depends on the threshold θ.

The symbol "0" denotes the input signal $s = -A$ and output signal $y = -1$. The symbol "1" denotes the input signal $s = A$ and output signal $y = 1$. We also assume subthreshold input signals: $A < \theta$. Then the conditional probabilities $P_{Y|S}(y|s)$ are

$$P_{Y|S}(0|0) = Pr\{s + n < \theta\}\Big|_{s=-A} = Pr\{n < \theta + A\} = \int_{-\infty}^{\theta+A} p(n)dn \tag{9}$$

$$P_{Y|S}(1|0) = 1 - P_{Y|S}(0|0) \tag{10}$$

$$P_{Y|S}(0|1) = Pr\{s + n < \theta\}\Big|_{s=A} = Pr\{n < \theta - A\} = \int_{-\infty}^{\theta-A} p(n)dn \tag{11}$$

$$P_{Y|S}(1|1) = 1 - P_{Y|S}(0|1) \tag{12}$$

and the marginal density is

$$P_Y(y) = \sum_s P_{Y|S}(y|s)P_S(s) \tag{13}$$

Researchers have derived the conditional probabilities $P_{Y|S}(y|s)$ of the threshold system with *Gaussian* noise with bipolar inputs [12] and Gaussian inputs [65]. We next derive $P_{Y|S}(y|s)$ for uniform, Laplace, and Cauchy noise as well. Figure 3 shows four examples of the unimodal noise densities and their realizations. Then we introduce stable distributions to model a spectrum of impulsive noise types.

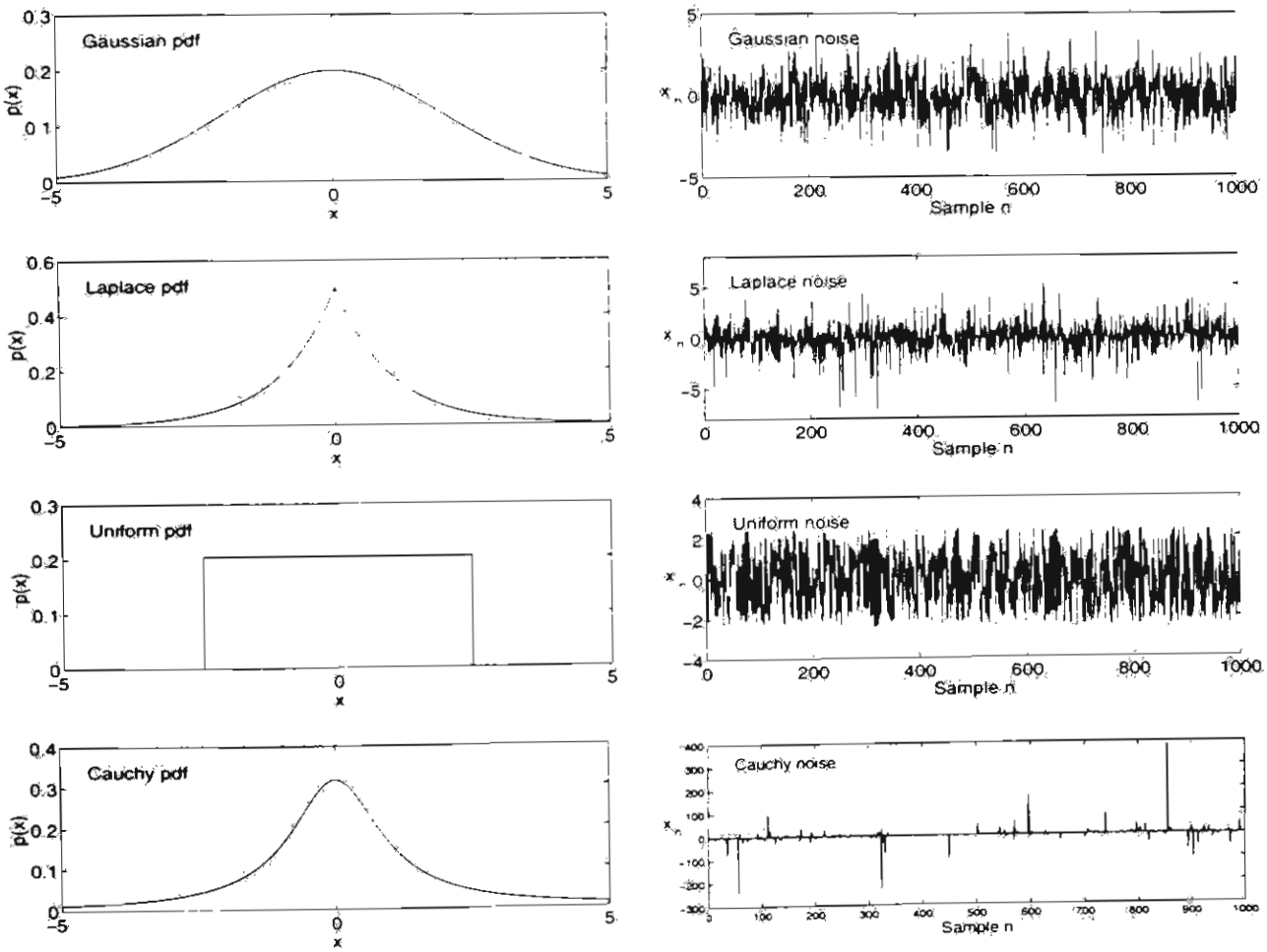

Figure 3. Probability density functions and sample realizations. The figure shows Gaussian, Laplace, and uniform random variables w with zero mean and variance of two: $E[x] = 0$ and $E[x^2] = \sigma^2 = 2$. The Cauchy density function has zero location and unit dispersion. The pseudo-random number generators in [62] act as noise sources for these probability densities.

- **Gaussian Noise**. The Gaussian density with zero mean and variance $\sigma_n^2 = \sigma^2$ has the form

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{n^2}{2\sigma^2}\} \tag{14}$$

Then the conditional probabilities $P_{Y|S}(y|s)$ are

$$P_{Y|S}(0|0) = \int_{-\infty}^{\theta+A} \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{n^2}{2\sigma^2}\} dn = \frac{1}{2} + \frac{1}{2}\text{erf}\frac{\theta + A}{\sigma\sqrt{2}} \tag{15}$$

$$P_{Y|S}(1|0) = \frac{1}{2} - \frac{1}{2}\text{erf}\frac{\theta + A}{\sigma\sqrt{2}} \tag{16}$$

$$P_{Y|S}(0|1) = \frac{1}{2} + \frac{1}{2}\text{erf}\frac{\theta - A}{\sigma\sqrt{2}} \tag{17}$$

$$P_{Y|S}(1|1) = \frac{1}{2} - \frac{1}{2}\text{erf}\frac{\theta - A}{\sigma\sqrt{2}} \tag{18}$$

The error function erf is

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp\{-t^2\}dt \tag{19}$$

• **Uniform Noise**. The uniform density with zero mean and variance $\sigma_n^2 = \frac{a^2}{12}$ has the form

$$p(n) = \begin{cases} \frac{1}{a} & \text{if } -\frac{a}{2} < n < \frac{a}{2} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Then the conditional probabilities $P_{Y|S}(y|s)$ are

$$P_{Y|S}(0|0) = \begin{cases} 1 & \text{if } \frac{a}{2} < \theta + A \\ \frac{1}{2} + \frac{A+\theta}{a} & \text{otherwise} \end{cases} = \min\{1, \frac{1}{2} + \frac{\theta + A}{a}\} \tag{21}$$

$$P_{Y|S}(1|0) = \max\{0, \frac{1}{2} - \frac{\theta + A}{a}\} \tag{22}$$

$$P_{Y|S}(0|1) = \min\{1, \frac{1}{2} + \frac{\theta - A}{a}\} \tag{23}$$

$$P_{Y|S}(1|1) = \max\{0, \frac{1}{2} - \frac{\theta - A}{a}\} \tag{24}$$

• **Laplace Noise**. The Laplace density with zero mean and variance $\sigma_n^2 = 2\beta^2$ has the form

$$p(n) = \frac{1}{2\beta}\exp\{-\left|\frac{n}{\beta}\right|\} \tag{25}$$

Then the conditional probabilities $P_{Y|S}(y|s)$ are

$$P_{Y|S}(0|0) = 1 - \frac{1}{2}\exp\{-\frac{\theta + A}{\beta}\} \tag{26}$$

$$P_{Y|S}(1|0) = \frac{1}{2}\exp\{-\frac{\theta + A}{\beta}\} \tag{27}$$

$$P_{Y|S}(0|1) = 1 - \frac{1}{2}\exp\{-\frac{\theta - A}{\beta}\} \tag{28}$$

$$P_{Y|S}(1|1) = \frac{1}{2}\exp\{-\frac{\theta - A}{\beta}\} \tag{29}$$

• **Cauchy Noise**. The Cauchy density with zero location and finite dispersion γ (but infinite variance) has the form

$$p(n) = \frac{1}{\pi}\frac{\gamma}{n^2 + \gamma^2}. \tag{30}$$

Then the conditional probabilities $P_{Y|S}(y|s)$ are

$$P_{Y|S}(0|0) = \frac{1}{2} + \frac{1}{\pi}\tan^{-1}\frac{\theta + A}{\gamma} \tag{31}$$

$$P_{Y|S}(1|0) = \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\frac{\theta + A}{\gamma} \tag{32}$$

$$P_{Y|S}(0|1) = \frac{1}{2} + \frac{1}{\pi}\tan^{-1}\frac{\theta - A}{\gamma} \tag{33}$$

$$P_{Y|S}(1|1) = \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\frac{\theta - A}{\gamma} \tag{34}$$

• Symmetric Alpha-Stable Noise: Thick-Tailed Bell Curves

We model many types of impulsive noise with symmetric alpha-stable bell-curve probability density functions with parameter α in the characteristic function $\phi(\omega) = \exp\{-\gamma|\omega|^\alpha\}$. Here γ is the *dispersion* parameter [6], [27], [33], [59]. The parameter α controls tail thickness and lies in $0 < \alpha \leq 2$. Noise grows more impulsive as α falls and the bell-curve tails grow thicker. The (thin-tailed) Gaussian density results when $\alpha = 2$ or when $\varphi(\omega) = \exp\{-\gamma\omega^2\}$. So the standard Gaussian random variable has zero mean and variance $\sigma^2 = 2$ (when $\gamma = 1$). The parameter α gives the thicker-tailed Cauchy bell curve when $\alpha = 1$ or $\varphi(\omega) = \exp\{-|\omega|\}$ for a zero *location* ($a = 0$) and unit dispersion ($\gamma = 1$) Cauchy random variable. The moments of stable distributions with $\alpha < 2$ are finite only up to the order k for $k < \alpha$. The Gaussian density alone has finite variance and higher moments. Alpha-stable random variables characterize the class of normalized sums of independent random variables that converge in distribution to a random variable [6] as in the famous Gaussian special case called the "central limit theorem." Alpha-stable models tend to work well when the noise or signal data contains "outliers" — and all do to some degree. Models with $\alpha < 2$ can accurately describe impulsive noise in telephone lines, underwater acoustics, low-frequency atmosphereic signals, fluctuations in gravitational fields and financial prices, and many other processes [44], [59]. Note that the best choice of α is an *empirical* question for bell-curve phenomena. Bell-curve behavior alone does not justify the (extreme) assumption of the Gaussian bell curve.

Figure 4 shows realizations of four symmetric alpha-stable random variables. A general alpha-stable probability density function f has characteristic function φ [2], [5], [33], [59]:

$$\varphi(\omega) = \exp\left\{ia\omega - \gamma|\omega|^\alpha\left(1 + i\beta\,\text{sign}(\omega)\tan\frac{\alpha\pi}{2}\right)\right\} \qquad \text{for } \alpha \neq 1 \tag{35}$$

and

$$\varphi(\omega) = \exp\left\{ia\omega - \gamma|\omega|(1 - 2i\beta\ln|\omega|\text{sign}(\omega)/\pi)\right\} \qquad \text{for } \alpha = 1 \tag{36}$$

where

$$\text{sign}(\omega) = \begin{cases} 1 & \text{if } \omega > 0 \\ 0 & \text{if } \omega = 0 \\ -1 & \text{if } \omega < 0 \end{cases} \tag{37}$$

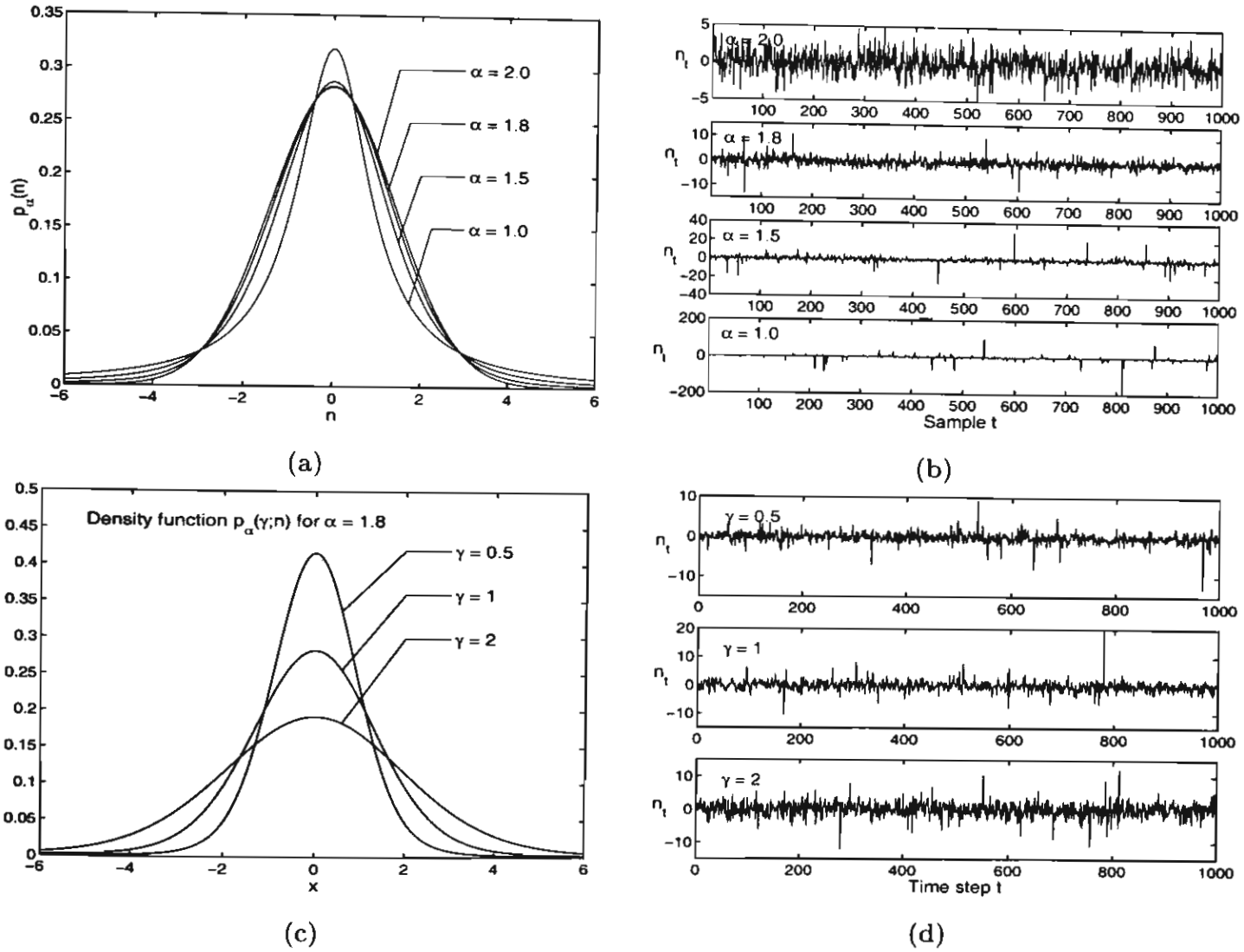

Figure 4. Samples of standard symmetric alpha-stable probability densities and their realizations. (a) Density functions with zero location ($a = 0$) and unit dispersion ($\gamma = 1$) for $\alpha = 2$, 1.8, 1.5, and 1. The densities are bell curves that have thicker tails as α decreases and thus that model increasingly impulsive noise as α decreases. The case $\alpha = 2$ gives a Gaussian density with variance two (or unit dispersion). The parameter $\alpha = 1$ gives the Cauchy density. (b) Samples of alpha-stable random variables with zero location and unit dispersion. The plots show realizations when $\alpha = 2$, 1.8, 1.5, and 1. Note the scale differences on the y-axes. The alpha-stable variable n becomes more impulsive as the parameter α falls. The algorithm in [13],[67] generates these realizations. (c) Density function for $\alpha = 1.8$ with dispersion $\gamma = 0.5$, 1, and 2. (d) Samples of alpha-stable noise n for $\alpha = 1.8$ with dispersions γ =0.5, 1, and 2.

and $i = \sqrt{-1}$, $0 < \alpha \leq 2$, $-1 \leq \beta \leq 1$, and $\gamma > 0$. The parameter α is the characteristic exponent. Again the variance of an alpha-stable density distribution does not exist if $\alpha < 2$. The location parameter a is the "mean" of the density when $\alpha > 1$. β is a skewness parameter. The density is symmetric about a when $\beta = 0$. The theorem below still holds even when $\beta \neq 0$. The dispersion parameter γ acts like a variance because it controls the width of a symmetric alpha-stable bell curve. There are no known closed forms of the α-stable densities for most α's. Numerical integration of φ gives the probability densities above.

The following theorem shows that noisy threshold neurons produce some SR effect for almost all noise probability descriptions. The proof shows that if $I(S,Y) > 0$ then eventually the mutual information $I(S,Y)$ tends toward zero as the noise variance or dispersion tends toward zero. So the mutual information $I(S,Y)$ must increase as the noise variance increases from zero. The crucial assumption is that the noise mean $E[n]$ (or location parameter) not lie in the signal-threshold inverval $[\theta - A, \theta + A]$.

Theorem. Suppose that the threshold signal system (6) has noise probability density function $p(n)$ and that the input signal S is subthreshold ($A < \theta$). Suppose that there is some statistical dependence between input random variable S and output random variable Y (so that $I(S,Y) > 0$). Suppose that the noise mean $E[n]$ does not lie in the signal-threshold interval $[\theta - A, \theta + A]$ if $p(n)$ has finite variance. Suppose that $a \notin [\theta - A, \theta + A]$ for the location parameter a of an alpha-stable noise density with characteristic function (35)-(36). Then the threshold system (6) exhibits the nonmonotone SR effect in the sense that $I(S,Y) \to 0$ as $\sigma \to 0$ or $\gamma \to 0$.

Proof. Assume $0 < P_S(s) < 1$ to avoid triviality when $P_S(s) = 0$ or 1. We show that S and Y are asymptotically independent: $I(\sigma) \to 0$ as $\sigma \to 0$ (or as $\gamma \to 0$). Recall that $I(S,Y) = 0$ if and only if S and Y are statistically independent [19]. So we need to show only that $P_{SY}(s,y) = P_S(s)P_Y(y)$ or $P_{Y|S}(y|s) = P_Y(y)$ as $\sigma \to 0$ (or as $\gamma \to 0$) for some signal symbols $s \in S$ and $y \in Y$. The two-symbol alphabet set S gives

$$
\begin{aligned}
P_Y(y) &= \sum_s P_{Y|S}(y|s)P_S(s) && (38)\\
&= P_{Y|S}(y|0)P_S(0) + P_{Y|S}(y|1)P_S(1) && (39)\\
&= P_{Y|S}(y|0)P_S(0) + P_{Y|S}(y|1)(1 - P_S(0)) && (40)\\
&= (P_{Y|S}(y|0) - P_{Y|S}(y|1))P_S(0) + P_{Y|S}(y|1). && (41)
\end{aligned}
$$

So we need to show only that $P_{Y|S}(y|0) - P_{Y|S}(y|1) = 0$ as $\sigma \to 0$ (or as $\gamma \to 0$). This condition implies that $P_Y(y) = P_{Y|S}(y|1)$ and $P_Y(y) = P_{Y|S}(y|0)$. We assume for simplicity that the noise density $p(n)$ is integrable. The argument below still holds if $p(n)$ is discrete and if we replace integrals with appropriate sums.

Consider y ='0'. Then (9) and (11) imply that

$$P_{Y|S}(0|0) - P_{Y|S}(0|1) = \int_{-\infty}^{\theta+A} p(n)dn - \int_{-\infty}^{\theta-A} p(n)dn \quad (42)$$

$$= \int_{\theta-A}^{\theta+A} p(n)dn \quad (43)$$

Similary for y ='1':

$$P_{Y|S}(1|0) = \int_{\theta+A}^{\infty} p(n)dn \quad (44)$$

$$P_{Y|S}(1|1) = \int_{\theta-A}^{\infty} p(n)dn \quad (45)$$

Then

$$P_{Y|S}(1|0) - P_{Y|S}(1|1) = -\int_{\theta-A}^{\theta+A} p(n)dn \quad (46)$$

The result now follows if we can show that

$$\int_{\theta-A}^{\theta+A} p(n)dn \to 0 \qquad \text{as } \sigma \to 0 \text{ or } \gamma \to 0 \quad (47)$$

Case 1. Finite-variance noise. Let the mean of the noise be $m = E[n]$ and the variance be $\sigma^2 = E[(x-m)^2]$. Then $m \notin [\theta - A, \theta + A]$ by hypothesis.

Now suppose that $m < \theta - A$. Pick $\varepsilon = \frac{1}{2}d(\theta - A, m) = \frac{1}{2}(\theta - A - m) > 0$. So $\theta - A - \varepsilon = \theta - A - \varepsilon + m - m = m + (\theta - A - m) - \varepsilon = m + 2\varepsilon - \varepsilon = m + \varepsilon$. Then

$$P_{Y|S}(0|0) - P_{Y|S}(0|1) = \int_{\theta-A}^{\theta+A} p(n)dn \quad (48)$$

$$\leq \int_{\theta-A}^{\infty} p(n)dn \quad (49)$$

$$\leq \int_{\theta-A-\varepsilon}^{\infty} p(n)dn \quad (50)$$

$$= \int_{m+\varepsilon}^{\infty} p(n)dn \quad (51)$$

$$= Pr\{n \geq m + \varepsilon\} = Pr\{n - m \geq \varepsilon\} \quad (52)$$

$$\leq Pr\{|n - m| \geq \varepsilon\} \quad (53)$$

$$\leq \frac{\sigma^2}{\varepsilon^2} \qquad \text{by Chebyshev's inequality} \quad (54)$$

$$\to 0 \qquad \text{as } \sigma \to 0 \quad (55)$$

Suppose next that $m > \theta + A$. Then pick $\varepsilon = \frac{1}{2}d(\theta + A, m) = \frac{1}{2}(m - \theta - A) > 0$ and so $\theta + A + \varepsilon = \theta + A + \varepsilon + m - m = m - (m - \theta - A) + \varepsilon = m - 2\varepsilon + \varepsilon = m - \varepsilon$. Then

$$P_{Y|S}(0|0) - P_{Y|S}(0|1) = \int_{\theta-A}^{\theta+A} p(n)dn \quad (56)$$

$$\leq \int_{-\infty}^{\theta+A} p(n)dn \quad (57)$$

$$\leq \int_{-\infty}^{\theta+A+\varepsilon} p(n)dn \tag{58}$$

$$= \int_{-\infty}^{m-\varepsilon} p(n)dn \tag{59}$$

$$= Pr\{n \leq m - \varepsilon\} = Pr\{n - m \leq -\varepsilon\} \tag{60}$$

$$\leq Pr\{|n - m| \geq \varepsilon\} \tag{61}$$

$$\leq \frac{\sigma^2}{\varepsilon^2} \qquad \text{by Chebyshev's inequality} \tag{62}$$

$$\rightarrow 0 \qquad \text{as } \sigma \rightarrow 0 \tag{63}$$

Case 2. Impulsive noise: Alpha-stable noise.

The characteristic function $\varphi(\omega)$ of alpha-stable density $p(n)$ has the exponential form (35)-(36). This reduces to a simple complex exponential in the zero-dispersion limit:

$$\lim_{\gamma \rightarrow 0} \varphi(\omega) = \exp\{ia\omega\} \tag{64}$$

for *all* α's, skewness β's, and location a's. So Fourier transformation gives the corresponding density function in the limiting case ($\gamma \rightarrow 0$) as a translated delta function

$$\lim_{\gamma \rightarrow 0} p(n) = \delta(n - a) \tag{65}$$

Then

$$P_{Y|S}(0|0) - P_{Y|S}(0|1) = \int_{\theta-A}^{\theta+A} p(n)dn \tag{66}$$

$$= \int_{\theta-A}^{\theta+A} \delta(n - a)dn \tag{67}$$

$$= 0 \tag{68}$$

because $a \notin [\theta - A, \theta + A]$.

Then $P_Y(y) = P_{Y|S}(y|s)$ as $\gamma \rightarrow 0$. So Cases 1 and 2 imply that $I(S, Y) \rightarrow 0$ as $\sigma \rightarrow 0$ for finite-variance noise or as $\gamma \rightarrow 0$ for alpha-stable noise. Q.E.D.

B. Theoretical Results for Closed-Form Noise Densities

We can derive more specific results for closed-form noise densities. Figure 5 shows I-versus-σ profiles of a threshold system with four kinds of noise: Gaussian, uniform, Laplace, and Cauchy. The I profile of the uniform noise has the highest peak among the four noise densities for the same system (same threshold θ and same input amplitude A). And the I profile has a distinct shape: it drops sharply after it reaches its peak as σ grows. Gaussian noise gives the

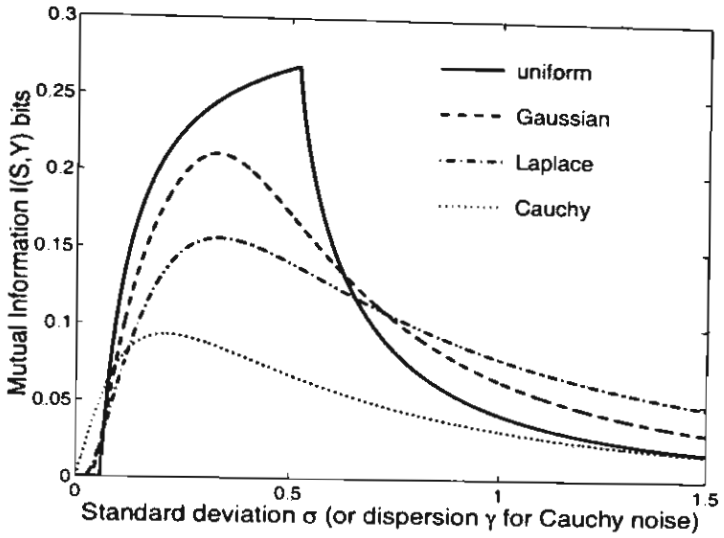

Figure 5. Mutual information I profiles of a threshold system with bipolar input for fours kinds of noise. The system has threshold $\theta = 0.5$. The input Bernoulli signal is bipolar with amplitude $A = 0.4$.

second highest I while Cauchy gives the lowest. The threshold system requires different optimal standard deviations (or dispersions) for different kinds of noise.

The closed form of the I-versus-σ profiles in Figure 5 also allows a direct analysis of how the optimal noise depends on the signal amplitude A for Gaussian, uniform, Laplace, and Cauchy noise. Suppose the signal amplitude A is a subthreshold input in a noisy threshold neuron with threshold θ: $A < \theta$. Then will the optimal noise σ_{opt} (or γ_{opt}) decrease as the signal amplitude A moves closer to the threshold θ?

Intuition suggests that the threshold system should need less noise to produce the entropic SR effect as the amplitude moves closer to the threshold θ. But the results in Figure 6 show that the compound nonlinearities involved produce no such simple relationship. The different noise types produce different SR optimality schedules. Figure 6 shows four optimal noise schedules for the threshold value $\theta = 0.5$. other threshold values produced similar results. Only optimal Laplace and Cauchy noise produce the more intuitive monotone decrease in the optimal noise level with rising signal amplitude A. Optimal uniform noise grows linearly with signal amplitude while optimal Gaussian noise defines a nonmontonic schedule.

IV. Stochastic Resonance in Computer Simulations

Discrete simulations can model continuous-time nonlinear dynamical systems if a stochastic numerical scheme approximates the system dynamics and its signal and noise response. We used a simple stochastic version of the Euler scheme to model a nonlinear system with input forcing signal and noise. We measured how the system performed based on only the system's input-output samples.

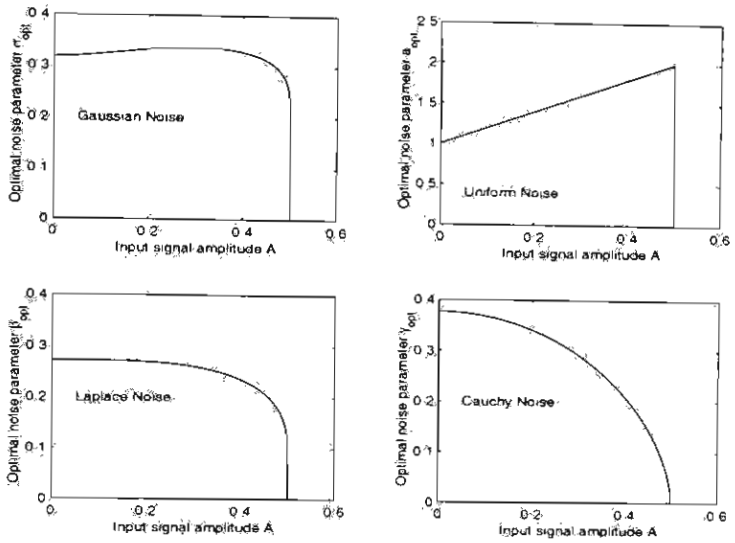

Figure 6. Optimal SR noise schedules for a noisy threshold neuron with threshold $\theta = 0.5$. The schedules show how optimal noise variance or dispersion depends on signal amplitude A for the four closed-form noise results in Figure 5.

Consider the forced dynamical system with additive forcing input signal s and "white" noise n

$$\dot{x} = f(x) + s(t) + n(t) \tag{69}$$

$$y(t) = g(x(t)). \tag{70}$$

These models simply add noise term to a differential equation rather than use a formal Ito or Stratonovich stochastic differentials [14], [23], [29]. By "whiteness" of a random variable n we mean that n is white only over some large but finite frequency bandwidth interval $[-B, B]$ for some large $B > 0$. Random numbers from the algorithms in [62], [13], [66] act as noise from various probability densities in our simulations. The next sections show how we discretized the continuous-time systems to the discrete-time systems to produce computer simulation.

A. *Nonlinear Systems with White Gaussian Noise*

Consider the dynamical system (69) with initial condition $x(t_0) = x_0$. Here the white Gaussian noise w has zero mean and unit variance so that $n = \sigma w$ has zero mean and variance σ^2. This system corresponds to the stochastic initial value problem [29]

$$dX = \tilde{f}(t, X) + \sigma(t, X)dW \tag{71}$$

for initial condition $X(t_0) = X_0$. Here $\tilde{f}(t, X) = f(X) + s(t)$, $\sigma(t, X) = \sigma$ and W is the standard Wiener process [29]. We used Euler's method (the Euler-Maruyama scheme) [20], [29], [40] to

obtain the discrete form for computer simulation:

$$x_{t+1} = x_t + \Delta T \left(f(x_t) + s_t \right) + \sigma \sqrt{\Delta T} w_t \tag{72}$$

$$y_t = g(x_t) \tag{73}$$

for $t = 0, 1, 2, \ldots$ and initial condition x_0. The input sample s_t has the value of the signal $s(t\Delta T)$ at time step t. The zero-mean white Gaussian noise sequence $\{w_t\}$ has unit variance $\sigma_w^2 = 1$. The term $\sqrt{\Delta T}$ scales w_t so that $\sqrt{\Delta T} w_t$ conforms with the Wiener increment [29], [40], [56]. The output sample y_t is some transformation g of the system's state x_t.

This simple algorithm gives fairly accurate results for moderate nonlinear systems [29], [40], [49], [56]. Other algorithms may give more accurate numerical solutions of the stochastic differential equations for more complicated system dynamics [29], [52]. All of our simulations used the Euler's scheme in (72)-(73).

The numerical algorithm in [62] generates a sequence of pseudo-random numbers from a Gaussian density with zero mean and unit variance for $\{w_t\}$ in (72). Figure 3 shows the Gaussian and other densities that have zero mean and a variance of two.

B. Nonlinear Systems with Other Finite-Variance Noise

We next consider a system (69) with finite-variance noise n. Suppose the noise n has variance σ^2 and again apply the above Euler's method:

$$x_{t+1} = x_t + \Delta T \left(f(x_t) + s_t \right) + \sigma \sqrt{\Delta T} w_t \tag{74}$$

$$y_{t+1} = g(x_{t+1}). \tag{75}$$

Here the random sequence $\{w_t\}$ has density function $p(w)$ with zero mean and unit variance. The numerical algorithms in [62] generate sequences of random variables for Laplace and uniform density functions. Figure 3 plots these probability density functions and their realizations with mean zero and variance of two: $E[x] = 0$ and $E[x^2] = 2$.

C. Nonlinear Systems with Alpha-Stable Noise

Figure 3 shows realizations of the symmetric alpha-stable random variable when $\alpha = 1$ (Cauchy density). Again we assume that the Euler's method above applies to this class of random variables with inifinite variance. Let w be a standard alpha-stable random variable with parameter α and zero location and unit dispersion: $a = 0$ and $\gamma = 1$. Let $\kappa = \gamma^{1/\alpha}$ denote a "scale" factor of a random variable. Then $n = \kappa w$ has zero location and dispersion $\gamma = \kappa^\alpha$. This leads us to the Euler's numerical solution

$$x_{t+1} = x_t + \Delta T \left(f(x_t) + s_t \right) + \kappa \sqrt{\Delta T} w_t \tag{76}$$

$$y_t = g(x_t). \tag{77}$$

The algorithm in [13], [66] generates a standard alpha-stable random variable w.

V. Derivation of SR Learning Law

We use a stochastic gradient ascent to learn the SR effect [45], [54]:

$$\sigma_{k+1} = \sigma_k + \mu_k \frac{\partial I}{\partial \sigma} \tag{78}$$

We assume that $P(s)$ does not depend on σ and we use the natural logarithm. Then the learning term $\frac{\partial I}{\partial \sigma}$ has the form

$$\begin{aligned}
\frac{\partial I}{\partial \sigma} &= \frac{\partial}{\partial \sigma}\left(-\sum_y P(y)\log P(y) + \sum_s P(s)\sum_y P(y|s)\log P(y|s)\right) && (79)\\
&= -\sum_y \left(P(y)\frac{1}{P(y)}\frac{\partial P(y)}{\partial \sigma} + \log P(y)\frac{\partial P(y)}{\partial \sigma}\right) \\
&\quad + \sum_s \sum_y \left(P(s)P(y|s)\frac{1}{P(y|s)}\frac{\partial P(y|s)}{\partial \sigma} + P(s)\log P(y|s)\frac{\partial P(y|s)}{\partial \sigma}\right) && (80)\\
&= -\sum_y \left(\frac{\partial P(y)}{\partial \sigma} + \log P(y)\frac{\partial P(y)}{\partial \sigma}\right) \\
&\quad + \sum_s \sum_y \left(P(s)\frac{\partial P(y|s)}{\partial \sigma} + P(s)\log P(y|s)\frac{\partial P(y|s)}{\partial \sigma}\right) && (81)
\end{aligned}$$

The sum $\sum_y P(y) = 1$ implies $\sum_y \frac{\partial P(y)}{\partial \sigma} = \frac{\partial}{\partial \sigma}\sum_y P(y) = 0$. And $\sum_s \sum_y \frac{\partial P(y|s)}{\partial \sigma} = 0$ because $\sum_y P(y|s) = 1$. So

$$\frac{\partial I}{\partial \sigma} = -\sum_y \log P(y)\frac{\partial P(y)}{\partial \sigma} + \sum_s \sum_y P(s)\log P(y|s)\frac{\partial P(y|s)}{\partial \sigma} \tag{82}$$

We estimate the partial derivative with a ratio of time differences and replace the denominator with the signum function to avoid numerical instability:

$$\begin{aligned}
\frac{\partial P(y)}{\partial \sigma} &\approx \frac{P_k(y) - P_{k-1}(y)}{\sigma_k - \sigma_{k-1}} \\
&\approx \operatorname{sgn}(\sigma_k - \sigma_{k-1})[P_k(y) - P_{k-1}(y)] && (83)\\
\frac{\partial P(y|s)}{\partial \sigma} &\approx \frac{P_k(y|s) - P_{k-1}(y|s)}{\sigma_k - \sigma_{k-1}} \\
&\approx \operatorname{sgn}(\sigma_k - \sigma_{k-1})[P_k(y|s) - P_{k-1}(y|s)] && (84)
\end{aligned}$$

where $P_k(y)$ is the marginal density function of the output Y at time t and $P_k(y|s)$ is the conditional density function at time t. Then the learning term becomes

$$\begin{aligned}
\frac{\partial I}{\partial \sigma} \approx \operatorname{sgn}(\sigma_k - \sigma_{k-1}) &\left(-\sum_y [P_k(y) - P_{k-1}(y)]\log P_k(y)\right. \\
&\left. + \sum_s \sum_y P_k(s)[P_k(y|s) - P_{k-1}(y|s)]\log P_k(y|s)\right) && (85)
\end{aligned}$$

Our previous work [45], [54] on adaptive SR found through statistical tests that the random learning term $\frac{\partial P}{\partial a}$ had an approximately Cauchy distribution for the spectral signal-to-noise and cross-correlation performance measures P. These frequent and energetic Cauchy impulse spikes destabilized the stochastic learning process. So we "robustified" the learning term with the standard Cauchy error suppressor $\psi(z_k) = 2z_k/(1 + z_k^2)$ [35], [39]. This included the threshold neuron given a periodic input sequence.

But detailed simulations revealed a special pattern in the case of mutual information: The density $P_k(y)$ tends to stay close to the past density $P_{k-1}(y)$ if the values of a_k and a_{k-1} are close. This causes the learning paths a_k to converge very quickly near the initial conditions. So we can replace the learning term $\frac{\partial I}{\partial a}$ with its sign $\text{sgn}(\frac{\partial I}{\partial a})$ and the learning law simplifies to

$$a_{k+1} = a_k + \gamma_k \text{sgn}(\frac{\partial I}{\partial a}) \tag{86}$$

The signum is a simple robustifier and formally consistent with a two-sided Laplacian distribution [35].

VI. Simulation Results

We tested the robust learning law in (86) with the approximation of the learning term in (85). We needed to estimate the marginal and conditional probability densities $P_k(y)$, $P_k(s)$, and $P_k(y|s)$ at each iteration k. So at each k we collected 1000 input-output samples $\{s_k, y_k\}$ and used them to estimate the densities with histograms for the threshold system. We used 500 of input-output symbols to estimate the pdfs for the continuous neuron model.

A. Noisy Threshold Neuron

The threshold neuron had a fixed threshold $\theta = 0.5$. The bipolar input Bernoulli signal has probability $P_S(-A) = P_S(A) = \frac{1}{2}$ where the amplitude A varied from $A = 0.1$ to $A = 0.4$ (subthreshold inputs). We tried several noise densities that included the Gaussian, uniform, Laplace, and the impulsive Cauchy density. All noise densities had zero mean (zero location for Cauchy) and we tried to learn the optimal standard deviation σ_{opt} (or optimal dispersion γ_{opt} for Cauchy noise). We used constant learning rates $\mu_k = 0.01$ for Gaussian and uniform noise and $\mu_k = 0.02$ for Laplace and Cauchy noise. We use $\mu_k = 0.02$ for alpha-stable noise with $\alpha = 1.9$ and $\alpha = 1.5$. We started the learning from several initial conditions with different noise seeds.

Figures 7-9 show the adapted SR profiles and the σ_{opt} learning paths for different noise types. The learning paths converged to the optimal standard deviation σ_{opt} (or dispersion γ_{opt}) if the initial value was near σ_{opt}. The learning paths tended to stay nearer the optimal values for larger input amplitudes.

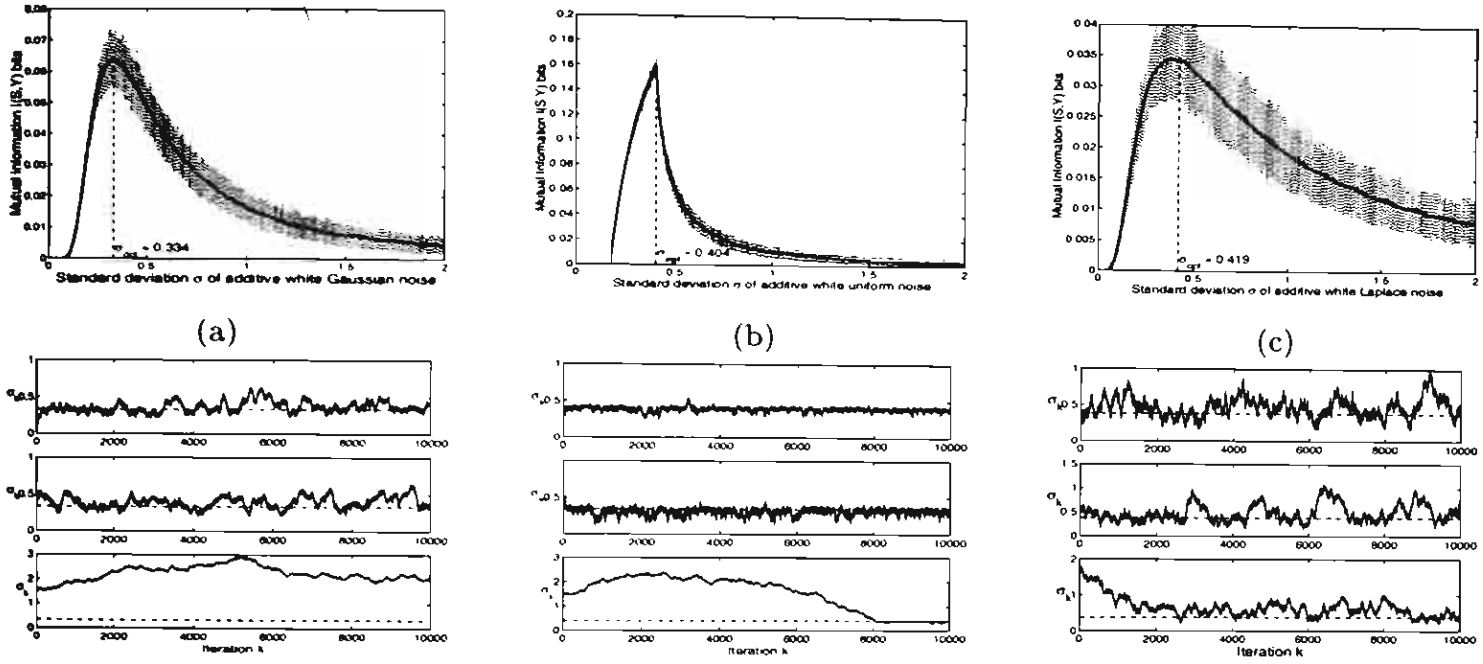

Figure 7. Finite-variance noise cases: Adaptive stochastic resonance for the noisy threshold neuron (6) with bipolar input signal s_t, amplitude $A = 0.2$, and threshold $\theta = 0.5$. The additive noise are (a) Gaussian, (b) uniform, and (c) Laplace. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom plots show the convergence if the initial condition σ_0 is close to the optimal noise level σ_{opt}. Distant initial conditions may lead to divergence as the third learning path in (a) shows. The constant learning rates are $\mu_k = 0.01$ for Gaussian and uniform noise and $\mu_k = 0.02$ for Laplace noise.

B. Noisy Continuous Neuron

We used the discrete model in Section IV for simulations. We used $dt = 0.01$ s and let each input symbol stays for 50 s. So for each input symbol we presented the corresponding "spikes" (plus noise) 5000 times to the neuron. And we collected 5000 discrete time output "spikes" and averaged them to get the output symbol.

The bipolar input Bernoulli signal had success probability $P_S(-A) = P_S(A) = \frac{1}{2}$ where the amplitude A varied from $A = 0.1$ to $A = 0.4$ (subthreshold inputs). We tried several noise densities that included the Gaussian, uniform, Laplace, and Cauchy densities. All noise densities had zero mean (zero location for Cauchy). We used constant learning rates $\mu_k = 0.03$ for Gaussian, uniform, and Laplace noise. We used the smaller learning rates $\mu_k = 0.02$ for alpha-stable noise with $\alpha = 1.9$ and $\alpha = 1.5$. We used the even smaller learning rate $\mu_k = 0.005$ for Cauchy noise. We started the learning from several initial conditions with different noise seeds.

Figures 10-12 show the adapted SR profiles and the σ_{opt} learning paths for different noise types. The learning paths converged near the optimal standard deviation σ_{opt} (or dispersion

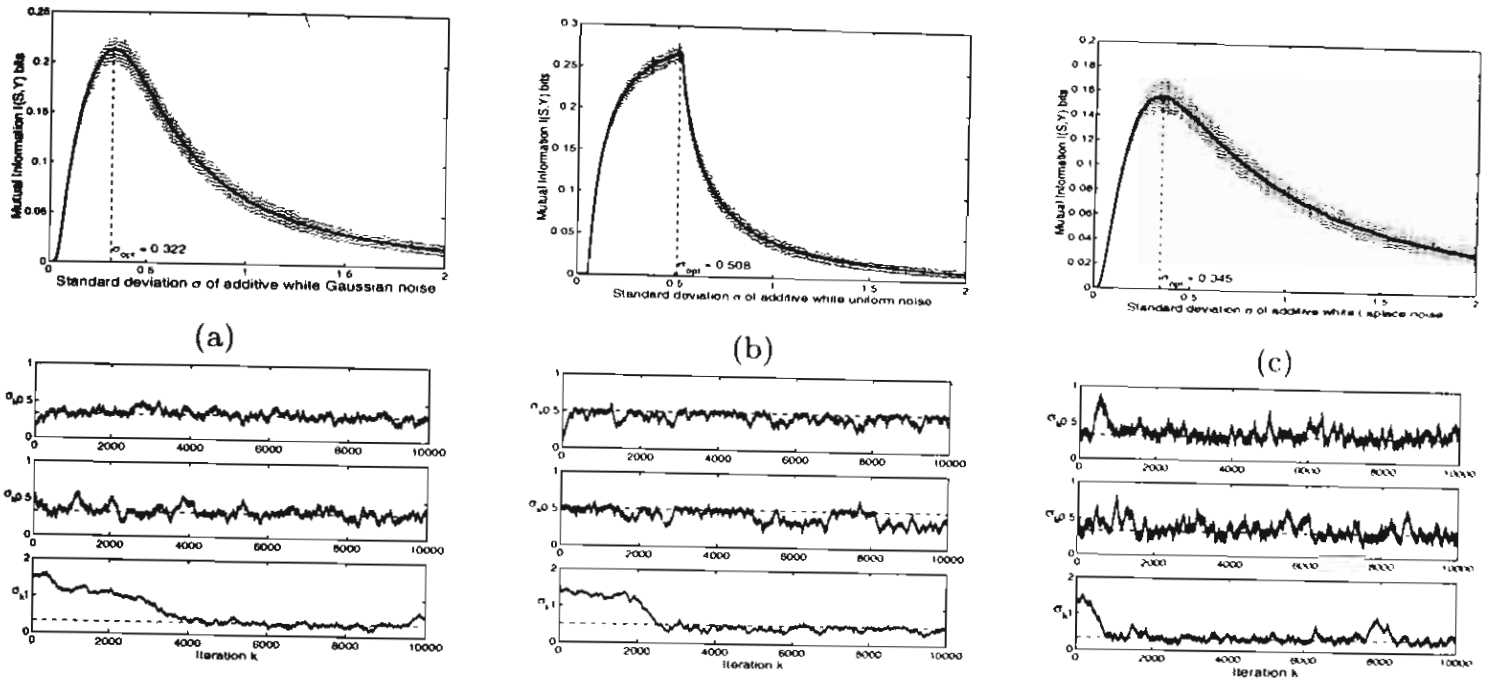

Figure 8. Finite-variance noise cases: Adaptive stochastic resonance for the noisy threshold neuron (6) with bipolar input signal s_t, amplitude $A = 0.4$, and threshold $\theta = 0.5$. The additive noise are (a) Gaussian, (b) uniform, and (c) Laplace. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom plots show the convergence of the noise standard deviation σ_k to the noise optimum σ_{opt} for each noise density. The constant learning rates are $\mu_k = 0.01$ for Gaussian and uniform noise and $\mu_k = 0.02$ for Laplace noise.

γ_{opt}) if the initial value was near σ_{opt}.

VII. Conclusion

Threshold neurons exhibit stochastic resonance–they increase their throughput mutual information when faint input noise increases in intensity. A theorem shows that this holds for almost all noise densities. Such noise-based information maximization is consistent with Linsker's principle of information maximization in neural networks [47], [48]. Closed-form noise densities allow us to derive the exact dependence of mutual information on noise dispersion and to observe the nonlinear relationships between the optimal noise level and the magnitude of the input signal amplitude. Extensive simulations confirmed this entropic SR effect for noisy threshold neurons and for a simple continuous neuron.

A simple robust stochastic learning law can find the entropically optimal noise level for both threshold and continuous neurons that process noisy bipolar input signals. This result holds for many types of finite-variance and infinite-variance (impulsive) noise. These noise types can model energetic disturbances that range from thermal jitter to unmodeled environmental effects to the random crosstalk of neurons in large neural networks. This robust finding supports the

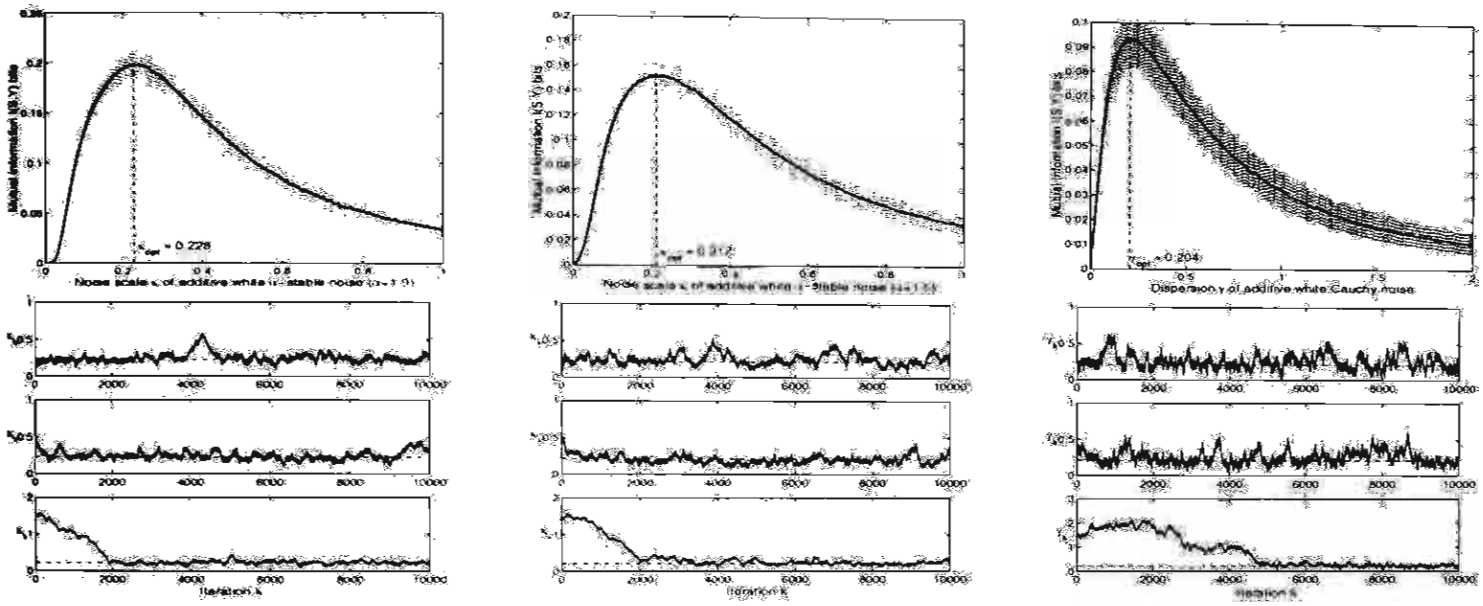

Figure 9. Impulsive noise cases: Adaptive stochastic resonance for the noisy threshold neuron (6) with bipolar input signal s_i, amplitude $A = 0.4$, and threshold $\theta = 0.5$. The additive noise are α-stable distributed with the parameter (a) $\alpha = 1.9$, (b) $\alpha = 1.5$, and (c) $\alpha = 1$ or Cauchy density. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom plots show the convergence of the noise scale κ_k to the noise optimum κ_{opt} for each noise density. The corresponding dispersions are $\gamma = \kappa^\alpha$ for each α-stable noise. The constant learning rates are $\mu_k = 0.01$ for $\alpha = 1.9$ and $\alpha = 1.5$ noise and $\mu_k = 0.02$ for Cauchy noise.

implicit SR conjecture that biological neurons [11], [18], [22], [46], [55], [61], [67] have evolved over genetic eons to exploit the noise energy freely available in their local environment.

VIII. Acknowledgments

National Science Foundation Grant No. ECS-0070284 and Thailand Research Fund Grant No. PDF/29/2543 supported this research.

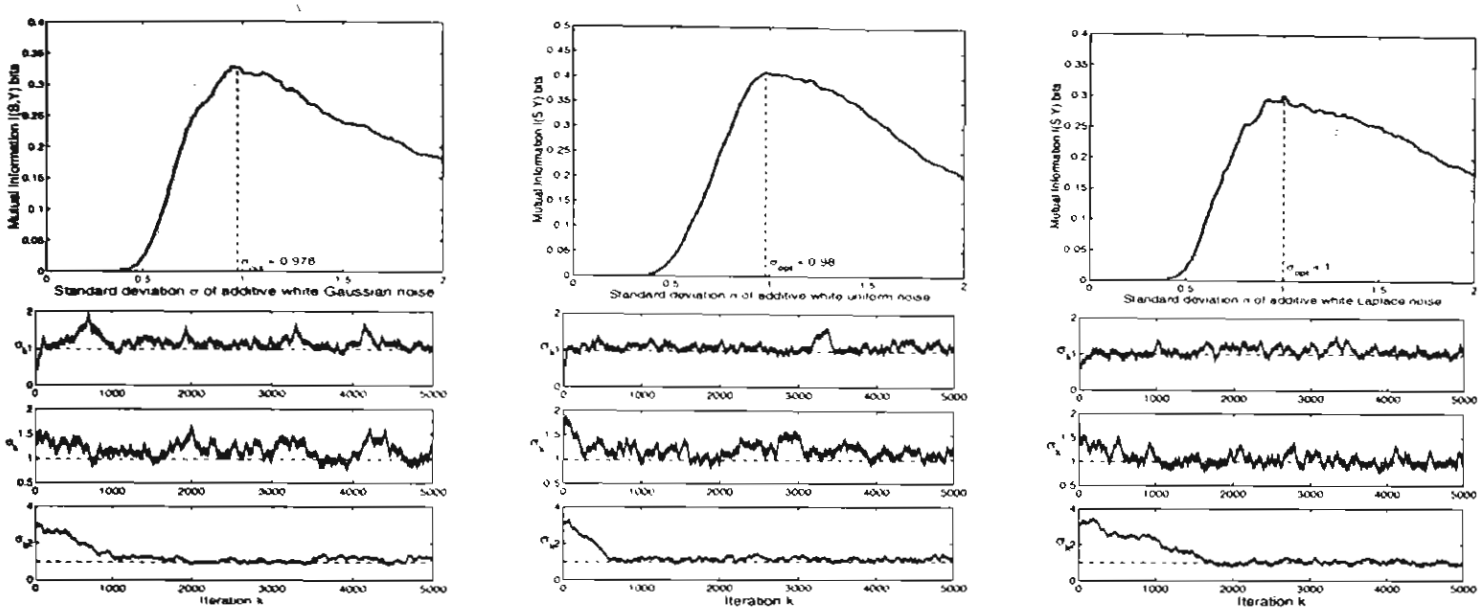

Figure 10. Finite-variance noise cases: Adaptive stochastic resonance for the noisy continuous neuron (7) with bipolar input signal s_t with amplitude $A = 0.2$. The additive noise are (a) Gaussian, (b) uniform, and (c) Laplace. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom plots show the convergence of the noise standard deviation σ_k to the noise optimum σ_{opt} for each noise density. The constant learning rates are $\mu_k = 0.03$ for all cases.

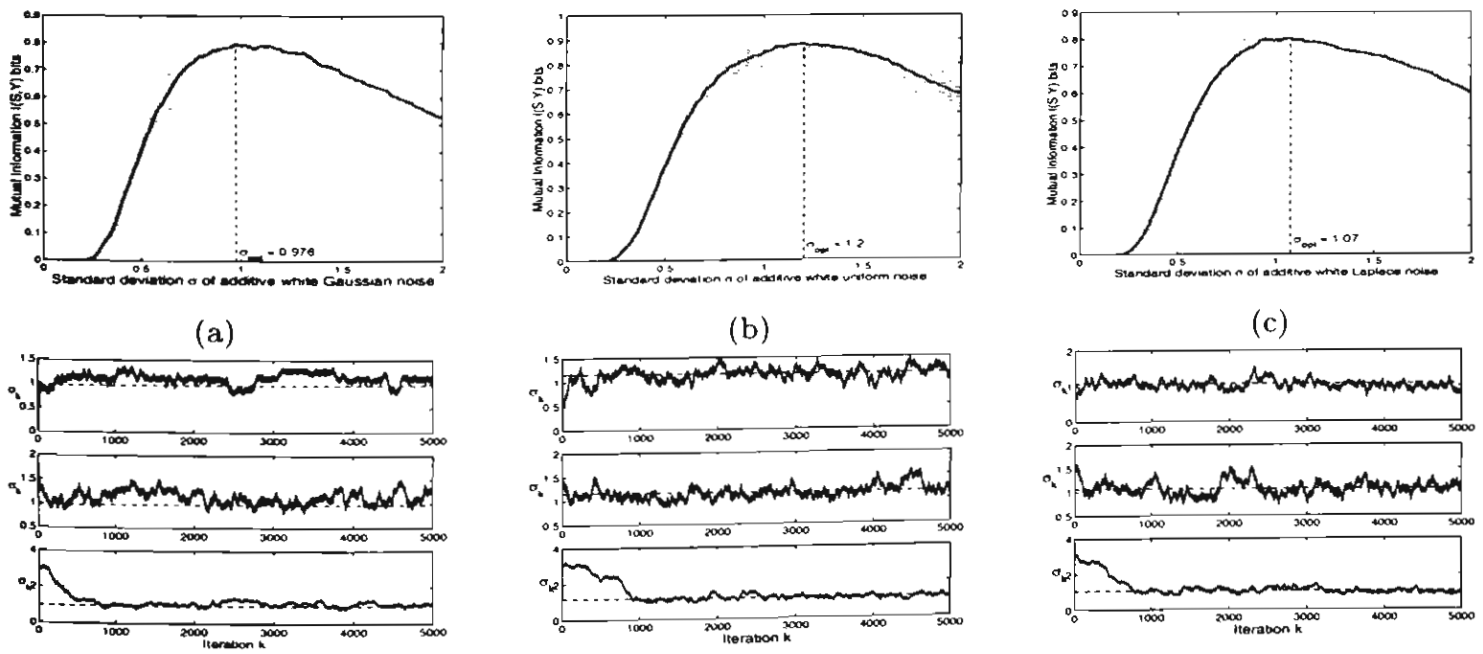

Figure 11. Finite-variance noise cases: Adaptive stochastic resonance for the noisy continuous neuron (7) with bipolar input signal s_t with amplitude $A = 0.4$. The additive noise are (a) Gaussian, (b) uniform, and (c) Laplace. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom plots show the convergence of the noise standard deviation σ_k to the noise optimum σ_{opt} for each noise density. The constant learning rates are $\mu_k = 0.03$ for all cases.

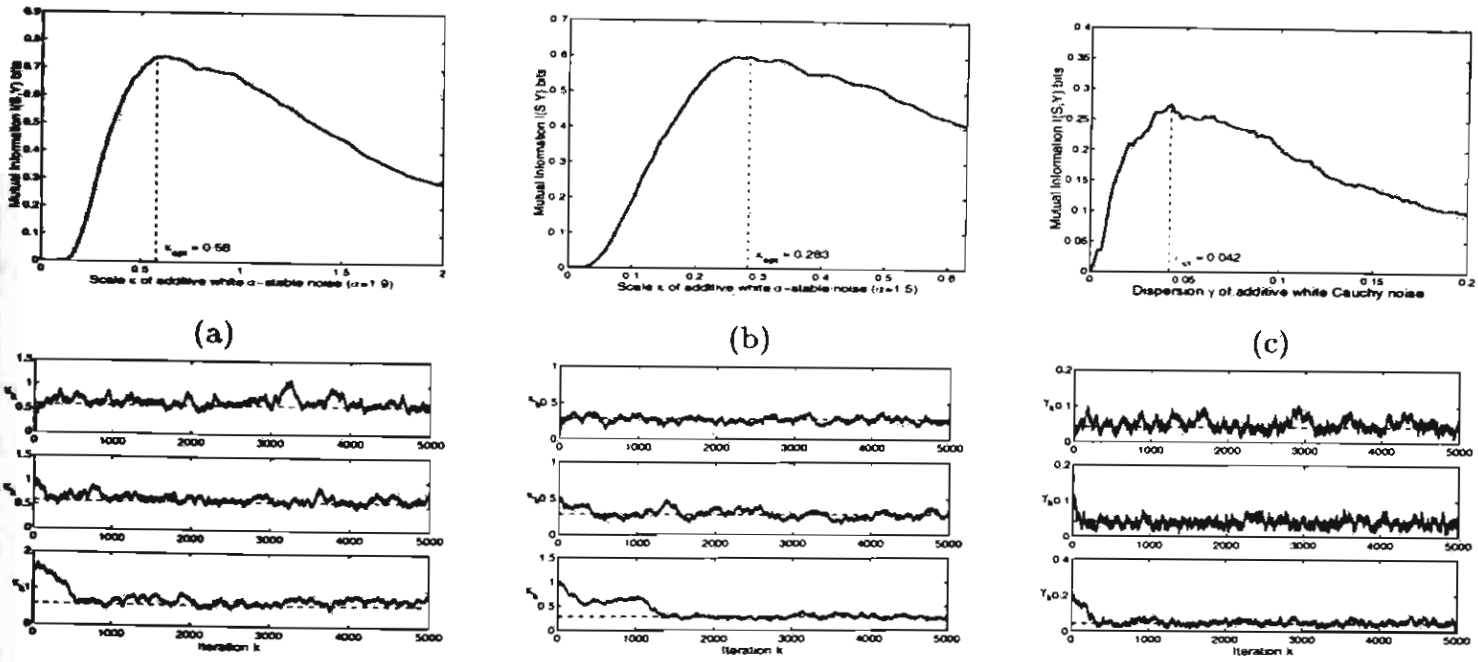

Figure 12. Impulsive noise cases: Adaptive stochastic resonance for the noisy continuous neuron (7) with bipolar input signal s_t with amplitude $A = 0.4$. The additive noise are α-stable distributed with the parameter (a) $\alpha = 1.9$, (b) $\alpha = 1.5$, and (c) $\alpha = 1$ or Cauchy density. The graphs at the top show the nonmonotonic signatures of SR. The sample paths at the bottom show the convergence of the noise standard deviation σ_k to the noise optimum σ_{opt} for each noise density. The constant learning rates are $\mu_k = 0.02$ for $\alpha = 1.9$, $\mu_k = 0.01$ for $\alpha = 1.5$, and $\mu_k = 0.005$ for $\alpha = 1$.

REFERENCES

[1] S. Amari, "Neural Theory of Association and Concept Formation," *Biological Cybernetics*, vol. 26, pp. 175–185, 1977.

[2] V. Akgiray and C. G. Lamoureux, "Estimation of Stable-Law Parameters: A Comparative Study," *Journal of Business and Economic Statistics*, vol. 7, pp. 85–93, January 1989.

[3] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, "Stochastic Resonance in Climatic Change," *Tellus*, vol. 34, pp. 10–16, 1982.

[4] R. Benzi, A. Sutera, and A. Vulpiani, "The Mechanism of Stochastic Resonance," *Journal of Physics A: Mathematical and General*, vol. 14, pp. L453–L457, 1981.

[5] H. Bergstrom, "On Some Expansions of Stable Distribution Functions," *Ark. Math.*, vol. 2, pp. 375–378, 1952.

[6] L. Breiman, *Probability*, Addison-Wesley, 1968.

[7] P. Bryant, K. Wiesenfeld, and B. McNamara, "The Nonlinear Effects of Noise on Parametric Amplification: An Analysis of Noise Rise in Josephson Junctions and Other Systems," *Journal of Applied Physics*, vol. 62, no. 7, pp. 2898-2913, October 1987.

[8] A. R. Bulsara, T. C. Elston, C. R. Doering, S. B. Lowen, and K. Lindenberg, "Cooperative Behavior in Periodically Driven Noisy Integrate-Fire Models of Neuronal Dynamics," *Physical Review E*, vol. 53, no. 4, pp. 3958–3969, April 1996.

[9] A. R. Bulsara and L. Gammaitoni, "Tuning in to Noise," *Physics Today*, vol. 49, no. 3, pp. 39–45, March 1996.

[10] A. R. Bulsara, E. W. Jacobs, T. Zhou, F. Moss, and L. Kiss, "Stochastic Resonance in a Single Neuron Model: Theory and Analog Simulation ," *Journal of Theoretical Biology*, vol. 152, pp. 531–555, 1991.

[11] A. R. Bulsara, A. J. Maren, and G. Schmera, "Single Effective Neuron: Dendritic Coupling Effects and Stochastic Resonance," *Biological Cybernetics*, vol. 70, pp. 145–156, 1993.

[12] A. R. Bulsara and A. Zador, "Threshold Detection of Wideband Signals: A Noise-Induced Maximum in the Mutual Information," *Physical Review E*, vol. 54, no. 3, pp. R2185–R2188, September 1996.

[13] J. M. Chambers, C. L. Mallows, and B. W. Stuck, "A Method for Simulating Stable Random Variables," *Journal of the American Statistical Association*, vol. 71, no. 354, pp. 340–344, 1976.

[14] K. L. Chung and R. J. Williams, *Introduction to Stochastic Integration*, Birkhäuser, second edition, 1990.

[15] M. A. Cohen and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, pp. 815–826, September 1983.

[16] J. J. Collins, C. C. Chow, A. C. Capela, and T. T. Imhoff, "Aperiodic Stochastic Resonance," *Physical Review E*, vol. 54, no. 5, pp. 5575–5584, November 1996.

[17] J. J. Collins, C. C. Chow, and T. T. Imhoff, "Stochastic Resonance without Tuning," *Nature*, vol. 376, pp. 236–238, July 1995.

[18] J. J. Collins, T. T. Imhoff, and P. Grigg, "Noise-Enhanced Information Transmission in Rat SA1 Cutaneous Mechanoreceptors via Aperiodic Stochastic Resonance," *Journal of Neurophysiology*, vol. 76, no. 1, pp. 642–645, July 1996.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[20] G. Dahlquist and Å. Björck, *Numerical Methods*, Prentice Hall, 1974.

[21] G. Deco and B. Schürmann, "Stochastic Resonance in the Mutual Information between Input and Output Spike Trains of Noisy Central Neurons," *Physica D*, vol. 117, pp. 276–282, 1998.

[22] J. K. Douglass, L. Wilkens, E. Pantazelou, and F. Moss, "Noise Enhancement of Information Transfer in Crayfish Mechanoreceptors by Stochastic Resonance," *Nature*, vol. 365, pp. 337–340, September 1993.

[23] R. Durrett, *Stochastic Calculus: A Practical Introduction*, CRC Press, 1996.

[24] M. I. Dykman, T. Horita, and J. Ross, "Statistical Distribution and Stochastic Resonance in a Periodically Driven Chemical System," *Journal of Chemical Physics*, vol. 103, no. 3, pp. 966–972, July 1995.

[25] J.-P. Eckmann and L. E. Thomas, "Remarks on Stochastic Resonances," *Journal of Physics A: Mathematical and General*, vol. 15, pp. L261–L266, 1982.

[26] S. Fauve and F. Heslot, "Stochastic Resonance in a Bistable System," *Physics Letters A*, vol. 97, no. 1,2, pp. 5–7, August 1983.

[27] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. II, John Wiley & Sons, 1966.

[28] L. Gammaitoni, "Stochastic Resonance in Multi-Threshold Systems," *Physics Letters A*, vol. 208, pp. 315–322, December 1995.

[29] T. C. Gard, *Introduction to Stochastic Differential Equations*, Marcel Dekker, Inc., 1988.

[30] J. Glanz, "Mastering the Nonlinear Brain," *Science*, vol. 277, pp. 1758–1760, September 1997.

[31] X. Godivier and F. Chapeau-Blondeau, "Stochastic Resonance in the Information Capacity of a Nonlinear Dynamic System," *International Journal of Bifurcation and Chaos*, vol. 8, no. 3, pp. 581–589, 1998.

[32] M. Grifoni, M. Sassetti, P. Hänggi, and U. Weiss, "Cooperative Effects in the Nonlinearly Driven Spin-Boson System," *Physical Review E*, vol. 52, no. 4, pp. 3596–3607, October 1995.

[33] M. Grigoriu, *Applied Non-Gaussian Processes*, Prentice Hall, 1995.

[34] A. Hibbs, E. W. Jacobs, J. Bekkedahl, A. R. Bulsara, and F. Moss, "Signal Enhancement in a r.f. SQUID Using Stochastic Resonance," *Il Nuovo Cimento*, vol. 17 D, no. 7-8, pp. 811–817, Luglio-Agosto 1995.

[35] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Prentice Hall, fifth edition, 1995.

[36] N. Hohn and A. N. Burkitt, "Enhanced Stochastic Resonance in Threhold Detectors," in *Proceedings of the 2001 IEEE International Joint Conference on Neural Networks*, 2001, vol. 1, pp. 644–647.

[37] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science*, vol. 79, pp. 2554–2558, 1982.

[38] J. J. Hopfield, "Neural Networks with Graded Resonse Have Collective Computational Properties Like Those of Two-State Neurons," *Proceedings of the National Academy of Science*, vol. 81, pp. 3088-3092, 1984.

[39] P. J. Huber, *Robust Statistics*, John Wiley & Sons, 1981.

[40] M. E. Inchiosa and A. R. Bulsara, "Nonlinear Dynamic Elements with Noisy Sinusoidal Forcing: Enhancing Response via Nonlinear Coupling," *Physical Review E*, vol. 52, no. 1, pp. 327–339, July 1995.

[41] M. E. Inchiosa, J. W. C. Robinson, and A. R. Bulsara, "Information-Theoretic Stochastic Resonance in Noise-Floor Limited Systems: The Case for Adding Noise," *Physical Review Letters*, vol. 85, no. 6, pp. 3369–3372, October 2000.

[42] P. Jung, "Stochastic Resonance and Optimal Design of Threshold Detectors," *Physics Letters A*, vol. 207, pp. 93–104, October 1995.

[43] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 1991.

[44] B. Kosko, *Fuzzy Engineering*, Prentice Hall, 1996.

[45] B. Kosko and S. Mitaim, "Robust Stochastic Resonance: Signal Detection and Adaptation in Impulsive Noise," *Physical Review E*, vol. 64, no. 051110, October 2001.

[46] J. E. Levin and J. P. Miller, "Broadband Neural Encoding in the Cricket Cercal Sensory System Enhanced by Stochastic Resonance," *Nature*, vol. 380, pp. 165–168, March 1996.

[47] R. Linsker, "Self-Organization in a Perceptual Network," *Computer*, vol. 21, no. 3, pp. 105–117, March 1988.

[48] R. Linsker, "A Local Learning Rule That Enables Information Maximization for Arbitrary Input Distributions," *Neural Computation*, vol. 9, no. 8, pp. 1661–1665, November 1997.

[49] A. Longtin, "Autonomous Stochastic Resonance in Bursting Neurons," *Physical Review E*, vol. 55, no. 1, pp. 868–876, January 1997.

[50] J. Maddox, "Towards the Brain-Computer's Code?," *Nature*, vol. 352, pp. 469, August 1991.

[51] J. Maddox, "Bringing More Order out of Noisiness," *Nature*, vol. 369, pp. 271, May 1994.

[52] R. Mannella and V. Palleschi, "Fast and Precise Algorithm for Computer Simulation of Stochastic Differential Equations," *Physical Review A*, vol. 40, no. 6, pp. 3381-3386, September 1989.

[53] B. McNamara, K. Wiesenfeld, and R. Roy, "Observation of Stochastic Resonance in a Ring Laser," *Physical Review Letters*, vol. 60, no. 25, pp. 2626–2629, June 1988.

[54] S. Mitaim and B. Kosko, "Adaptive Stochastic Resonance," *Proceedings of the IEEE: Special Issue on Intelligent Signal Processing*, vol. 86, no. 11, pp. 2152–2183, November 1998.

[55] R. P. Morse and E. F. Evans, "Enhancement of Vowel Coding for Cochlear Implants by Addition of Noise," *Nature Medicine*, vol. 2, no. 8, pp. 928 932, August 1996.

[56] F. Moss and P. V. E. McClintock, Eds., *Noise in Nonlinear Dynamical Systems*, vol. I-III, Cambridge University Press, 1989.

[57] D. C. Munson, "A Note on Lena," *IEEE Transactions on Image Processing*, vol. 5, no. 1, pp. 3, January 1996.

[58] C. Nicolis, "Stochastic Aspects of Climatic Transitions-Response to a Periodic Forcing," *Tellus*, vol. 34, pp. 1–9, 1982.

[59] C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications*, John Wiley & Sons, 1995.

[60] X. Pei, K. Bachmann, and F. Moss, "The Detection Threshold, Noise and Stochastic Resonance in the Fitzhugh-Nagumo Neuron Model," *Physics Letters A*, vol. 206, pp. 61–65, October 1995.

[61] X. Pei, L. Wilkens, and F. Moss, "Light Enhances Hydrodynamic Signaling in the Multimodal Caudal Photoreceptor Interneurons of the Crayfish," *Journal of Neurophysiology*, vol. 76, no. 5, pp. 3002–3011, November 1996.

[62] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, second edition, 1993.

[63] D. F. Russell, L. A. Willkens, and F. Moss, "Use of Behavioural Stochastic Resonance by Paddle Fish for Feeding," *Nature*, vol. 402, pp. 291–294, November 1999.

[64] M. Shao and C. L. Nikias, "Signal Processing with Fractional Lower Order Moments: Stable Processes and Their Applications," *Proceedings of the IEEE*, vol. 81, pp. 984–1010, July 1993.

[65] N. G. Stocks, "Information Transmission in Parallel Threshold Arrays: Suprathreshold Stochastic Resonance," *Physical Review E*, vol. 63, no. 041114, 2001.

[66] P. Tsakalides and C. L. Nikias, "The Robust Covariation-Based MUSIC (ROC-MUSIC) Algorithm for Bearing Estimation in Impulsive Noise Environments," *IEEE Transactions on Signal Processing*, vol. 44, no. 7, pp. 1623–1633, July 1996.

[67] M. Usher and M. Feingold, "Stochastic Resonance in the Speed of Memory Retrieval," *Biological Cybernetics*, vol. 83, pp. L11–L16, 2000.

[68] K. Wiesenfeld and F. Moss, "Stochastic Resonance and the Benefits of Noise: From Ice Ages to Crayfish and SQUIDs," *Nature*, vol. 373, pp. 33–36, January 1995.

Robust stochastic resonance: Signal detection and adaptation in impulsive noise

Bart Kosko[1] and Sanya Mitaim[2]
[1]*Department of Electrical Engineering, Signal and Image Processing Institute, University of Southern California, Los Angeles, California 90089-2564*
[2]*Department of Electrical Engineering, Faculty of Engineering, Thammasat University Rangsit Campus, Klong Luang, Pathumthani 12121, Thailand*

(Received 23 October 2000; revised manuscript received 7 May 2001; published 22 October 2001)

Stochastic resonance (SR) occurs when noise improves a system performance measure such as a spectral signal-to-noise ratio or a cross-correlation measure. All SR studies have assumed that the forcing noise has finite variance. Most have further assumed that the noise is Gaussian. We show that SR still occurs for the more general case of impulsive or infinite-variance noise. The SR effect fades as the noise grows more impulsive. We study this fading effect on the family of symmetric α-stable bell curves that includes the Gaussian bell curve as a special case. These bell curves have thicker tails as the parameter α falls from 2 (the Gaussian case) to 1 (the Cauchy case) to even lower values. Thicker tails create more frequent and more violent noise impulses. The main feedback and feedforward models in the SR literature show this fading SR effect for periodic forcing signals when we plot either the signal-to-noise ratio or a signal correlation measure against the dispersion of the α-stable noise. Linear regression shows that an exponential law $\gamma_{opt}(\alpha) = cA^{\alpha}$ describes this relation between the impulsive index α and the SR-optimal noise dispersion γ_{opt}. The results show that SR is robust against noise "outliers." So SR may be more widespread in nature than previously believed. Such robustness also favors the use of SR in engineering systems. We further show that an adaptive system can learn the optimal noise dispersion for two standard SR models (the quartic bistable model and the FitzHugh-Nagumo neuron model) for the signal-to-noise ratio performance measure. This also favors practical applications of SR and suggests that evolution may have tuned the noise-sensitive parameters of biological systems.

DOI: 10.1103/PhysRevE.64.051110 PACS number(s): 05.40.−a

I. IMPULSIVE NOISE AND STOCHASTIC RESONANCE

Most noise processes have infinite variance. This mathematical fact is almost trivial. Even most bell-curve probability densities do not have finite variance or any finite higher-order moments. Yet this fact finds scant expression in over a century of published research in science and engineering. A review of the published statistical research in any field shows a common practice. Most random models assume that the dispersion of a random variable equals its squared-error measure of variance. But other measures of dispersion may be finite while the variance measure is infinite. The popularity of the finite-variance assumption may attest to its usefulness in many cases. But that does not lessen its severity. The assumption persists even though such a squared-error term seldom exists in any formal generality and even though such a squared-error term is not robust against data "outliers" when it does exist. Celebrated examples of the finite-variance hypothesis range from the Heisenberg uncertainty principle in quantum mechanics to the least-squares regression framework that underlies statistical curve fitting and forecasting in fields as disparate as astronomy and sociology.

The presence of infinite variance in a random model does not itself nullify the model or count as some sort of stochastic *reductio ad absurdum*. Infinite variance does not imply that we lack all statistical knowledge about the position or momentum of a random particle or about the value of any random variable if we assume only that the random variable has a probability density function in the shape of a bell curve. Many infinite-variance bell curves are locally indistinguishable from the thinner-tailed Gaussian bell curve. Infinite-variance noise itself produces impulses of only finite magnitude. Nor does infinite variance imply that a real system must have infinite energy. This holds for the same reason that the use of a Gaussian bell curve in a model does not imply that axes extend to infinity in the real world. Other events can explain the presence of infinite variance. We may have measured the random dispersion involved with the wrong measure. We may have applied a good but approximate measure to extreme cases that lie outside the measure's particular structure. Or we may simply have used or encountered a bell curve that has thicker tails than a Gaussian bell curve has.

Stochastic resonance [1–13] offers a recent and stark example of the finite-variance assumption. A dynamical system stochastically resonates or shows the stochastic resonance (SR) effect when noise increases its signal-to-noise ratio or other system performance measure. Almost all SR research has assumed that the noise process is Gaussian and hence has finite variance. A few SR studies have explored uniform and other non-Gaussian but finite-variance noise-types [14–17]. The SR signature of a nonmonotonic signal-to-noise graph gives perhaps the best evidence of the universality of the finite-variance assumption in SR research. All SR studies plot the dynamical system's signal-to-noise ratio against either the variance or the standard deviation of the driving noise process. So the very notation excludes the presence of infinite variance. This practice rules out a vast set of possible SR scenarios and suggests that SR is not robust against noise outliers. The simulation results below show that the SR effect can indeed occur when infinite-variance noise drives nonlinear feedback and feedforward systems.

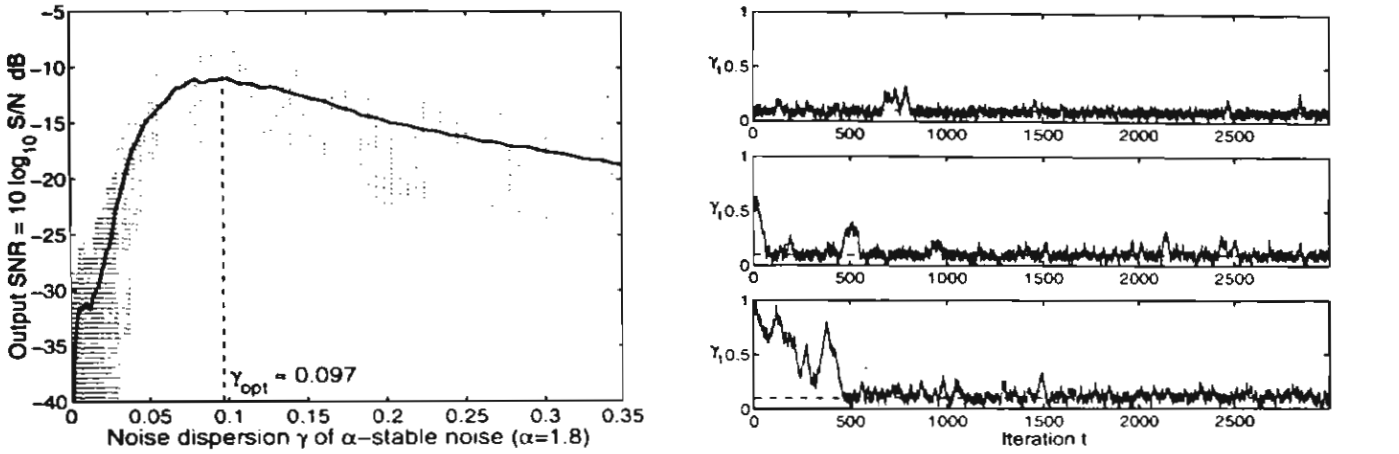

FIG. 1. Impulsive stochastic resonance SNR measure and convergence of learned dispersion to the SR effect. (a) The graph shows the smoothed output SNR as a function of the dispersion of additive α-stable (infinite-variance) noise with $\alpha=1.8$. The vertical dotted lines show the absolute deviation between the smallest and largest SNR outliers in each sample average of 100 outcomes. The vertical dashed line shows the optimal noise level at the dispersion $\gamma\approx 0.097$. The noisy signal-forced quartic bistable dynamical system has the form $\dot{x}=x-x^3+s+n$ with binary output $y(t)=\mathrm{sgn}(x(t))$. The α-stable noise $n(t)$ (with $\alpha=1.8$) adds to the external forcing narrow band signal $s(t)=0.1\sin 2\pi(0.01)t$. (b) Learning paths of γ_t with the Cauchy impulse suppressor $\phi(z)=2z/(1+z^2)$ for the quartic bistable system with sinusoidal input. The Cauchy impulse suppressor $\phi(\partial \mathrm{SNR}_t/\partial\sigma)$ replaces $\partial \mathrm{SNR}_t/\partial\sigma$ in the SR learning law [16] as in Eq. (43) below. The learning paths converged to and wander about the optimal noise dispersion $\gamma_{opt}\approx 0.097$.

Stochastic resonance occurs in a signal-forced dynamical system when noise improves its performance by increasing its signal-to-noise ratio (SNR) [18–22] or some other performance measure such as a signal cross correlation [23–27] or mutual entropy [25–27]. Then the noise process $n(t)$ and signal process $s(t)$ force a feedback dynamical system of the form $\dot{x}=f(x)$ to give $\dot{x}=f(x)+s(t)+n(t)$. The forced system's signal-to-noise ratio has the form $\mathrm{SNR}=S/N$ where S measures the spectral content of the forcing signal $s(t)$ in the forced system and N measures the spectral content of the noise $n(t)$ [as entangled with each other and with the system state dynamics $\dot{x}=f(x)$]. Most SR systems in the literature have assumed that the forcing signal has the simple periodic form of a sinusoid. Aperiodic SR [23,24] is an important exception that we do not consider here.

The figures show the main results of this research. Figure 1 shows an SR profile when the additive forcing impulsive noise has infinite variance. The noise has alpha value $\alpha=1.8$ and so the noise is only mildly impulsive compared to the noise that arises from bell curves with thicker tails. Figure 1 also shows the more complex result that a stochastic learning algorithm can learn to locate the SR-optimal dispersion value in this impulsive environment and do so based not on the functional form of the dynamical stable (the quartic bistable system in this case) but based on only input-output training samples of dispersion and SNR values. Each SNR value depends on the noise-corrupted system dynamics. This allows the learning process to in effect slowly estimate the system dynamics. The presence of system dynamics means that the same dispersion value or the same noise impulse will at different times produce different SNR values. Learning based on a correlation measure requires direct use of the state dynamics.

Figure 2 shows four α-stable bell curves and the noise samples they produce [28,29]. It also shows three infinite-variance curves for $\alpha=1.8$ based on three dispersion values and the resulting samples of impulsive noise. The three impulsive SR profiles for the SNR measure in Fig. 3 show that the SR mode occurs for even smaller dispersion values as the impulsiveness grows (as α falls). Figure 4 shows that the pattern in Fig. 3 generalizes. Impulsiveness decreases stochastic resonance because the exponential law $\gamma_{opt}(\alpha)=cA^{\alpha}$ tends to hold for all the dynamical systems we studied. Figure 5 confirms this pattern for the cross-correlation performance measure for a quartic bistable system. Figure 6 shows that any SNR-based learning scheme faces Cauchy-like impulsiveness as it approaches the first-order condition for an SR optimum. This impulsiveness occurs for all noise-types including the Gaussian. This in turn implies that both biological and engineering systems must find some way to suppress this second level of impulsiveness if they try to learn the SR optimum or otherwise search for it based on noisy training data.

II. SYMMETRIC α-STABLE NOISE: THICK-TAILED BELL CURVES

We use a class of symmetric α-stable bell-curve probability density functions with parameter α in the characteristic function $\phi(\omega)=\exp[-\gamma|\omega|^{\alpha}]$ where γ is the *dispersion* parameter [30–33]. The parameter α lies in $0<\alpha\leq 2$ and gives the Gaussian random variable when $\alpha=2$ or when $\varphi(\omega)=\exp\{-\gamma\omega^2\}$. So the standard Gaussian random variable has zero mean and variance $\sigma^2=2$ (when $\gamma=1$). The parameter α gives the thicker-tailed Cauchy bell curve when $\alpha=1$ or $\varphi(\omega)=\exp\{-|\omega|\}$ for a zero *location* ($a=0$) and unit disper-

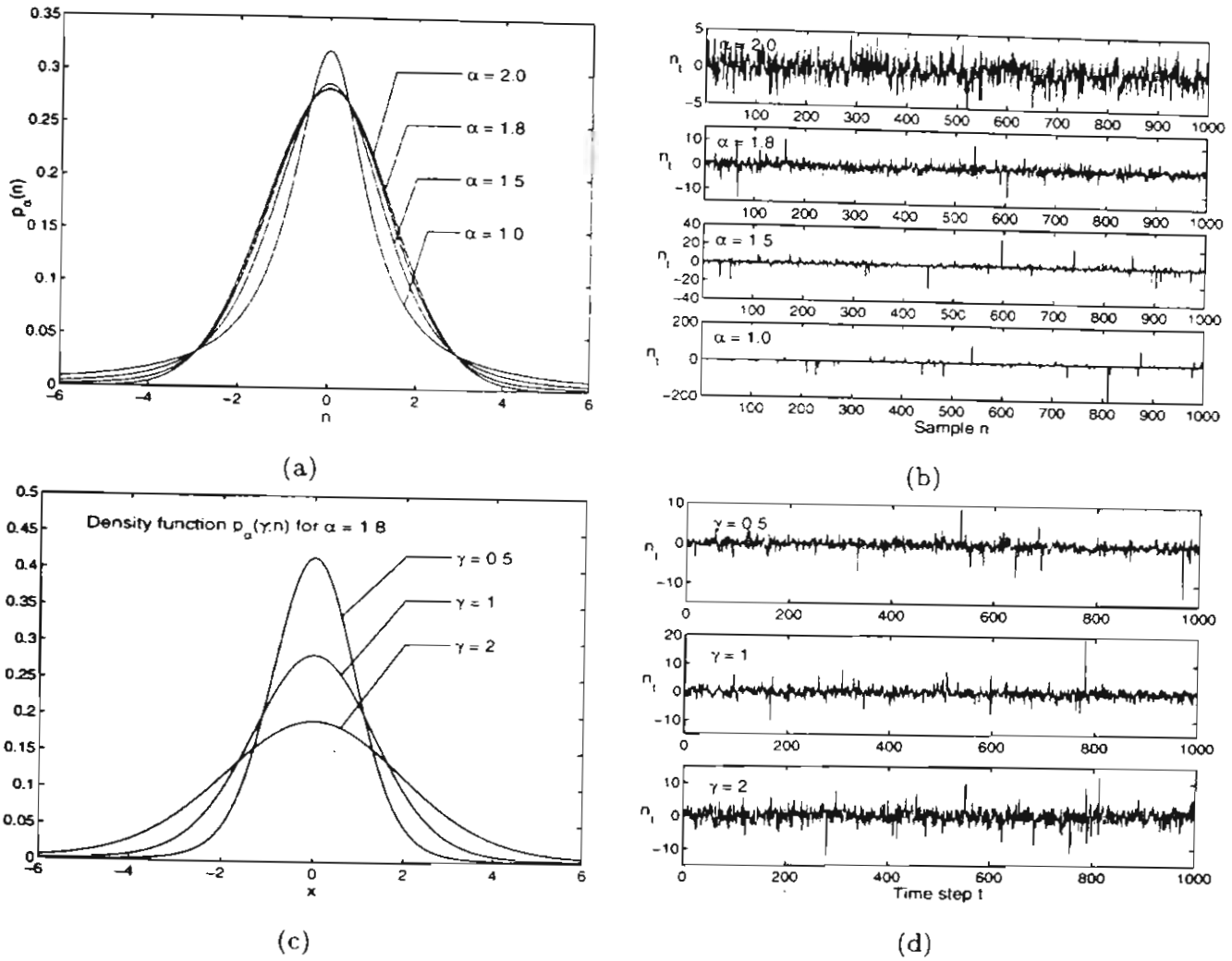

FIG. 2. Samples of standard symmetric α-stable probability densities and their realizations. (a) Density functions with zero location ($a=0$) and unit dispersion ($\gamma=1$) for $\alpha=2$, 1.8, 1.5, and 1. The densities are bell curves that have thicker tails as α decreases. The case $\alpha=2$ gives a Gaussian density with variance two (or unit dispersion). The parameter $\alpha=1$ gives the Cauchy density. (b) Samples of α-stable random variables with zero location and unit dispersion. The plots show realizations when $\alpha=2$, 1.8, 1.5, and 1. Note the scale differences on the y axes. The α-stable variable x becomes more impulsive as the parameter α falls. The algorithm in [28,29] generates these realizations. (c) Density function for $\alpha=1.8$ with dispersion $\gamma=0.5$, 1, and 2. (d) Samples of α-stable noise n for $\alpha=1.8$ with dispersions $\gamma=0.5$, 1, and 2.

sion ($\gamma=1$) Cauchy random variable. The moments of stable distributions with $\alpha<2$ are finite only up to the order k for $k<\alpha$. The Gaussian density alone has finite variance and higher moments. α-stable random variables characterize the class of normalized sums of independent random variables that converge in distribution to a random variable [30] as in the famous Gaussian special case called the "central limit theorem." α-stable models tend to work well when the noise or signal data contains "outliers"—and all do to some degree. Models with $\alpha<2$ can accurately describe impulsive noise in telephone lines, underwater acoustics, low-frequency atmospheric signals, fluctuations in gravitational fields and financial prices, and many other processes [33,34]. The best choice of α is always an *empirical* question for bell-curve phenomena.

Figure 2 shows realizations of four symmetric α-stable random variables. An α-stable probability density f has the characteristic function [32,33,35,36] φ:

$$\varphi(\omega)=\exp\left[ia\omega-\gamma|\omega|^{\alpha}\left\{1+i\beta\,\mathrm{sgn}(\omega)\tan\frac{\alpha\pi}{2}\right\}\right] \quad \text{for } \alpha\neq 1 \quad (1)$$

and

$$\varphi(\omega)=\exp[ia\omega-\gamma|\omega|(1+2i\beta\ln|\omega|\mathrm{sgn}(\omega)/\pi)] \quad \text{for } \alpha=1 \quad (2)$$

where

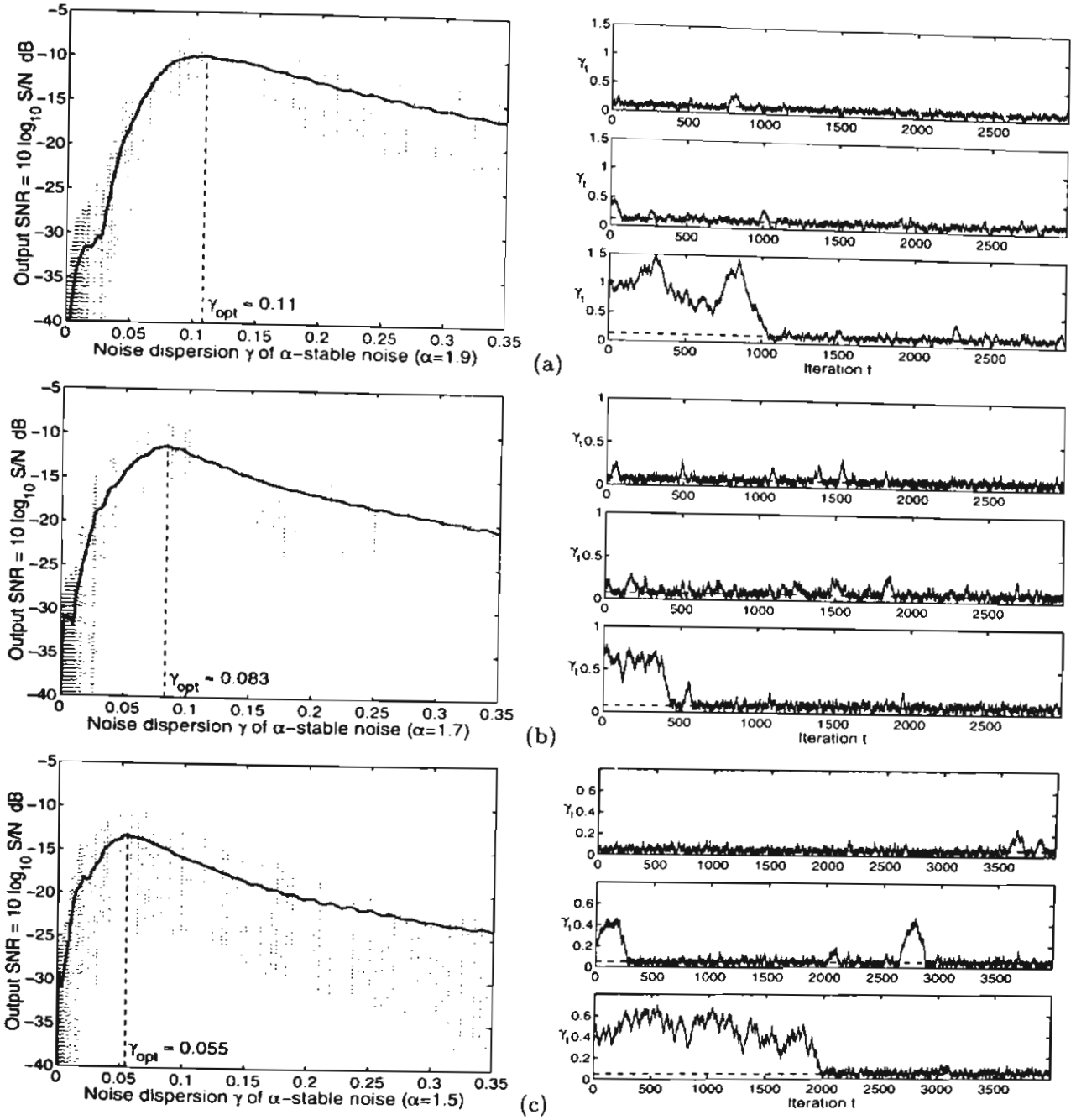

FIG. 3. The optimal dispersion γ_{opt} becomes smaller as the tails on the noise bell curves become thicker and thus as the infinite-variance noise becomes more impulsive. The three SR profiles show that γ_{opt} shifts to the left as α falls. Figure 4 shows that this trend generalizes to an exponential relationship between α and γ_{opt}. The dynamical system is the quartic bistable $\dot{x}=x-x^3+s+n$ modified for saturations effects and where the signal s is the sinusoid $s(t)=0.1\sin 2\pi(0.01)t$. The plots on the left side show the SNR-dispersion profiles for (a) $\alpha=1.9$, (b) $\alpha=1.7$, and (c) $\alpha=1.5$. The dotted lines show the absolute deviation between the smallest and largest SNR outliers in each sample average of 100 outcomes. The vertical dashed lines show the SR effect or mode at the optimal noise dispersion γ_{opt}. The plots on the right side of (a)–(c) show the learning paths of γ as it slowly and noisily converges to γ_{opt} per the robustified learning law in Eq. (43).

$$\mathrm{sgn}(\omega)=\begin{cases}1 & \text{if } \omega>0\\ 0 & \text{if } \omega=0\\ -1 & \text{if } \omega<0,\end{cases} \tag{3}$$

and $i=\sqrt{-1}$, $0<\alpha\leq 2$, $-1\leq\beta\leq 1$, and $\gamma>0$. The α is the characteristic exponent parameter. An α-stable density with $\alpha<2$ has finite moments only of order less than α. Again the variance of an α-stable density distribution does not exist if $\alpha<2$. The location parameter a is the "mean" of the density when $\alpha>1$ and β is a skewness parameter. The density is symmetric about a when $\beta=0$. The dispersion parameter γ acts like a variance because it controls the width of a symmetric α-stable bell curve.

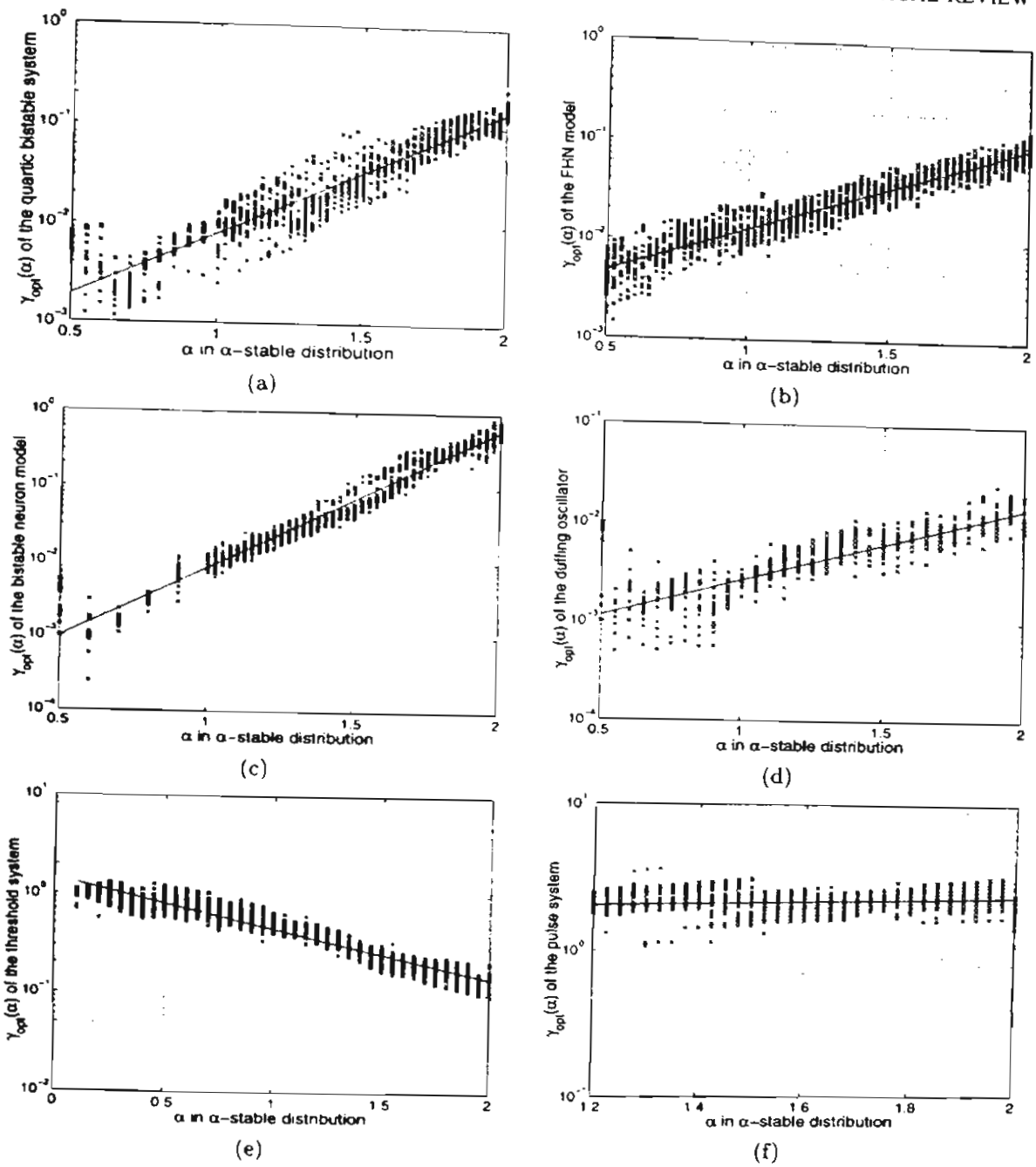

FIG. 4. Exponential laws for optimal noise dispersion γ and parameter α for the SNR performance measure. The optimal noise dispersion γ depends on the parameter α through the exponential relation $\gamma_{opt}(\alpha) = cA^{\alpha}$ for some constants c and A. Table I shows the constants c and A for the dynamical systems we tested. (a) the Quartic bistable system (modified), (b) the FHN model (modified), (c) the bistable neuron model (Hopfield), (d) the duffing oscillator, (e) the feedforward threshold system, and (f) the random pulse system. The slope of the pulse-system in (f) is so close to zero as to undermine the log-linear (exponential) relationship. The small correlation coefficient for the pulse system in Table I reflects this nearly flat log-linear relationship.

III. AN EXPONENTIAL LEARNING LAW: IMPULSIVENESS DECREASES RESONANCE

This section lists the SR performance measures and state models that we used in the simulations. Four of the six state models are feedback or dynamical systems. The neuron and pulse models are feedforward models. All give rise to the exponential law $\gamma_{opt}(\alpha) = cA^{\alpha}$ but the pulse model does so with only a small correlation coefficient of linear regression because its log-plot is almost flat.

A. SR performance measures

This section reviews the two most popular measures of SR. These performance measures depend on the forcing sig-

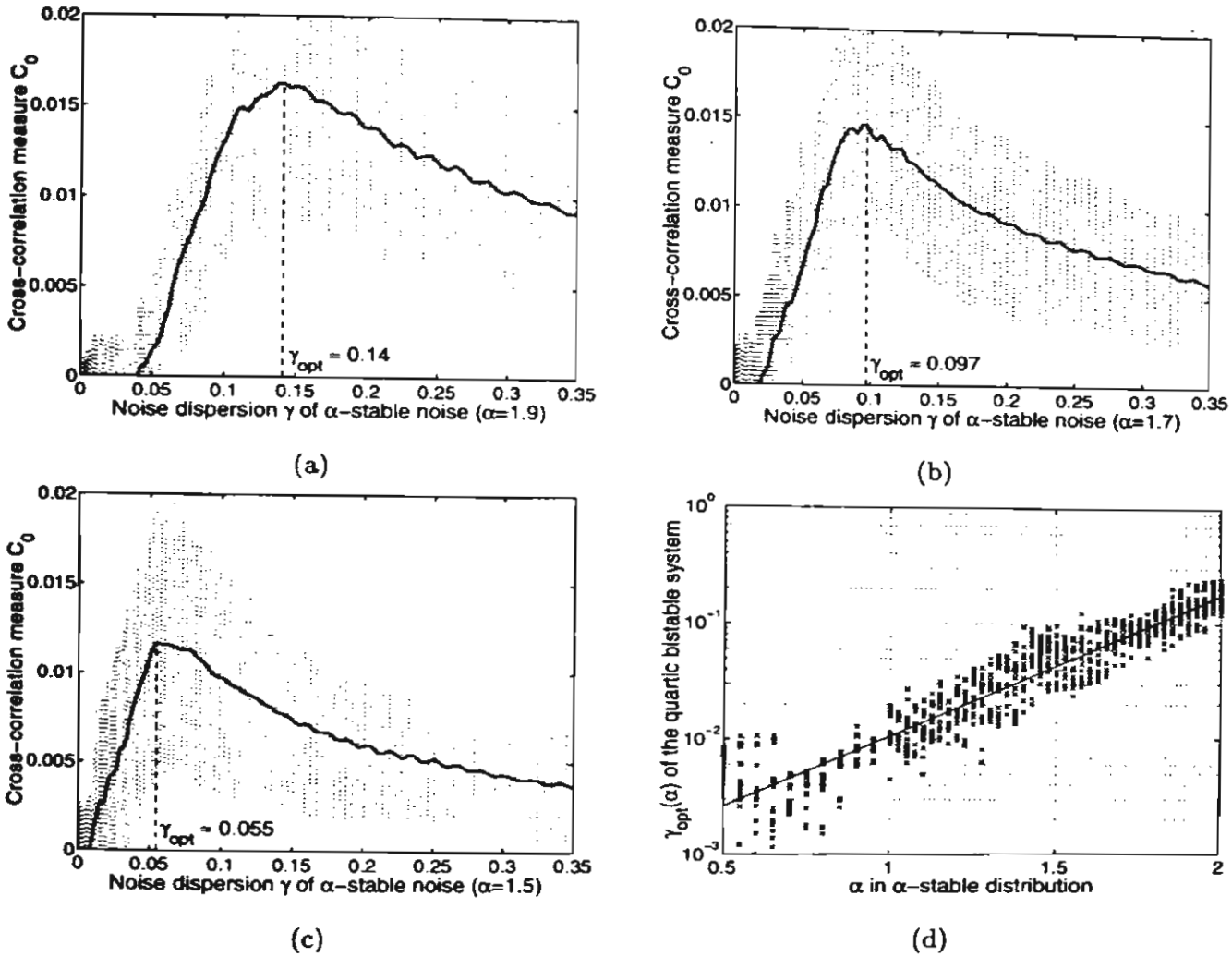

FIG. 5. The optimal dispersion γ_{opt} still becomes smaller as the infinite-variance noise becomes more impulsive for a cross-correlation performance measure. The dynamical system is the quartic bistable $\dot{x}=x-x^3+s+n$ modified for the saturation effects. The signal s is the sinusoid $s(t)=0.1\sin 2\pi(0.01)t$ as in Fig. 3 but with cross-correlation measure C_0. The plots (a)–(c) show the C_0-dispersion profiles for (a) $\alpha=1.9$, (b) $\alpha=1.7$, and (c) $\alpha=1.5$. The dotted lines show the absolute deviation between the smallest and largest cross-correlation outliers in each sample average of 100 outcomes. The vertical dashed lines show the SR effect or mode at the optimal noise dispersion γ_{opt}. The plot (d) shows the exponential law for optimal noise dispersion γ and parameter α.

nal and noise and can vary from system to system. There is no consensus in the SR literature on how to measure the SR effect.

a. Signal-to-noise ratio. The most common SR measure is some form of a signal-to-noise ratio (SNR) [18–22,37]. This seems the most intuitive measure even though there are many ways to define a SNR.

Suppose the input signal is the sinewave $s(t)=\varepsilon\sin\omega_0 t$. Then the SNR measures how much the system output $y=g(x)$ contains the input signal frequency ω_0:

$$\mathrm{SNR}=10\log_{10}\frac{S}{N} \tag{4}$$

$$=10\log_{10}\frac{S(\omega_0)}{N(\omega_0)}\ \mathrm{dB}. \tag{5}$$

The signal power $S=|Y(\omega_0)|^2$ is the magnitude of the output power spectrum $Y(\omega)$ at the input frequency ω_0. The background noise spectrum $N(\omega_0)$ at input frequency ω_0 is some average of $|Y(\omega)|^2$ at nearby frequencies [21,26,38]. The discrete Fourier transform (DFT) $Y[k]$ for $k=0,\ldots,L-1$ is an exponentially weighted sum of elements of a discrete-time sequence $\{y_0,y_1,\ldots,y_{L-1}\}$ of output signal samples

$$Y[k]=\sum_{t=0}^{L-1} y_t e^{-i(2\pi kt/L)}. \tag{6}$$

The signal frequency ω_0 corresponds to bin k_0 in the DFT for integer $k_0=L\Delta T f_0$ and for $\omega_0=2\pi f_0$. This gives the output signal in terms of a DFT as $S=|Y[k_0]|^2$. The noise

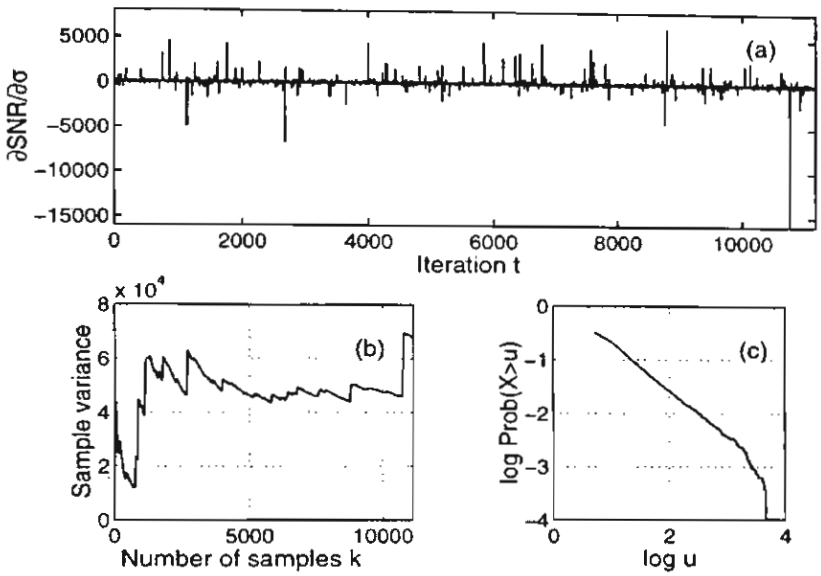

FIG. 6. Visual display of sample statistics of $\partial \mathrm{SNR}_t/\partial\sigma$ for the saturation-modified quartic bistable system $\dot{x}=x-x^3+s+n$ with sinusoidal input $s(t)=0.1\sin 2\pi(0.01)t$ and α-stable noise $n(t)$ with $\alpha=1.8$. The system has binary output $y(t)=\mathrm{sgn}(x(t))$. (a) Cauchy-like samples of $\partial \mathrm{SNR}_t/\partial\sigma$ at each iteration t at the noise dispersion $\gamma=0.1$ (which is the optimal dispersion for this signal system). The plot shows impulsiveness of the random variable $\partial \mathrm{SNR}_t/\partial\sigma$. (b) Test of infinite variance. The sequence of sample variances converges to a finite value if the underlying probability density has finite variance. Else it has infinite variance. (c) Log-tail test of the parameter α in for an α-stable bell curve. The test plots $\log \mathrm{Prob}(X>u)$ versus $\log_{10} u$ for large u. If the underlying density is α-stable with $\alpha<2$ then the slope of this plot is approximately $-\alpha$. This test found that $\alpha\approx 1$ and so the density was approximately Cauchy. The result is that we need to apply the Cauchy impulse suppressor [53] $\phi(x)=2x/(1+x^2)$ to the approximate SR gradient $\partial \mathrm{SNR}_t/\partial\sigma$.

power $N=N[k_0]$ is the average power in the adjacent bins $k_0-M,\ldots,\ k_0-1,\ k_0+1,\ldots,\ k_0+M$ for some integer M [22,39]

$$N=\frac{1}{2M}\sum_{j=1}^{M}\left(|Y[k_0-j]|^2+|Y[k_0+j]|^2\right). \tag{7}$$

There is no standard definition of system-level signal and noise in nonlinear systems. We work with a SNR that is easy to compute and that depends on standard spectral power measures in signal processing. We start with a sinewave input and view the output state $y(t)=g(x(t))$ of the dynamical system as a mixture of signal and noise. We arrange the DFT computation so that the energy of the sine term lies in frequency bin k_0. The squared magnitude of this energy spectrum $Y[k_0]$ acts as the system-level signal: $S=2|Y[k_0]|^2$. We view all else in the spectrum as noise: $N=P-S=P-2|Y[k_0]|^2$ where the total energy is $P=\Sigma_{k=0}^{L-1}|Y[k]|^2$. We ignore the factor L that scales S and P since the ratio S/N cancels its effect.

b. Cross-correlation measures. These "shape matchers" can measure SR when inputs are not periodic signals. Researchers coined the term "aperiodic stochastic resonance" [23,40–42] for such cases. They defined cross-correlation measures for the input signal s and the system response in terms of the mean transition rate r in the FHN model in Eqs. (16)–(18):

$$C_0=\max\{\overline{s(t)r(t+\tau)}\}, \tag{8}$$

$$C_1=\frac{C_0}{[\overline{s^2(t)}]^{1/2}\{\overline{[r(t)-\bar{r}(t)]^2}\}^{1/2}}. \tag{9}$$

where $\bar{x}$ is the time average $\bar{x}=1/T\int_0^T x(t)dt$.

B. SR systems and simulation models

The computer simulation uses a discrete version

$$x_{t+1}=x_t+\Delta T[f(x_t)+s_t]+\sqrt{\Delta T}\kappa w_t, \tag{10}$$

$$y_{t+1}=g(x_{t+1}), \tag{11}$$

with initial condition x_0 and output y_t. We assume that this discrete model applies to systems with α-stable noise. The zero location white α-stable random sequence $\{w_t\}$ has unit dispersion $\gamma_w=1$. So $n_t=\kappa w_t$ has dispersion $\gamma=\kappa^\alpha$. Note that a unit dispersion for Gaussian density (when $\alpha=2$) equals a variance of two. We tested the following six models:

(a) Quartic bistable system. The forced quartic bistable system has the form

$$\dot{x}=x-x^3+s(t)+n(t), \tag{12}$$

$$y(t)=\mathrm{sgn}(x(t)), \tag{13}$$

for binary output $y(t)$. We tested the quartic bistable system model with the sinusoid input $s(t)=\varepsilon\sin 2\pi f_0 t$ for $\varepsilon=0.1$

and $f_0=0.01$. The discrete version of the quartic bistable follows from Eqs. (10)–(11) as

$$x_{t+1}=x_t+\Delta T(x_t-x_t^3+s_t)+\sqrt{\Delta T}\kappa w_t, \tag{14}$$

$$y_{t+1}=\mathrm{sgn}(x_{t+1}). \tag{15}$$

We limit the magnitude of the system state x_t to 10 in the simulation model (14) to account for physical and computer saturation effects. We put $x_{t+1}=10$ when $x_{t+1}>10$ and put $x_{t+1}=-10$ when $x_{t+1}<-10$ in the discrete dynamic system (14). This gives a modified version of the quartic bistable system. The optimal dispersion γ_{opt} has the form $\gamma_{opt}(\alpha)=\kappa^{\alpha}$ for the noise scale κ in Eq. (14).

(b) FHN model. The forced FHN model has the form

$$\epsilon\dot{x}=-x\left(x^2-\frac{1}{4}\right)-z+A'+s'(t)+n'(t), \tag{16}$$

$$\dot{z}=x-z, \tag{17}$$

$$y(t)=x(t), \tag{18}$$

for $\epsilon=0.005$ and $A'=-(5/12\sqrt{3}+0.07)=-0.31056$ as in [42] and linear output $y(t)$. We use a sinusoidal input $s'(t)=\varepsilon\sin 2\pi f_0 t$ where $\varepsilon=0.01$ and $f_0=0.5$. We can rewrite Eqs. (16)–(18) as

$$\dot{x}=-\frac{1}{\epsilon}x\left(x^2-\frac{1}{4}\right)-\frac{1}{\epsilon}z+\frac{1}{\epsilon}A'+\frac{1}{\epsilon}s'(t)+\frac{1}{\epsilon}n'(t)$$

$$=-\frac{1}{\epsilon}x\left(x^2-\frac{1}{4}\right)-\frac{1}{\epsilon}z+A+s(t)+n(t), \tag{19}$$

$$\dot{z}=x-z, \tag{20}$$

$$y(t)=x(t), \tag{21}$$

for $A=A'/\epsilon$. Then Eqs. (10) and (11) give the discrete version to simulate the FHN model as

$$x_{t+1}=x_t+\Delta T\left[-\frac{1}{\epsilon\varepsilon}x\left(x^2-\frac{1}{4}\right)-\frac{1}{\epsilon}z+A+s_t\right]+\sqrt{\Delta T}\kappa w_t, \tag{22}$$

$$z_{t+1}=z_t+\Delta T(x_t-z_t), \tag{23}$$

$$y_{t+1}=x_{t+1}, \tag{24}$$

We also modify the recursive relation (22) to allow for saturation effects by requiring the magnitude of x_{t+1} not to exceed 2. The optimal dispersion γ_{opt} has the form $\gamma_{opt}(\alpha)=\kappa^{\alpha}$ for the noise scale κ in Eq. (22).

(c) Bistable potential neuron model [43]. The bistable potential neuron model [44] with stable white noise has the form

$$\dot{x}=-x+2\tanh x+s(t)+n(t), \tag{25}$$

$$y(t)=\mathrm{sgn}(x(t)). \tag{26}$$

The sinusoid input is $s(t)=\varepsilon\sin 2\pi f_0 t$ for $\varepsilon=0.1$ and $f_0=0.01$. The discrete version has the form

$$x_{t+1}=x_t+\Delta T(-x_t+2\tanh x_t+s_t)+\varepsilon\sqrt{\Delta T}\kappa w_t, \tag{27}$$

$$y_{t+1}=\mathrm{sgn}(x_{t+1}). \tag{28}$$

We test this neuron model with sinusoidal input $s(t)=\varepsilon\sin 2\pi f_0 t$ where $\varepsilon=0.1$ and $f_0=0.01$.

(d) Duffing oscillator [45]. The forced duffing oscillator has the form

$$\dot{x}=-0.15\dot{x}+x-x^3+\varepsilon\sin(\omega_0 t)+n(t), \tag{29}$$

$$y(t)=x(t), \tag{30}$$

We test the duffing oscillator with sinusoidal input $s(t)=\varepsilon\sin 2\pi f_0 t$ for $\varepsilon=0.3$ and $f_0=0.01$. The discrete version of the duffing oscillator has the form

$$x_{t+1}=x_t+\Delta T z_t, \tag{31}$$

$$z_{t+1}=z_t+\Delta T(-\delta z_t+x_t-x_t^3+s_t)+\kappa\sqrt{\Delta T}w_t, \tag{32}$$

$$y_{t+1}=\mathrm{sgn}(x_{t+1}). \tag{33}$$

(e) Threshold system [15,46–50]. The output y of a simple feedforward threshold system has the form

$$y_t=\mathrm{sgn}(s_t+n_t-\Theta)=\mathrm{sgn}(s_t+\kappa w_t-\Theta). \tag{34}$$

The optimal dispersion γ_{opt} has the form $\gamma_{opt}(\alpha)=\kappa^{\alpha}$ for κ in Eq. (34).

(f) Pulse system [51]. This doubly Poisson process generates a pulse train with probability r that depends on the input $V(t)=s(t)+n(t)$

$$r(V(t))=r(0)\exp(V(t)). \tag{35}$$

Here we let $r(0)=1$. The sinusoid input is $s(t)=\varepsilon\sin 2\pi f_0 t$ for $\varepsilon=0.5$ and $f_0=0.05$. The system generates an output $y(t)$ as a unit pulse with a rate $r(t)$.

I. Exponential law with linear least-squares fit of log data

The optimal dispersion $\gamma_{opt}(\alpha)$ of the system obeys the exponential law

$$\gamma_{opt}(\alpha)=c\,A^{\alpha} \tag{36}$$

for real constants c and A. Then

$$\log_{10}\gamma_{opt}(\alpha)=\log_{10}c+\alpha\log_{10}A=a\alpha+c' \tag{37}$$

for $a=\log_{10}A$ and $c'=\log_{10}c$. The least-squares method gives the a and c' values as

$$a=\frac{\sum_{i=1}^{N}(\alpha_i-\bar{\alpha})w_i}{\sum_{i=1}^{N}\alpha_i^2-N(\bar{\alpha})^2} \quad \text{and} \quad c'=\bar{w}-a\bar{\alpha}, \tag{38}$$

for N data pairs (α_i,w_i) where $w_i=\log_{10}\gamma_{opt}(\alpha_i)$ at the experiment i with the parameter α_i. This method is the same

TABLE I. Linear least-squares fit of the log of optimal dispersion γ and the parameter α in an α-stable density. The parameters a and c' relate $\log_{10}\gamma$ and α through a straight line: $\log_{10}\gamma(\alpha)=a\alpha+c'$.

	SNR		Cross correlation	
	Parameters	r^2	Parameters	r^2
Quartic bistable	$a=1.2444$, $c'=-3.3411$	0.8923	$a=1.2177$, $c'=-3.1889$	0.8463
FHN	$a=0.8622$, $c'=-2.7496$	0.9098	$a=0.6518$, $c'=-2.4869$	0.7510
Bistable neuron	$a=1.8552$, $c'=-3.9344$	0.9593	$a=1.9581$, $c'=-4.0252$	0.9641
Duffing oscillator	$a=0.7320$, $c'=-3.3057$	0.7444	$a=0.8912$, $c'=-3.3204$	0.8175
Threshold	$a=-0.5020$, $c'=0.1638$	0.9215	$a=-0.5036$, $c'=0.1658$	0.9196
Pulse	$a=0.0692$, $c'=0.2267$	0.0406	$a=0.2478$, $c'=0.2516$	0.3361

as the minimum variance method for arbitrary random variables and the maximum likelihood method for normal random variables [52].

The correlation coefficient r^2 indicates how good the linear model fits the data

$$r^2=\frac{\Sigma(\hat{w}_i-\bar{w})^2}{\Sigma(w_i-\bar{w})^2}=\frac{[\Sigma(\alpha_i-\bar{\alpha})(w_i-\bar{w})]^2}{\Sigma(ga_i-\bar{\alpha})^2\Sigma(w_i-\bar{w})^2}, \tag{39}$$

where $0\leq|r|\leq 1$ and $|r|=1$ iff $w_i=\hat{w}_i=a\alpha_i+c'$ for every i. The positive and negative signs reflect the positive and negative slopes.

2. Test results

Table I shows the parameters a and c' of the linear least-squares fit of logarithm of the optimal dispersion γ_{opt} and the parameter α. The correlation coefficients r^2 measure how well the regression $a\alpha+b$ fits the data and how much $\log_{10}\gamma_{opt}$ linearly depends on α. Figure 4 shows the SR-optimal dispersion $\gamma_{opt}(\alpha)$ versus the parameter α. The plots in Figs. 4(a)–4(d) for feedback systems agree with the exponential law. Figures 4(e) and 4(f) show the plots for the threshold and feedforward pulse systems. The correlation coefficients r^2 for the pulse system for both the SNR and cross-correlation measures are small due to the small slopes a and the large spread of the data $\log_{10}(\gamma_{opt})$. But their trends still show a linear relationship.

Note also that the slopes of the plots can be positive or negative or zero depending on the time scale factor of the dynamical system and on the noise when we consider the noise scale κ that gives the dispersion $\gamma=\kappa^\alpha$. Consider, for example, the two FHN models (16)–(18) and (19)–(21) are the same system. But the noise $n'(t)=\kappa' w(t)$ in Eq. (16) differs from the optimal noise $n(t)=\kappa w(t)$ in Eq. (19) by the scale ϵ. So at SR the two optimal noise scales obey the relation $\kappa'_{opt}=\epsilon\kappa_{opt}$. Then $\gamma'_{opt}(\alpha)=\kappa'^{\alpha}_{opt}(\alpha)=c(AB)^\alpha$ if $\gamma_{opt}(\alpha)=\kappa^{\alpha}_{opt}(\alpha)=cA^\alpha$. So the factor ϵ can change the slope of the plot from positive to negative for this FHN model.

IV. LEARNING THE OPTIMAL NOISE DISPERSION IN IMPULSIVE ENVIRONMENTS

We applied the stochastic SR gradient-ascent learning law of [15] to the problem of finding the optimal noise dispersion γ_{opt} for infinite-variance noise. This learning law has the form

$$\gamma_{t+1}=\gamma_t+\mu_t\frac{\partial\mathrm{SNR}}{\partial\gamma}, \tag{40}$$

where μ_t is a decreasing sequence of learning coefficients. A like learning law holds for the correlation measure in Eq. (9). The spectral relation $\mathrm{SNR}=S/N$ and the chain rule of calculus show that

$$\frac{\partial\mathrm{SNR}}{\partial\gamma}=\frac{\partial\mathrm{SNR}}{\partial S}\frac{\partial S}{\partial\gamma}+\frac{\partial\mathrm{SNR}}{\partial N}\frac{\partial N}{\partial\gamma} \tag{41}$$

$$=\frac{1}{N}\frac{\partial S}{\partial\gamma}-\frac{\mathrm{SNR}}{N}\frac{\partial N}{\partial\gamma}. \tag{42}$$

The first-order condition for an SR maximum is $\partial\mathrm{SNR}/\partial\gamma=0$. This leads to the optimality condition $S/N=S'/N'$ where $S'=\partial S/\partial\gamma$. But the optimality error process $\mathcal{E}=S/N-S'/N'$ itself is impulsive. Indeed a converging-variance test and log-tail test confirm that this random process obeys the highly impulsive Cauchy probability density (with $\alpha\approx 1$). Figure 6 shows samples of this Cauchy-like error process. These impulses destabilized all attempts to learn γ_{opt} with Eq. (42). This Cauchy impulsiveness holds for forcing noise with finite as well as infinite variance and for all the SR models and performance measures. It is systemic to the gradient-learning process. But its Cauchy nature suggests an immediate remedy. We can apply the well-known Cauchy impulse suppressor $\phi(z_t)=2z_t/(1+z_t^2)$ from the theory of robust statistics [53]. This gives the final robustified form of the learning law:

$$\gamma_{t+1}=\gamma_t+\mu_t\phi\left(\frac{\partial\mathrm{SNR}}{\partial\gamma}\right). \tag{43}$$

The robustified learning law (43) learned the optimal dispersions γ_{opt} in Figs. 1 and 3. It successfully found γ_{opt} for α values in the range [1.4, 2) for both the quartic bistable and Fitzhugh-Nagumo models but only for the SNR performance measure. The learning law often converged to γ_{opt} for α values in [1, 1.4) but with decreasing frequency and accuracy for the lower α values. The learning scheme often did not converge when the forcing noise was Cauchy ($\alpha=1$).

Learning with the SNR measure did not require knowledge of the system dynamics while learning with the correlation measure did require some knowledge of the system Jacobian. Learning is slow in any case because the system must in effect estimate at least part of the system dynamics based on the sampled SNR inputs to the learning process. The robustified gradient scheme (43) can use other performance measures or can include more information from the system dynamics to help the system more accurately estimate the stochastic term $\partial \mathrm{SNR}/\partial \gamma$.

V. CONCLUSION

We have shown that stochastic resonance is robust against noise outliers. Sufficiently large and sufficiently frequent noise impulses can overwhelm any SR system. But an SR effect still emerges even for the wide range of infinite-variance noise-types that lie between the extremes of the wildly impulsive Cauchy bell curve and the nonimpulsive Gaussian. The approximate exponential relationship $\gamma_{opt}(\alpha) = cA^{\alpha}$ shows this. This result is encouraging because all real noise is impulsive to some degree—the best-fit α is seldom the Gaussian case of $\alpha = 2$. This robustness favors engineering designs that may not conform to the ideal standards of Gaussian noise. It also suggests that SR may occur more widely in nature than many had believed.

The success of the dispersion-learning simulations further suggests that evolution could have tuned biological parameters to exploit the SR affect for signal detection in noisy environments. No living organism can control the noise structure of the environment. But gene selection over thousands of generations might act as if a gene pool slowly and noisily tuned its own noise parameters. Each act of reproductive fitness would count as only a lone noisy spike in evolution's learning process. The battle of genetic countermeasures between predator and prey suggest that if the predator or prey evolved SR-sensitive signal detection (as Moss [11,54] has shown for crayfish that use SR to detect a largemouth bass's periodic fin pattern or paddlefish [55] that use SR to detect plankton) then they would have to evolve new SR parameter settings as their opponents evolved new countermeasures.

The problem with such an SR evolutionary hypothesis is the Cauchy impulsiveness of gradient-ascent learning (40) for either a signal-to-noise or correlation performance measure. Biological systems would have to further evolve a robustifier of some sort to suppress extremely large learning outliers as Eq. (43) does with the Cauchy impulse suppresser. A meta-level threshold system might suffice for that task.

ACKNOWLEDGMENTS

National Science Foundation Grant Nos. ECS-9906251 and ECS-0070284 and the Thailand Research Fund Grant No. PDF/29/2543 partly supported this research.

[1] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, Tellus **34**, 10 (1982).
[2] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, SIAM (Soc. Ind. Appl. Math.) J. Appl. Math. **43**, 565 (1983).
[3] R. Benzi, A. Sutera, and A. Vulpiani, J. Phys. A **14**, L453 (1981).
[4] K. S. Brown, New Sci. **150**, 28 (1996).
[5] A. R. Bulsara and L. Gammaitoni, Phys. Today **49** (3), 39 (1996).
[6] J.-P. Eckmann and L. E. Thomas, J. Phys. A **15**, L261 (1982).
[7] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, Rev. Mod. Phys. **70**, 223 (1998).
[8] J. Glanz, Science **277**, 1758 (1997).
[9] P. Jung and K. Wiesenfeld, Nature (London) **385**, 291 (1997).
[10] F. Moss, D. Pierson, and D. O'Gorman, Int. J. Bifurcation Chaos Appl. Sci. Eng. **4**, 1383 (1994).
[11] F. Moss and K. Wiesenfeld, Sci. Am. **273**, 66 (1995).
[12] C. Nicolis, Tellus **34**, 1 (1982).
[13] K. Wiesenfeld and F. Moss, Nature (London) **373**, 33 (1995).
[14] F. Chapeau-Blondeau and X. Godivier, Phys. Rev. E **55**, 1478 (1997).
[15] L. Gammaitoni, Phys. Lett. A **208**, 315 (1995).
[16] S. Mitaim and B. Kosko, Proc. IEEE **86**, 2152 (1998).
[17] B. R. Parnas, IEEE Trans. Biomed. Eng. **43**, 313 (1996).
[18] S. Fauve and F. Heslot, Phys. Lett. **97A**, 5 (1983).
[19] R. F. Fox, Phys. Rev. A **39**, 4148 (1989).
[20] G. Hu, G. Nicolis, and N. Nicolis, Phys. Rev. A **42**, 2030 (1990).
[21] B. McNamara and K. Wiesenfeld, Phys. Rev. A **39**, 4854 (1989).
[22] T. Zhou and F. Moss, Phys. Rev. A **41**, 4255 (1990).
[23] J. J. Collins, C. C. Chow, A. C. Capela, and T. T. Imhoff, Phys. Rev. E **54**, 5575 (1996).
[24] J. J. Collins, C. C. Chow, and T. T. Imhoff, Nature (London) **376**, 236 (1995).
[25] A. R. Bulsara and A. Zador, Phys. Rev. E **54**, R2185 (1996).
[26] A. Neiman, B. Shulgin, V. Anishchenko, W. Ebeling, L. Schimansky-Geier, and J. Freund, Phys. Rev. Lett. **76**, 4299 (1996).
[27] M. Stemmler, Network Comput. Neural Syst. **7**, 687 (1996).
[28] J. M. Chambers, C. L. Mallows, and B. W. Stuck, J. Am. Stat. Assoc. **71**, 340 (1976).
[29] P. Tsakalides and C. L. Nikias, IEEE Trans. Signal Process. **44**, 1623 (1996).
[30] L. Breiman, *Probability* (Addison-Wesley, Reading, MA, 1968).
[31] W. Feller, *An Introduction to Probability Theory and Its Applications*, (Wiley, New York, 1966), Vol. II.
[32] M. Grigoriu, *Applied Non-Gaussian Processes* (Prentice Hall, Englewood Cliff, NJ, 1995).
[33] C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications* (Wiley, New York, 1995).
[34] B. Kosko, *Fuzzy Engineering* (Prentice Hall, Englewood Cliffs, NJ, 1996).

[35] V. Akgiray and C. G. Lamoureux, J. Bus. Econ. Stat. **7**, 85 (1989).
[36] H. Bergstrom, Ark. Math. **2**, 375 (1952).
[37] L. Gammaitoni, E. Menichella-Saetta, S. Santucci, F. Marchesoni, and C. Pressilla, Phys. Rev. A **40**, 2114 (1989).
[38] M. E. Inchiosa and A. R. Bulsara, Phys. Rev. E **53**, R2021 (1996).
[39] A. S. Asdi and A. H. Tewfik, in Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95), Detroit, MI, 1995, Vol. 2, pp. 1332–1335.
[40] D. R. Chialvo, A. Longtin, and J. Müller-Gerking, Phys. Rev. E **55**, 1798 (1997).
[41] J. J. Collins, C. C. Chow, and T. T. Imhoff, Phys. Rev. E **52**, R3321 (1995).
[42] C. Heneghan, C. C. Chow, J. J. Collins, T. T. Imhoff, S. B. Lowen, and M. C. Teich, Phys. Rev. E **54**, R2228 (1996).
[43] A. R. Bulsara, E. W. Jacobs, T. Zhou, F. Moss, and L. Kiss, J. Theor. Biol. **152**, 531 (1991).
[44] M. A. Cohen and S. Grossberg, IEEE Trans. Neural Netw. **SMC-13**, 815 (1983).
[45] G. Nicolis, C. Nicolis, and D. McKernan, J. Stat. Phys. **70**, 125 (1993).
[46] Z. Gingl, L. B. Kiss, and F. Moss, Europhys. Lett. **29**, 191 (1995).
[47] X. Godivier and F. Chapeau-Blondeau, Signal Process. **56**, 293 (1997).
[48] P. Jung, Phys. Rev. E **50**, 2513 (1994).
[49] P. Jung, Phys. Lett. A **207**, 93 (1995).
[50] A. Restrepo (Palacios), L. F. Zuluaga, and L. E. Pino, in Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Munich, Germany, 1997, Vol. III, pp. 2365–2368.
[51] S. M. Bezrukov and I. Vodyanoy, Nature (London) **385**, 319 (1997).
[52] A. Papoulis, *Probability and Statistics* (Prentice Hall, Englewood Cliffs, NJ, 1990).
[53] P. J. Huber, *Robust Statistics* (Wiley, New York, 1981).
[54] J. K. Douglass, L. Wilkens, E. Pantazelou, and F. Moss, Nature (London) **365**, 337 (1993).
[55] D. F. Russell, L. A. Willkens, and F. Moss, Nature (London) **402**, 291 (1999).

The Shape of Fuzzy Sets in Adaptive Function Approximation

Sanya Mitaim and Bart Kosko

***Abstract*—The shape of if-part fuzzy sets affects how well feedforward fuzzy systems approximate continuous functions. We explore a wide range of candidate if-part sets and derive supervised learning laws that tune them. Then we test how well the resulting adaptive fuzzy systems approximate a battery of test functions. No one set shape emerges as the best shape. The sinc function often does well and has a tractable learning law. But its undulating sidelobes may have no linguistic meaning. This suggests that the engineering goal of function-approximation accuracy may sometimes have to outweigh the linguistic or philosophical interpretations of fuzzy sets that have accompanied their use in expert systems. We divide the if-part sets into two large classes. The first class consists of n-dimensional joint sets that factor into n scalar sets as found in almost all published fuzzy systems. These sets ignore the correlations among vector components of input vectors. Fuzzy systems that use factorable if-part sets suffer in general from exponential rule explosion in high dimensions when they blindly approximate functions without knowledge of the functions. The factorable fuzzy sets themselves also suffer from what we call the second curse of dimensionality: The fuzzy sets tend to become binary spikes in high dimension. The second class of if-part sets consists of the more general but less common n-dimensional joint sets that do not factor into n scalar fuzzy sets. We present a method for constructing such unfactorable joint sets from scalar distance measures. Fuzzy systems that use unfactorable if-part sets need not suffer from exponential rule explosion but their increased complexity may lead to intractable learning laws and inscrutable if-then rules. We prove that some of these unfactorable joint sets still suffer the second curse of dimensionality of spikiness. The search for the best if-part sets in fuzzy function approximation has just begun.**

***Index Terms*—Adaptive fuzzy system, curse of dimensionality, fuzzy function approximation, fuzzy sets.**

I. The Shape of Fuzzy Sets: From Triangles to What?

WHAT is the best shape for fuzzy sets in function approximation? Fuzzy sets can have any shape. Each shape affects how well a fuzzy system of if-then rules approximate a function. Triangles have been the most popular if-part set shape but they surely are not the best choice [24], [32] for approximating nonlinear systems. Overlapped symmetric triangles or trapezoids reduce fuzzy systems to piecewise linear systems. Gaussian bell-curve sets give richer fuzzy systems with simple learning laws that tune the bell-curve means and variances. But this popular choice comes with a special cost: It converts fuzzy systems to radial-basis-function neural networks or to other well-known systems that predate fuzzy systems [3], [17], [20], [27], [28], [30]. These Gaussian systems make important benchmarks but there is no scientific advance involved in their rediscovery.

Triangles and Gaussian bell curves also do not represent the vast function space of if-part fuzzy sets. But then which shapes do? This question has no easy answer. A key part of the problem is that we do not know what should count as a meaningful taxonomy of fuzzy sets. We can distinguish continuous fuzzy sets from discontinuous sets, differentiable from nondifferentiable sets, monotone from nonmonotone sets, unimodal from multimodal sets, and so on. But these binary classes of fuzzy sets may still be too general to permit a fruitful analysis in terms of function approximation or in terms of other performance criteria. Yet a taxonomy requires that we draw lines somewhere through the function space of all fuzzy sets.

We draw two lines. The first line answers whether a joint fuzzy set is factorable or unfactorable. Consider any fuzzy set $A \subset R^n$ with arbitrary set function $a : R^n \rightarrow [0, 1]$ (or the slightly more general case where a maps R^n or some other space X into some connected real interval $[u, v] \subset R$). The multidimensional nature of fuzzy set A presents a structural question that does not arise in the far more popular scalar or one-dimensional case: Is A *factorable*? Does $A \subset R^n$ factor into a Cartesian product of n scalar fuzzy sets $A_j \subset R : A = A_1 \times \cdots \times A_n$?

The general answer is no. Factorability is rare in the space of all n-dimensional mappings of R^n into numbers. It corresponds to uncorrelatedness or independence in probability theory. Yet much analysis focuses on the factorable exceptions of hyperrectangles and multivariate Gaussian probability densities. And almost all published fuzzy systems use rules that deliberately factor the if-part sets into scalar sets. This often yields factorable joint set functions of the form $a_j(x) = a_j^1(x_1) \times \cdots \times a_j^n(x_n)$ or $a_j(x) = \min(a_j^1(x_1), \ldots, a_j^n(x_n))$. Consider this rule for a simple air-conditioner controller: "If the air is warm and the humidity is high then set the blower to fast." A triangle or trapezoid or bell curve might describe the fuzzy subset of warm air temperatures or of high humidity values. A product of these two scalar sets forms a factorable fuzzy subset $A_1 \times A_2 \subset R^2$. But users tend not to work with even simple unfactorable two-dimensional (2-D) sets such as ellipsoids: "If the temperature-humidity values lie in the warm-high planar ellipsoid then set the motor speed to fast." Few unfactorable fuzzy subsets of the plane or of R^n are as simple geometrically or as tractable mathematically as ellipsoids [1], [2].

Below we study how well feedforward additive fuzzy systems can approximate test functions for both adaptive factorable and unfactorable if-part fuzzy sets. We first derive supervised learning laws for a wide range of fuzzy sets of different shape and then test them against one another in terms of how accurately they approximate the test functions in a squared-error

Manuscript received November 19, 1999; revised April 13, 2001. This work was supported in part by the National Science Foundation under Grant ECS-0070284 and in part by the Thailand Research Fund under Grant PDF/29/2543.

S. Mitaim is with the Department of Electrical Engineering, Thammasat University, Pathumthani 12121, Thailand.

B. Kosko is with the Department of Electrical Engineering—Systems, Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564 USA.

Publisher Item Identifier S 1063-6706(01)06658-9.

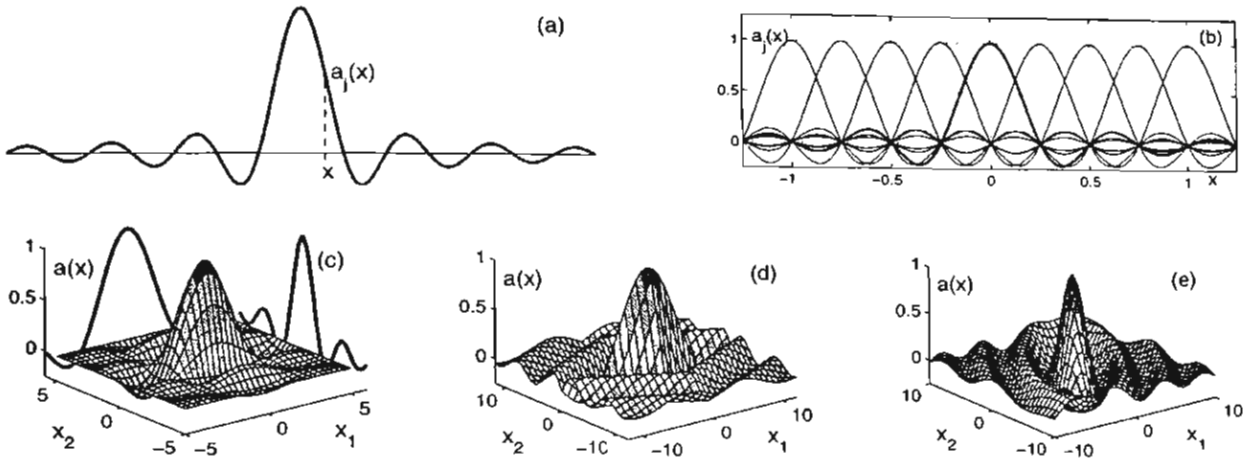

Fig. 1. Samples of sinc set functions for one-input and two-input cases. (a) Scalar sinc set function for one-input case. (b) Nine scalar sinc set functions for input x. All sinc set functions have the same width but their centers differ. (c) Product sinc set function for two-input case. The set function has the form $a_j(x) = a_j(x_1, x_2) = a_j^1(x_1) \times a_j^2(x_2)$. The shadows show the scalar sinc set functions $a_j^i : R \to R$ for $i = 1, 2$ that generate $a_j : R^2 \to R$. (d) Joint l^1 metrical sinc set function: $a_j(x) = \mathrm{sinc}(d_j^1(x, m_j))$. (e) Joint quadratic metrical sinc set function: $a_j(x) = \mathrm{sinc}(d_j^2(x, m_j))$.

sense. Then we form factorable n-dimensional fuzzy sets from the scalar factors and compare them both against one another and against some new unfactorable joint fuzzy sets. Exponential rule explosion severely constrains the extent of the simulations. We also uncover a second curse of dimensionality: Factorable sets tend toward binary spikes in high dimension. Unfactorable sets need not suffer exponential rule explosion. But we prove that some of them also suffer from spikiness in high dimensions.

We draw a second line between parametrized and nonparametrized fuzzy sets. We study only parametrized fuzzy sets because only for them could we define learning laws (that tune the parameters). We did not study recursive fuzzy sets such as those that can arise with B-splines [33] or other recursive algorithms. It also is not clear how to fairly compare parametrized if-part set functions with nonparametrized set functions for the task of *adaptive* function approximation.

The simulation results do not pick a clear-cut winner. Nor would we expect them to do so given the ad hoc nature of our choices of both candidate set functions and test functions. But the results do suggest that some nonobvious set functions should be among those that a fuzzy engineer considers when building or tuning a fuzzy system. Along the way we also developed an extensive library of new set functions and derived their often quite complex learning laws.

Perhaps the most surprising and durable finding is that the sinc function $(\sin x/x)$ of signal processing often converges fastest and with greatest accuracy among candidates that include triangles, Gaussian and Cauchy bell curves, and other familiar set shapes. This appears to be the first use of the sinc function as a fuzzy set. We could find no theoretical reason for its performance as a nonlinear interpolator in a fuzzy system despite its well-known status as the linear interpolator in the Nyquist sampling theorem and its signal-energy optimality properties [21]. We also combined two hyperbolic tangents to give a new bell curve that often competes favorably with other if-part set candidates. We call this new bell curve the difference hyperbolic tangent [18].

Fig. 1 shows scalar and joint sinc set functions. Fig. 1(a) shows the decaying sidelobes that can take on negative values. This requires that we view the sinc as a generalized fuzzy set [14] whose set function maps into a totally ordered interval that includes negative values: $a : R \to [-0.217, 1]$. An exercise shows that such a bipolar set-function range does not affect the set-theoretic structure of A in terms intersection, union, or complementation because the corresponding operations of minimum, maximum, and order reversal depend on only the total ordering (with a like result for triangular or t-norms [8]). Fig. 1(c) shows the 2-D factorable sinc that results when we multiply two scalar sinc functions as we might do to compute the degree to which a two-vector input $x = (x_1, x_2)$ fires the two if-part factors of a rule of the form "If X_1 is A_1 and X_2 is A_2 then Y is B_1." Fig. 1(d) and (e) show two new unfactorable 2-D set functions built from the scalar sinc function and a distance metric.

Below we derive the supervised learning law that tunes these sinc set functions given input–output samples from a test function. The factorable joint set functions are far easier to tune than are the unfactorable sets because we need only add one more term to a partial-derivative expansion and then multiply the results for tuning the individual factors. Fig. 2 shows how a 2-D factorable or product sinc set evolves as the process of supervised learning unfolds when a sinc-based fuzzy system approximates a test function.

The sinc finding raises a broader issue: Does an if-part fuzzy set need to have a linguistic meaning? The very definition of the sinc set function $a : R \to [-0.217, 1]$ already requires that we broaden our usual notion of "degrees" that range from 0% to 100% to a more general totally ordered scale. But the sinc's undulating and decaying sidelobes admit no easy linguistic interpretation. We could simply think of the smooth bell-shaped *envelope* of the sinc and treat it as we would any other unimodal curve that stands for warm air or high humidity or fast blower speeds. That would solve the problem in practice and would allow engineers to safely interpret a domain expert's fuzzy concepts as appropriately centered and scaled sinc sets. But that would not address the conceptual problem of how to make sense

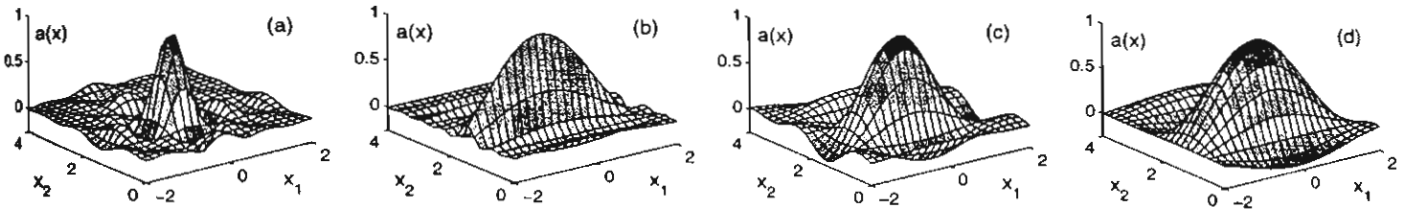

Fig. 2. Samples of evolution of a product sinc if-part set function in an adaptive function approximator. Supervised learning tunes the parameters of the product sinc set function such as its center and width on each parameter axis x_1 and x_2: (a) a sinc set function at initial state, (b) the same sinc set after 10 epochs of learning, (c) after 500 epochs, and (d) the sinc set converges after 3000 epochs

of all those local minima and maxima in such a multimodal set function.

A pragmatic answer is that a given if-part fuzzy set need not have a precise linguistic meaning or have any tie to natural language at all. Function approximation is a global property of a fuzzy system. If-part fuzzy sets are local parts of local if-then rules. The central goal is accurate function approximation. This can outweigh the linguistic and philosophical concerns that may have attended earlier fuzzy expert systems. Engineers designed many of those earlier systems not to accurately approximate some arbitrary nonlinear function but to accurately model an expert's knowledge as the expert stated it in if-then rules.

So the real issue is the gradual shift in performance criteria from accuracy of linguistic modeling to accuracy of function approximation. Progress in fuzzy systems calls into question the earlier goal of simply modeling what a human says. That goal remains important for many applications and no doubt always will. But it should not itself constrain the broader considerations of fuzzy function approximation. The function space of all if-part fuzzy sets is simply too vast and too rich for natural language to restrict searches through it.

II. Fuzzy Function Approximation and Two Curses of Dimensionality

We work with scalar-valued additive fuzzy systems $F : R^n \rightarrow R$. These systems approximate a function $f : R^n \rightarrow R$ by covering the graph of f with fuzzy rule patches and averaging patches that overlap [14]. An if-then rule of the form "If X is A then Y is B" defines a fuzzy Cartesian patch $A \times B$ in the input–output space $X \times Y$. The rules can use fuzzy sets of any shape for either their if-part sets A or then-part sets B. This holds for the feedforward standard additive model (SAM) fuzzy systems discussed below. Their generality further permits any scheme for combining if-part vector components because all theorems assume only that the set function maps to numbers as in $a : R^n \rightarrow [u, v]$. The general fuzzy approximation theorem [11] also allows any choice of if-part set or then-part sets for a general additive model and still allows any choice of if-part set for the SAM case that in turn includes most fuzzy systems in use [15].

The fuzzy approximation theorem does not say which shape is the best shape for an if-part fuzzy set or how many rules m a fuzzy system should use when it approximates a function. The shape of if-part sets A_j affects how well the feedforward SAM F approximates a function f and how quickly an adaptive SAM F approximates it when learning based on input–output samples from f tunes the parameters of A_j and the centroids c_j and volumes V_j of the then-part set B_j. The shape of then-part sets B_j does not affect the *first-order* behavior of a feedforward SAM F beyond the effect of the volume V_j and centroid c_j. This holds because the SAM output computes only a convex-weighted sum of the then-part centroids c_j for each vector input x

$$F(x) = \sum_{j=1}^{m} p_j(x) c_j \tag{1}$$

where $p_j(x) \geq 0$ and $\sum_{j=1}^{m} p_j(x) = 1$ for each $x \in R^n$ as defined in (6). p_j depends on B_j only through its volume or area V_j (and perhaps through its rule weight). We also note that (1) and (2) imply that $F(x) = E[Y \mid X = x]$ [14]. But the shape of B_j does affect the *second-order* uncertainty or conditional variance $V[Y \mid X = x]$ of the SAM output $F(x)$ [14]

$$V[Y \mid X = x] = \sum_{j=1}^{m} p_j(x) \sigma_{B_j}^2 + \sum_{j=1}^{m} p_j(x)[c_j - F(x)]^2 \tag{2}$$

where $\sigma_{B_j}^2$ in an SAM is the then-part set variance

$$\sigma_{B_j}^2 = \int_{-\infty}^{\infty} (y - c_j)^2 p_{B_j}(y)\, dy \tag{3}$$

and where $p_{B_j}(y) = b_j(y)/V_j$ is an integrable probability density function and $b_j : R \rightarrow [0, 1]$ is the integrable set function of then-part set B_j. The first term on the right side of (2) gives an input-weighted sum of the then-part set uncertainties. The second term measures the interpolation penalty that results from computing the SAM output $F(x)$ in (1) as simply the weighted sum of centroids. The output conditional variance (2) further simplifies if all then-part sets B_j have the same shape and thus all have the same inherent uncertainty σ^2

$$V[Y \mid X = x] = \sigma^2 + \sum_{j=1}^{m} p_j(x)[c_j - F(x)]^2. \tag{4}$$

So a given input x minimizes the system uncertainty or gives an output $F(x)$ with maximal confidence if it fires the jth rule dead-on (so $F(x) = c_j$) and does not fire the other $m-1$ rules at all ($p_k(x) = 0$ for $k \neq j$). This justifies the common practice of centering a symmetric unimodal if-part fuzzy set A_j at a point where the other $m-1$ if-part sets have zero membership degree. It does not justify the equally common practice of ignoring the thickness or thinness of the then-part sets B_j and even replacing them with the maximally confident choice of binary "singleton" spikes centered at the centroid c_j. The second-order structure of

fuzzy system's output depends crucially on the size and shape of the then-part sets B_j.

We allow learning to tune the volumes V_j and centroids c_j of the then-part sets B_j in our adaptive function-approximation simulations. A then-part set B_j with volume V_j and centroid c_j can have an infinitude of shapes. And again many of these shapes will change the output uncertainty in (2) or (4). But we too shall ignore the second-order behavior that (2) and (4) describe.

High dimensions present further problems for fuzzy function approximation. Feedforward fuzzy systems suffer at least two curses of dimensionality. The first is the familiar exponential rule explosion. This results directly from the factorability of if-part fuzzy sets in fuzzy if-then rules. The second curse is one that we call the second curse of dimensionality: factorable if-part sets tend to binary spikes as the dimension n increases.

Consider first rule explosion for blind function approximation. Suppose we can factor the if-part fuzzy set A : $A = A_1 \times \cdots A_n$. Nontrivial if-then rules require that we use at least two scalar factors for each of the n orthogonal axes in R^n as in the minimal fuzzy partition of air temperatures into warm and not-warm temperatures or into low and high temperatures. A fuzzy system must cover the graph of the function f with rule patches. That entails that the if-part sets cover the system's domain—else the fuzzy system F would not be defined on those regions of the input space. So such a rule-patch cover of the domain of a fuzzy system $F : C \subset R^n \rightarrow R$ entails a rule explosion on the order of k^n where C is some compact subset of R^n. We will for convenience often denote functions as $F : R^n \rightarrow R$ or as $a : R^n \rightarrow [0, 1]$ where we understand that the domain is only some compact subset of R^n.

There is a related exception that deserves comment. Watkins [31], [32] has shown that if we not only know the functional form of f but build it into the very structure of the if-part sets A_j then we can exactly *represent* many functions in the sense of $F(x) = f(x)$ for all x and can do so with a number of rules that grows linearly with the dimension n. This does not apply in *blind* approximation where we pick the tunable if-part sets A_j in advance and then train them and other parameters based on exact or noisy input–output samples from the approximand function f. But it suggests that there may be many types of middle ground where partial knowledge of f may reduce the rule complexity from exponential to polynomial or perhaps to some other tractable function of dimension.

All factorable if-part sets suffer the second curse of dimensionality. They ignore input structure and collapse to binary-like spikes in high dimensions. The separate factors a_j^i ignore correlations and other nonlinearities among the input variables [5]. This structure can be quite complex in high dimensions. The product form $a_j^1(x_1) \times \cdots \times a_j^n(x_n)$ tends toward a spike in R^n for large n when $a_j^i < 1$. The Borel–Cantelli lemma of probability theory shows that $\min(a_j^1(x_1), \ldots, a_j^n(x_n))$ tends to zero with probability one [9] as $n \rightarrow \infty$ if the random sequence $x_1, x_2, \ldots$ is independent and identically distributed. This also holds for *any* t-norm combination of factors because of the generalized t-norm bound $T(a_j^1(x_1), \ldots, a_j^n(x_n)) \leq \min(a_j^1(x_1), \ldots, a_j^n(x_n))$. Factorable joint set functions degenerate in high dimensions. This curse of dimensionality can combine with the better known curse of exponential rule explosion. The result can be a function approximator with a vast set of spiky rules.

Joint unfactorable sets tend to preserve input correlations [5]. They need not collapse to spikes in high dimensions or suffer from the like rotten-apple effect of falling to zero when just one term equals zero. This also suggests that some unfactorable joint fuzzy sets may lessen or even defeat the curse of dimensionality.

The second part of this paper shows how to create and tune metrical joint set functions. These joint set functions preserve at least the metrical structure of inputs and do not try to factor a nonlinear function into a product or other combination of n terms. The idea is to use one well-behaved scalar set function like $\mathrm{sinc}(x)$ [18] and apply it to an n-dimensional distance function $d_j(x)$ rather than multiply n of the scalar set functions: $a_j(x) = \mathrm{sinc}(d_j(x))$ rather than $a_j(x) = \prod_{i=1}^n \mathrm{sinc}(x_i)$. Then supervised learning tunes the metrical joint set function as it tunes the metric. The next section reviews the standard additive fuzzy systems that we use to derive parameter learning laws and to test candidate if-part sets in terms of their accuracy of function approximation.

III. Additive Fuzzy Systems and Function Approximation

This section reviews the basic structure of additive fuzzy systems. The Appendix reviews and extends the more formal math structure that underlies these adaptive function approximators.

A fuzzy system $F : R^n \rightarrow R^p$ stores m rules of the word form "If $X = A_j$ Then $Y = B_j$" or the patch form $A_j \times B_j \subset X \times Y = R^n \times R^p$. The if-part fuzzy sets $A_j \subset R^n$ and then-part fuzzy sets $B_j \subset R^p$ have set functions $a_j : R^n \rightarrow [0, 1]$ and $b_j : R^p \rightarrow [0, 1]$. Generalized fuzzy sets [14] map to intervals other than $[0, 1]$. The system can use the joint set function a_j or some factored form such as $a_j(x) = a_j^1(x_1) \ldots a_j^n(x_n)$ or $a_j(x) = \min(a_j^1(x_1), \ldots, a_j^n(x_n))$ or any other conjunctive form for input vector $x = (x_1, \ldots, x_n) \in R^n$ [10].

An additive fuzzy system [10], [11] sums the "fired" then-part sets B_j'

$$B(x) = \sum_{j=1}^m w_j B_j' = \sum_{j=1}^m w_j a_j(x) B_j. \tag{5}$$

Fig. 3(a) shows the parallel fire-and-sum structure of the SAM. These nonlinear systems can uniformly approximate any continuous (or bounded measurable) function f on a compact domain [19]. Engineers often apply fuzzy systems to problems of control [4] but fuzzy systems can also apply to problems of communication [22] and signal processing [5], [6] and other fields.

Fig. 3(b) shows how three rule patches can cover part of the graph of a scalar function $f : R \rightarrow R$. The patch-cover structure implies that fuzzy systems $F : R^n \rightarrow R^p$ suffer from *rule explosion* in high dimensions. A fuzzy system F needs on the order of k^{n+p-1} rules to cover the graph and thus to approximate a vector function $f : R^n \rightarrow R^p$. Optimal rules can help deal with the exponential rule explosion. Lone or local mean-

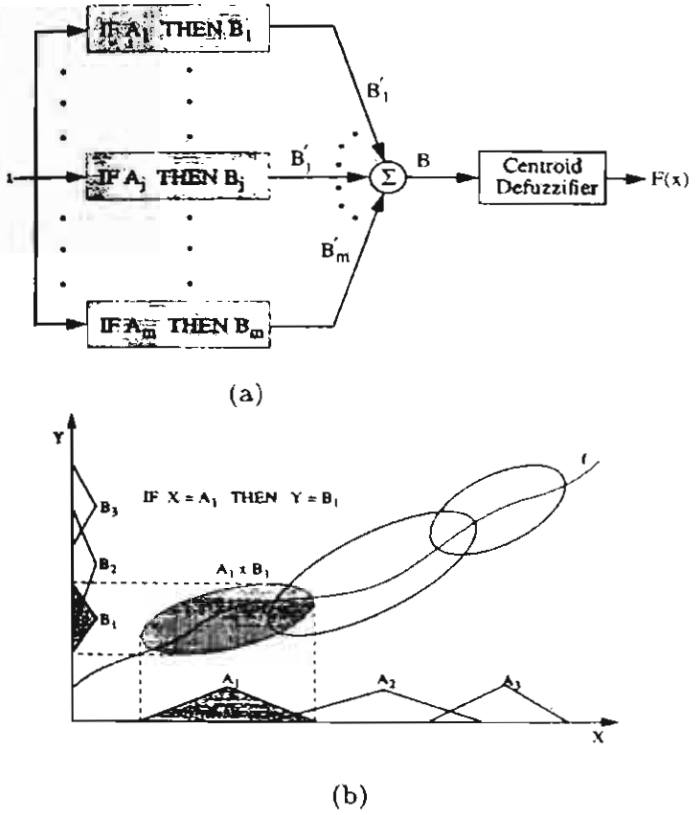

Fig. 3. Feedforward fuzzy function approximator. (a) The parallel associative structure of the additive fuzzy system $F: R^n \to R^p$ with m rules. Each input $x_0 \in R^n$ enters the system F as a numerical vector. At the set level x_0 acts as a delta pulse $\delta(x - x_0)$ that combs the if-part fuzzy sets A_j and gives the m set values $a_j(x_0) = \int_{R^n} \delta(x - x_0) a_j(x)\, dx$. The set values "fire" or scale the then-part fuzzy sets B_j to give B'_j. An SAM scales each B_j with $a_j(x)$. Then the system sums the B'_j sets to give the output "set" B. The system output $F(x_0)$ is the centroid of B. (b) Fuzzy rules define Cartesian rule patches $A_j \times B_j$ in the input–output space and cover the graph of the approximand f. This leads to exponential rule explosion in high dimensions. Optimal lone rules cover the extrema of the approximand as in Fig. 4.

squared optimal rule patches cover the extrema of the approximand f [13], [14]. They "patch the bumps" as in Fig. 4. The Appendix presents a simple proof of this fact. Better learning schemes move rule patches to or near extrema and then fill in between extrema with extra rule patches if the rule budget allows.

The scaling choice $B'_j = a_j(x)B_j$ gives an SAM. The Appendix further shows that taking the centroid of $B(x)$ in (5) gives the following SAM ratio [10], [11], [13], [14]:

$$F(x) = \frac{\sum_{j=1}^{m} w_j a_j(x) V_j c_j}{\sum_{j=1}^{m} w_j a_j(x) V_j} = \sum_{j=1}^{m} p_j(x) c_j. \tag{6}$$

Here V_j is the finite positive volume or area of then-part set B_j and c_j is the centroid of B_j or its center of mass. The convex weights $p_1(x), \ldots, p_m(x)$ have the form $p_j(x) = (w_j a_j(x) V_j / \sum_{i=1}^{m} w_i a_i(x) V_i)$. The convex coefficients $p_j(x)$ change with each input vector x. Sections V and VIII derive the gradient learning laws of all parameters of the SAM for different shapes of if-part sets.

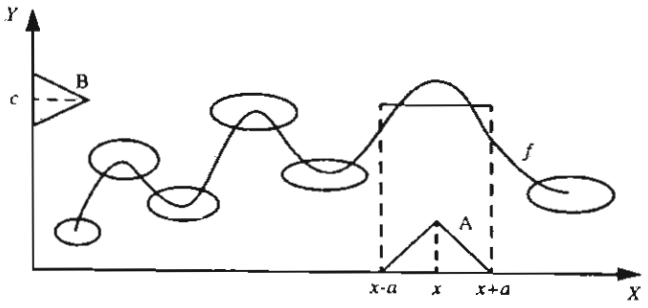

Fig. 4. Lone optimal fuzzy rule patches cover the extrema of approximand f. A lone rule defines a flat line segment that cuts the graph of the local extremum in at least two places. The mean value theorem implies that the extremum lies between these points. This can reduce much of fuzzy function approximation to the search for zeroes $\dot{x}$ of the derivative map $f' : f'(\dot{x}) = 0$.

IV. Scalar and Joint Factorable Fuzzy Set Functions

A scalar set function $a_j : R \to [0, 1]$ measures the degree to which input $x \in R$ belongs to the fuzzy or multivalued set $A_j : a_j(x) = \text{Degree}(x \in A_j)$. A joint factorable set $A_j \subset R^n$ derives from n scalar sets $A'_j \subset R$. Any conjunctive operator such as a t-norm can combine n scalar sets to obtain a joint factorable set.

A. Scalar Fuzzy Sets

We tested a wide range of if-part set functions. Below we list the scalar form of most of these set functions. The sinc function was multimodal and could take on negative values in $[-0.217, 1]$. We viewed these negative values as low degrees of set membership.

1) *Triangle set function.* We define the triangle set function as a three-tuple (l_j, m_j, r_j) where $l_j > 0$ and $r_j > 0$. $m_j \in R$ denotes the location of a peak of the triangle

$$a_j(x) = \begin{cases} 1 - \dfrac{m_j - x}{l_j}, & \text{if } m_j - l_j \le x \le m_j \\ 1 - \dfrac{x - m_j}{r_j}, & \text{if } m_j < x \le m_j + r_j \\ 0, & \text{else} \end{cases} \tag{7}$$

We can also define the symmetric triangle set function with two parameters that are its *center* m_j and *width* d_j as

$$a_j(x) = \begin{cases} 1 - \left| \dfrac{x - m_j}{d_j} \right|, & \text{if } |x_j - m_j| < d_j \\ 0, & \text{else.} \end{cases} \tag{8}$$

2) *Trapezoid set function.* We define the trapezoid set function as a four-tuple (l_j, ml_j, mr_j, r_j) where $ml_j \le mr_j \in R$. $l_j > 0$ and $r_j > 0$ denote the distance of the support of a function to the left and right of ml_j and mr_j. We can view the *center* as $m_j = (1/2)(ml_j + mr_j)$

$$a_j(x) = \begin{cases} 1 - \frac{ml_j - x}{l_j}, & \text{if } ml_j - l_j \le x \le ml_j \\ 1, & \text{if } ml_j \le x \le mr_j \\ 1 - \frac{x - mr_j}{r_j}, & \text{if } mr_j < x \le mr_j + r_j \\ 0, & \text{else} \end{cases} \tag{9}$$

3) *Clipped-parabola (Quadratic) set function.* A clipped-parabola set function (or quadratic set function) centered at m_j and with "width" d_j has the form

$$a_j(x) = \begin{cases} 1 - \left(\frac{x - m_j}{d_j}\right)^2, & \text{if } \left(\frac{x - m_j}{d_j}\right)^2 < 1 \\ 0, & \text{else} \end{cases} \quad (10)$$

This quadratic set function differs from the quadratic set function in [26].

4) *Gaussian set function.* The Gaussian set function depends on the mean m_j and standard deviation $d_j/\sqrt{2}$

$$a_j(x) = \exp\left\{-\left(\frac{x - m_j}{d_j}\right)^2\right\}. \quad (11)$$

5) *Cauchy set function.* The Cauchy set function is a bell curve with thicker tails than the Gaussian bell curve and with infinite variance and higher order moments [5]

$$a_j(x) = \frac{1}{1 + \left(\frac{x - m_j}{d_j}\right)^2}. \quad (12)$$

6) *Laplace set function.* The Laplace set function is an exponential curve

$$a_j(x) = \exp\left\{-\frac{|x - m_j|}{d_j}\right\} \quad (13)$$

where m_j is the center and $d_j > 0$ picks the decay rate of the curve.

7) *Sinc set function.* We define the sinc set function centered at m_j and *width* $d_j > 0$ as

$$a_j(x) = \sin\left(\frac{x - m_j}{d_j}\right) \Big/ \left(\frac{x - m_j}{d_j}\right). \quad (14)$$

The sinc set function is a map $a_j : R \to [-0.217, 1]$. So the denominator of a sinc SAM can in theory become zero or negative. The system design must take care when these negative set values enter the SAM ratio in (6). We set a logic flag to check if the denominator is zero or negative.

8) *Logistic set function.* The logistic or sigmoid function has the form of $S_j(x) = 1/(1 + \exp\{-x\})$. We define a symmetric logistic set function centered at m_j with *width* $d_j > 0$ as

$$a_j(x) = 2S\left(-\left(\frac{x - m_j}{d_j}\right)^2\right) = \frac{2}{1 + e^{\left(\frac{x - m_j}{d_j}\right)^2}}. \quad (15)$$

The factor 2 gives $\max_{x \in R} a_j(x) = 1$.

9) *Hyperbolic tangent set function.* This set function has the form

$$a_j(x) = 1 + \tanh\left(-\left(\frac{x - m_j}{d_j}\right)^2\right) \quad (16)$$

where m_j and d_j define the center and the width of the bell curve.

10) *Hyperbolic secant set function.* Again m_j and $d_j > 0$ define the center and width of this scalar set function

$$a_j(x) = \operatorname{sech}\left(\frac{x - m_j}{d_j}\right). \quad (17)$$

11) *Differential logistic set function.* The derivative of the logistic function is a bell curve form of probability density function. $S'(x) = S(x)(1 - S(x))$ holds for a logistic function $S(x) = 1/(1 + \exp\{-x\})$. So we define this new set function as

$$a_j(x) = 4S\left(\frac{x - m_j}{d_j}\right)\left[1 - S\left(\frac{x - m_j}{d_j}\right)\right]. \quad (18)$$

The factor 4 gives $\max_{x \in R} a_j(x) = 1$.

12) *Difference logistic set function.* The logistic or sigmoid function with *steepness* $\alpha_j > 0$ has the form of $S_j(x) = 1/(1 + \exp\{-\alpha_j x\})$. We define a symmetric logistic set function centered at m_j with *width* $l_j > 0$ as

$$a_j(x) = \frac{1}{D_j}[S_j(x - m_j + l_j) - S_j(x - m_j - l_j)], \quad (19)$$

The normalizer $D_j = S_j(l_j) - S_j(-l_j)$ ensures that $\max_{x \in R} a_j(x) = 1$.

13) *Difference hyperbolic tangent set function.* This new set function has the difference form

$$a_j(x) = \frac{1}{D_j}\left[\tanh\left(\frac{x - m_j + l_j}{d_j}\right) - \tanh\left(\frac{x - m_j - l_j}{d_j}\right)\right]. \quad (20)$$

This results in a bell curve. The term $l_j > 0$ defines the "width" of the function and $D_j = 2\tanh(l_j/d_j)$ gives the normalization factor.

Fig. 5 plots the scalar set functions for sample choices of parameters. Simulations in Section VI compare how these scalar set functions perform in adaptive fuzzy function approximation in terms of squared error.

B. Joint Factorable Sets: Product Set Functions

This class includes joint set functions $a_j : R^n \to [0, 1]$ that factor $a_j(x) = g(a_j^1(x_1), \ldots, a_j^n(x_n))$ for some function $g : [0, 1]^n \to [0, 1]$. The popular factorable joint set functions combine the scalar set functions with product

$$a_j(x) = a_j^1(x_1) \times \cdots \times a_j^n(x_n) \quad (21)$$

or other t-norms such as min

$$a_j(x) = \min(a_j^1(x_1), \ldots, a_j^n(x_n)) \quad (22)$$

for scalar set functions $a_j^i : R \to [0, 1]$. We form the product set functions from scalar set functions in Section IV-A as in Fig. 6. Section VI compares the results of adaptive function approximation of these set functions for two- and three-input cases.

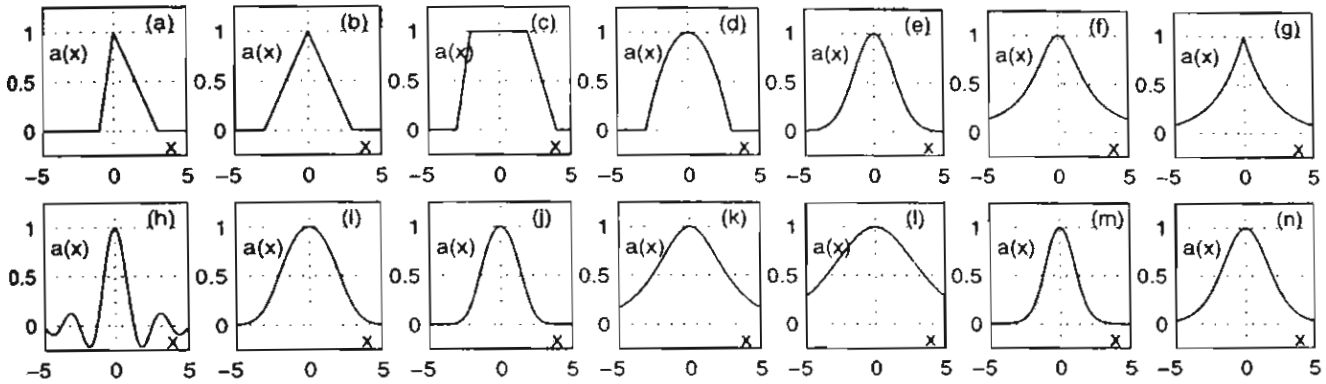

Fig. 5. Set functions centered at $m = 0$. (a) Triangle: $l = 1$ and $r = 3$. (b) Symmetric triangle: $d = 3$. (c) Trapezoid: $l = 1$, $ml = -2$, $mr = 2$, and $r = 2$. (d) Parabola: $d = 2$. (e) Gaussian: $d = 2$. (f) Cauchy: $d = 2$. (g) Laplace: $d = 2$. (h) Sinc: $d = 0.4$. (i) Logistic: $d = 2$. (j) Hyperbolic Tangent: $d = 2$. (k) Hyperbolic secant: $d = 2$. (l) Differential logistic: $d = 2$. (m) Difference logistic: $a = 2$ and $l = 1$. (n) Difference hyperbolic tangent: $d = 2$ and $l = 1$.

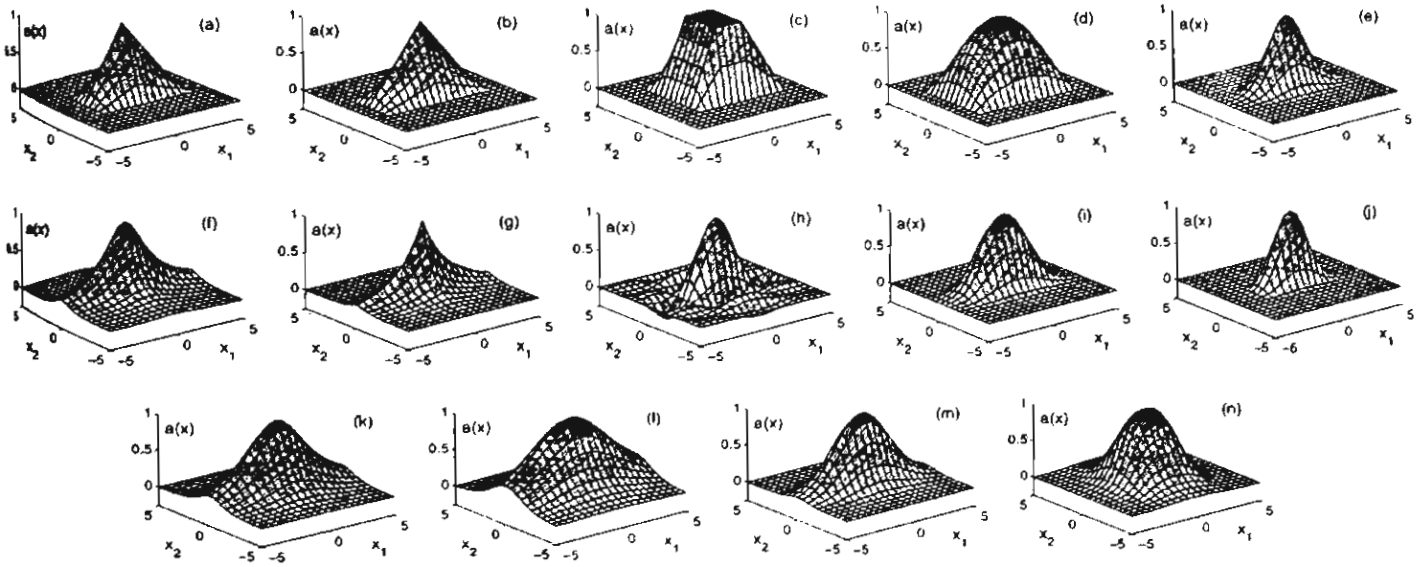

Fig. 6. Product joint set functions centered at $m = 0$. (a) Triangle: $a^1 \sim (0, 3, 4)$ and $a^2 \sim (0, 2, 2)$. (b) Symmetric triangle: $d^1 = 4$ and $d^2 = 2$. (c) Trapezoid: $a^1 \sim (2, -1, 1, 3)$ and $a^2 \sim (1, -1.5, 1.5, 2)$. (d) Parabola: $d^1 = 4$ and $d^2 = 3$. (e) Gaussian: $d^1 = 2$ and $d^2 = 1$. (f) Cauchy: $d^1 = 2$ and $d^2 = 1$. (g) Laplace: $d^1 = 2$ and $d^2 = 1$. (h) Sinc: $d^1 = 0.8$ and $d^2 = 0.4$. (i) Logistic: $d^1 = 2$ and $d^2 = 1$. (j) Hyperbolic tangent: $d^1 = 2$ and $d^2 = 1$. (k) Hyperbolic secant: $d^1 = 2$ and $d^2 = 1$. (l) Differential logistic: $d^1 = 2$ and $d^2 = 1$. (m) Difference logistic: $a^1 = 1$, $l^1 = 2$, $a^2 = 2$, and $l^2 = 1$. (n) Difference hyperbolic tangent: $d^1 = 1$, $l^1 = 2$, $d^2 = 2$, and $l^2 = 1$.

V. Supervised Learning in SAMs: Scalar and Product Sets

Supervised gradient descent can tune all the parameters in the SAM model (6) [12], [14]. A gradient descent learning law for a SAM parameter ξ has the form

$$\xi(t+1) = \xi(t) - \mu_t \frac{\partial E}{\partial \xi} \tag{23}$$

where μ_t is a learning rate at iteration t. We seek to minimize the squared error

$$E(x) = \frac{1}{2}(f(x) - F(x))^2 \tag{24}$$

of the function approximation. The vector function $f : R^n \to R^p$ has components $f(x) = (f_1(x), \ldots, f_p(x))^T$ and so does the vector function F. We consider the case when $p = 1$. A general form for multiple output when $p > 1$ expands the error function $E(x) = \|f(x) - F(x)\|$ for some norm $\|\cdot\|$. Let ξ_j^k denote the kth parameter in the set function a_j. Then the chain rule gives the gradient of the error function with respect to the if-part set parameter ξ_j^k with respect to the then-part set centroid $c_j = (c_j^1, \ldots, c_j^p)^T$ and with respect to the then-part set volume V_j

$$\frac{\partial E}{\partial \xi_k^j} = \frac{\partial E}{\partial F}\frac{\partial F}{\partial a_j}\frac{\partial a_j}{\partial \xi_k^j}, \quad \frac{\partial E}{\partial c_j} = \frac{\partial E}{\partial F}\frac{\partial F}{\partial c_j}, \quad \text{and} \quad \frac{\partial E}{\partial V_j} = \frac{\partial E}{\partial F}\frac{\partial F}{\partial V_j} \tag{25}$$

where

$$\frac{\partial E}{\partial F} = -(f(x) - F(x)) = -\varepsilon(x) \tag{26}$$

$$\frac{\partial F}{\partial a_j} = \frac{(\sum_{i=1}^m a_i(x)V_i)(V_j c_j) - V_j(\sum_{i=1}^m a_i(x)V_i c_i)}{(\sum_{i=1}^m a_i(x)V_i)^2} = \frac{[c_j - F(x)]V_j}{\sum_{i=1}^m a_i(x)V_i} = [c_j - F(x)]\frac{p_j(x)}{a_j(x)}. \tag{27}$$

The SAM ratios (6) with equal rule weights $w_1 = w_2 = \cdots = w_m = w$ give [12], [14]

$$\frac{\partial F}{\partial c_j} = \frac{a_j(x)V_j}{\sum_{i=1}^m a_i(x)V_i} = p_j(x) \tag{28}$$

$$\frac{\partial F}{\partial V_j} = \frac{a_j(x)[c_j - F(x)]}{\sum_{i=1}^m a_i(x)V_i} = [c_j - F(x)]\frac{p_j(x)}{V_j}. \tag{29}$$

Then the learning laws for the then-part set centroids c_j and volumes V_j have the final form

$$c_j(t+1) = c_j(t) + \mu_t \varepsilon(x) p_j(x) \tag{30}$$

$$V_j(t+1) = V_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{V_j}. \tag{31}$$

The learning laws for the if-part set parameters follow in like manner for both scalar and joint sets as we show below.

We first derive learning laws for parameters of the scalar if-part set functions. Each set function a_j gives different partial derivatives of a_j with respect to its kth parameter ξ_k^j in (25). The learning laws for the parameters of each scalar set functions are as follows.

1) Triangle set function

$$m_j(t+1) = \begin{cases} m_j(t) - \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{1}{l_j}, \\ \quad \text{if } m_j - l_j < x < m_j \\ m_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{1}{r_j}, \\ \quad \text{if } m_j < x < m_j + r_j \\ m_j(t), \quad \text{else} \end{cases} \tag{32}$$

$$l_j(t+1) = \begin{cases} l_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{m_j - x}{l_j^2}, \\ \quad \text{if } m_j - l_j < x < m_j \\ l_j(t), \quad \text{else} \end{cases} \tag{33}$$

$$r_j(t+1) = \begin{cases} r_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{x - m_j}{r_j^2}, \\ \quad \text{if } m_j < x < m_j + r_j \\ r_j(t), \quad \text{else.} \end{cases} \tag{34}$$

2) Trapezoid set function

$$ml_j(t+1) = \begin{cases} ml_j(t) - \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{1}{l_j}, \\ \quad \text{if } ml_j - l_j < x < ml_j \\ ml_j(t), \quad \text{else} \end{cases} \tag{35}$$

$$mr_j(t+1) = \begin{cases} mr_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{1}{r_j}, \\ \quad \text{if } mr_j < x < mr_j + r_j \\ mr_j(t), \quad \text{else} \end{cases} \tag{36}$$

$$l_j(t+1) = \begin{cases} l_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{ml_j - x}{l_j^2}, \\ \quad \text{if } ml_j - l_j < x < ml_j \\ l_j(t), \quad \text{else} \end{cases} \tag{37}$$

$$r_j(t+1) = \begin{cases} r_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{x - mr_j}{r_j^2}, \\ \quad \text{if } mr_j < x < mr_j + r_j \\ r_j(t), \quad \text{else.} \end{cases} \tag{38}$$

3) Clipped-parabola set function

$$m_j(t+1) = \begin{cases} m_j(t) + 2\mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{x - m_j}{d_j^2}, \\ \quad \text{if } \left(\frac{x - m_j}{d_j}\right)^2 < 1 \\ m_j(t), \quad \text{else} \end{cases} \tag{39}$$

$$d_j(t+1) = \begin{cases} d_j(t) + 2\mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)}\frac{(x - m_j)^2}{d_j^3}, \\ \quad \text{if } \left(\frac{x - m_j}{d_j}\right)^2 < 1 \\ d_j(t), \quad \text{else.} \end{cases} \tag{40}$$

4) Gaussian set function

$$m_j(t+1) = m_j(t) + 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)]\frac{x - m_j}{d_j^2} \tag{41}$$

$$d_j(t+1) = d_j(t) 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)]\frac{(x - m_j)^2}{d_j^3}. \tag{42}$$

5) Cauchy set function

$$m_j(t+1) = m_j(t) + 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)]\frac{x - m_j}{d_j^2} a_j(x) \tag{43}$$

$$d_j(t+1) = d_j(t) + 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)]\frac{(x - m_j)^2}{d_j^3} \times a_j(x). \tag{44}$$

6) Laplace set function

$$m_j(t+1) = m_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]p_j(x) \times \operatorname{sign}(x - m_j)\frac{1}{|d_j|} \tag{45}$$

$$d_j(t+1) = d_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]p_j(x) \times \operatorname{sign}(d_j)\frac{|x - m_j|}{d_j^2}. \tag{46}$$

7) Sinc set function

$$m_j(t+1) = m_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)} \times \left(a_j(x) - \cos\left(\frac{x - m_j}{d_j}\right)\right)\frac{1}{x - m_j} \tag{47}$$

$$d_j(t+1) = d_j(t) + \mu_t \varepsilon(x)[c_j - F(x)]\frac{p_j(x)}{a_j(x)} \times \left(a_j(x) - \cos\left(\frac{x - m_j}{d_j}\right)\right)\frac{1}{d_j}. \tag{48}$$

8) Logistic set function

$$m_j(t+1) = m_j(t) + \mu_t \varepsilon(x) p_j(x)[c_j - F(x)] \times (2 - a_j(x))\frac{x - m_j}{d_j^2} \tag{49}$$

$$d_j(t+1) = d_j(t) + \mu_t \varepsilon(x) p_j(x)[c_j - F(x)] \times (2 - a_j(x))\frac{(x - m_j)^2}{d_j^3}. \tag{50}$$

9) Hyperbolic tangent set function

$$m_j(t+1) = m_j(t) + 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)] \times (2 - a_j(x))\frac{x - m_j}{d_j^2} \tag{51}$$

$$d_j(t+1) = d_j(t) + 2\mu_t \varepsilon(x) p_j(x)[c_j - F(x)] \times (2 - a_j(x))\frac{(x - m_j)^2}{d_j^3}. \tag{52}$$

10) Hyperbolic secant set function

$$m_j(t+1) = m_j(t) + \mu_t \epsilon(x) p_j(x)[c_j - F(x)] \times \frac{1}{d_j} \tanh\left(\frac{x - m_j}{d_j}\right) \quad (53)$$

$$d_j(t+1) = d_j(t) + \mu_t \epsilon(x) p_j(x)[c_j - F(x)] \times \frac{x - m_j}{(d_j)^2} \tanh\left(\frac{x - m_j}{d_j}\right). \quad (54)$$

11) Differential logistic set function

$$m_j(t+1) = m_j(t) + \mu_t \epsilon(x) p_j(x)[c_j - F(x)] \times \frac{1}{d_j}\left[1 - 2S\left(\frac{x - m_j}{d_j}\right)\right] \quad (55)$$

$$d_j(t+1) = m_j(t) + \mu_t \epsilon(x) p_j(x)[c_j - F(x)] \times \frac{x - m_j}{d_j^2}\left[1 - 2S\left(\frac{x - m_j}{d_j}\right)\right]. \quad (56)$$

12) Difference logistic set function

$$m_j(t+1) = m_j(t) + \mu_t \epsilon(x) p_j(x)[c_j - F(x)] \times \alpha_j[S_j(x - m_j + l_j) + S_j(x - m_j - l_j) - 1] \quad (57)$$

$$\alpha_j(t+1) = \alpha_j(t) + \mu_t \epsilon(x) \frac{p_j(x)}{a_j(x)}[c_j - F(x)] \frac{1}{D_j}[[x - m_j + l_j] \times S_j(x - m_j + l_j)[1 - S_j(x - m_j + l_j)] - [x - m_j - l_j] S_j(x - m_j - l_j)[1 - S_j(x - m_j - l_j)] - l_j a_j(x)(S_j(l_j)[1 - S_j(l_j)] + S_j(-l_j)[1 - S_j(-l_j)])] \quad (58)$$

$$l_j(t+1) = l_j(t) + \mu_t \epsilon(x)[c_j - F(x)] \frac{p_j(x)}{a_j(x)} \frac{\alpha_j}{D_j} \times [S_j(x - m_j + l_j)[1 - S_j(x - m_j + l_j)] + S_j(x - m_j - l_j)[1 - S_j(x - m_j - l_j)] - a_j(x)(S_j(l_j)[1 - S_j(l_j)] + S_j(-l_j)[1 - S_j(-l_j)])]. \quad (59)$$

13) Difference hyperbolic tangent set function

$$m_j(t+1) = m_j(t) + \mu_t \epsilon(x) p_j(x) \frac{c_j - F(x)}{d_j} \times \left[\tanh\left(\frac{x - m_j + l_j}{d_j}\right) + \tanh\left(\frac{x - m_j - l_j}{d_j}\right)\right] \quad (60)$$

$$d_j(t+1) = d_j(t) + \mu_t \epsilon(x) \frac{p_j(x)}{a_j(x)}[c_j - F(x)] \frac{1}{D_j d_j} \times \left[\frac{x - m_j + l_j}{d_j} \tanh^2\left(\frac{x - m_j + l_j}{d_j}\right) - \frac{x - m_j - l_j}{d_j} \tanh^2\left(\frac{x - m_j - l_j}{d_j}\right) - \frac{2l_j}{d_j} + 2\frac{l_j}{d_j} a_j(x)\left[1 - \tanh^2\left(\frac{l_j}{d_j}\right)\right]\right] \quad (61)$$

$$l_j(t+1) = l_j(t) + \mu_t \epsilon(x) \frac{p_j(x)}{a_j(x)}[c_j - F(x)] \frac{1}{D_j d_j} \times \left[2 - \tanh^2\left(\frac{x - m_j + l_j}{d_j}\right) - \tanh^2\left(\frac{x - m_j - l_j}{d_j}\right) - 2a_j(x)\left[1 - \tanh^2\left(\frac{l_j}{d_j}\right)\right]\right]. \quad (62)$$

We also can approximate the learning laws for the symmetric triangle and trapezoid set functions with Gaussian learning laws for their centers and the widths. Like results hold for the learning laws of factorable n-D set functions. A factored set function $a_j(x) = a_j^1(x_1) \cdots a_j^n(x_n)$ leads to a new form for the error gradient. The gradient with respect to the parameter m_j^k of the jth set function a_j has the form

$$\frac{\partial E}{\partial m_j^k} = \frac{\partial E}{\partial F} \frac{\partial F}{\partial a_j} \frac{\partial a_j}{\partial a_j^k} \frac{\partial a_j^k}{\partial m_j^k} \quad (63)$$

where

$$\frac{\partial a_j}{\partial a_j^k} = \prod_{i \neq k}^{n} a_j^i(x_i) = \frac{a_j(x)}{a_j^k(x_k)}. \quad (64)$$

VI. Simulation Results I: Scalar and Joint Product Sets

We trained the SAMs with different set functions to approximate different functions. We scored each test in terms of the squared error (SE) of the function approximation for a constant learning rate μ.

We uniformly sampled 201 points of the function in the one-dimensional (1-D) case to give a training set. The 2-D case used $31 \times 31 = 961$ samples. The 3-D case used $20 \times 20 \times 20 = 8000$ samples. One epoch passed all 201, 961, or 8000 samples through the SAM to train it.

We then finely sampled the function to obtain the test data for each function. So the training data set and the test data set are different but do overlap due to the sampling pattern. The one-input cases used 241 samples, the two-input cases used $51 \times 51 = 2601$ samples and the three-input cases used $31 \times 31 \times 31 = 29\,791$ samples to test how well fuzzy systems approximate the approximands.

The 1-D SAMs used 12 rules, while the 2-D SAMs used 64 rules. The 3-D SAMs used 125 rules. Different initializations led to convergence to different local minima of the SE surface. There is no formal way to find the initial conditions that lead to the global minimum, so we had to guess at them. We spread rule patches uniformly along the input space. So we spread the if-part set centers m_j uniformly along the x-axis. We picked the then-part set centroids c_j as the values of the sampled approximand f at m_j: $c_j = f(m_j)$. We set the then-part volumes (areas) to unity at first: $V_1 = \cdots = V_m = 1$. Then supervised learning tuned each SAM parameter.

We used a constant learning rate μ throughout each training session. We also tried different learning rates to see whether the system converged to different solutions and picked the best results as a representative for that case. But at each try the learning rates for each parameter were the same. The learning rates were

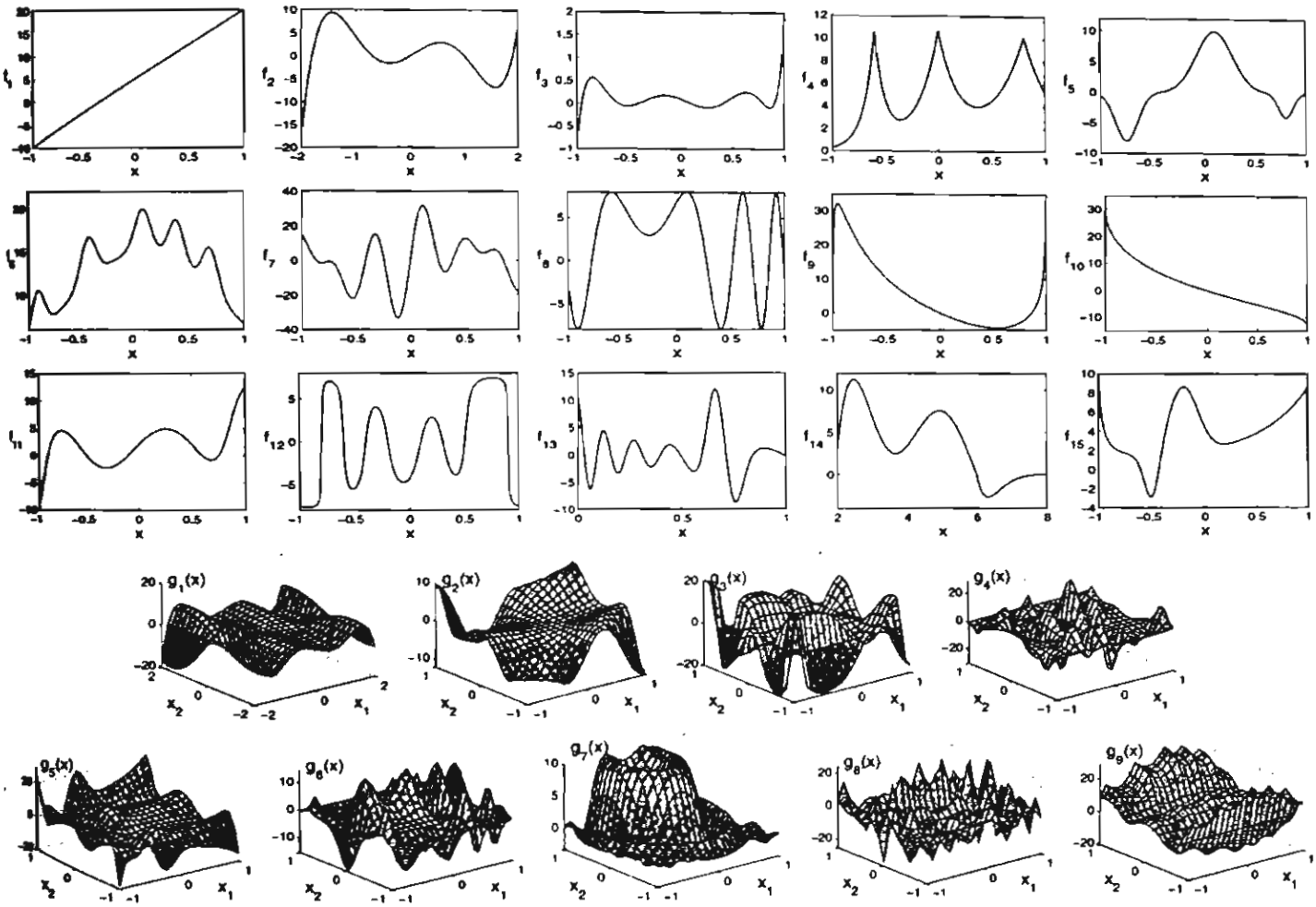

Fig. 7. Samples of 1-D and 2-D test approximands.

small because each learning law is highly nonlinear–else the learning might not have converged. The learning rates that we used ranged from $\mu = 10^{-8}$ to 10^{-4}. We compared the results for each learning rates and picked the best ones. Below we list test functions we used as approximands.

A. 1-D Test Functions

We defined functions of one variable $f : X \subset R \to R$ to test the scalar fuzzy sets in the SAM models. We also used functions from the literature [1], [7]. We roughly classify the test functions that we used and list some of them as follow.

1) Polynomial and Rational Functions: This class of approximands consisted of polynomial functions and rational functions of different degrees. The two simplest functions in this class are a constant function and a straight line function. We do not list constant functions here because we can represent any constant function with any kind of fuzzy system with only one rule. We did include a straight line function in our test case (see Fig. 7). The test functions were as follows:

$$f_1(x) = 15x + 5 \quad \text{for } x \in [-1, 1] \tag{65}$$

$$f_2(x) = 3x(x-1)(x-1.9)(x-0.7)(x+1.8) \quad \text{for } x \in [-2, 2] \tag{66}$$

$$f_3(x) = \frac{\left(\begin{array}{c} 100(x+0.95)(x+0.6)(x+0.4) \\ \times(x-0.1)(x-0.4)(x-0.8)(x-0.9) \end{array}\right)}{(x+1.7)(x-2)^2} \quad \text{for } x \in [-1, 1]. \tag{67}$$

2) Exponential Functions: This class of set function includes Gaussian bell-curve and Laplace functions. The hyperbolic tangent is one form of ratio of exponential functions. We tested the approximands below on the interval $x \in [-1, 1]$

$$f_4(x) = 10\left(e^{-\frac{x}{0.2}} + e^{-\frac{x-0.8}{0.3}} + e^{-\frac{|x+0.6}{0.1}}\right) \tag{68}$$

$$f_5(x) = 10e^{-(\frac{x-0.1}{0.25})^2} - 8e^{-(\frac{x+0.75}{0.15})^2} - 4e^{-(\frac{x-0.8}{0.1})^2} \tag{69}$$

$$f_6(x) = 15e^{-(x-0.1)^2} + 5\left[e^{-(\frac{x-0.1}{0.1})^2} + e^{-(\frac{x-0.4}{0.1})^2} + e^{-(\frac{x-0.7}{0.1})^2} + e^{-(\frac{x+0.3}{0.1})^2} + e^{-(\frac{x+0.9}{0.1})^2}\right]. \tag{70}$$

3) Polynomials Based on Trigonometric Functions: This class of functions includes many functions. A truncated Fourier expansion of any function belongs to this class. We also include the inverse of these trigonometric functions within this class

test cases. All of the functions have as their domain the set $X = [-1, 1]$

$$f_7(x) = 10[\sin(4x + 0.1) + \sin(14x) + \sin(11x - 0.2) + \sin(17x + 0.3)] \quad (71)$$

$$f_8(x) = 8\sin(10x^2 + 5x + 1) \quad (72)$$

$$f_9(x) = 0.01\tan^3(1.5x) + 10\tan^2(x) - 20\tan(0.7x) \quad (73)$$

$$f_{10}(x) = \arccos^3(x) - \arccos^2(-x) - \arccos(-x) \quad (74)$$

$$f_{11}(x) = 10\tan^{-1}[10(x + 0.9)(x + 0.5) \times (x + 0.1)(x - 0.6)(x - 0.75)] \quad (75)$$

$$f_{12}(x) = 5\tan^{-1}\left(\frac{2000(x - .1)(x - .3)(x - .5) \times (x - .9)(x - 1.1)(x + .2) \times (x + .4)(x + .6)(x + .8)(x + 1)}{x^2 + 1.5x + 1}\right). \quad (76)$$

4) Combination of Exponential, Rational, and Trigonometric Functions: We formed a mixed class of functions from the above classes. A sinc function $\sin x/x$ also belongs to this class because it is a rational function of trigonometric and polynomial functions

$$f_{13}(x) = 1 + 10e^{-100(x-0.7)^2}\frac{\sin\left(\frac{125}{x+1.5}\right)}{x + 0.1} \quad \text{for } x \in [0, 1] \quad (77)$$

$$f_{14}(x) = \begin{cases} \frac{10-x}{8}\left(\frac{(x-2.5)^2(x-5)^2(x-9)^3x^3}{400+200(x-0.8)^2} + 12\right) \\ \quad \text{for } 2 \le x < 6 \\ e^{-3(x-6)}\left(\frac{0.005(x-2.5)^2(x-5)^2(x-9)^3x^3}{400+200(x-0.8)^2} + 12\right) \\ \quad \text{for } 6 \le x \le 8 \end{cases} \quad (78)$$

$$f_{15}(x) = \frac{1}{x^3 + 1.1} - 5e^{-(\frac{x+0.5}{0.1})^2} + 7e^{-(\frac{x+0.2}{0.2})^2} + 2e^{-2(x-0.3)} \quad \text{for } x \in [-1, 1]. \quad (79)$$

Fig. 7 plots some of the 1-D approximands.

2-D Test Functions

We created 2-D test functions $f : X \subset R^2 \to R$ from the 1-D test functions. A product of two 1-D functions created 2-D test functions. We also defined new 2-D set functions that were unfactorable. Below we list some samples of the approximands that we tested. All test functions have as their domain the set $X = [-1, 1] \times = [-1, 1] \times [-1, 1]$ except for the test function g_3

$$g_1(x_1, x_2) = 3x_1(x_1 - 1)(x_1 - 1.9)(x_1 + 0.7) \times (x_1 + 1.8)\sin(x_2) \quad \text{for } -2 \le x_1, x_2 \le 2 \quad (80)$$

$$g_2(x_1, x_2) = 5x_1^2x_2 + 2x_1^2 - 3x_2^2 + 6\sin(5x_1x_2^2) \quad (81)$$

$$g_3(x_1, x_2) = 10\tan^{-1}\left(\frac{10(x_1 - 0.2)(x_1 - 0.7)(x_1 + 0.8)}{x_1 + 1.4}\right) \times \tan^{-1}\left(\frac{10(x_2 - 0.2)(x_2 + 0.8)(x_2 - 0.7) \times (x_2 + 0.2)(x_2 - 1.5)}{(x_2 + 1.4)(x_2 - 1.1)x_2(x_2 + 0.3) + 0.7}\right) \quad (82)$$

$$g_4(x_1, x_2) = \frac{1}{10}f_7(x_1)f_5(x_2) \quad (83)$$

$$g_5(x_1, x_2) = \frac{1}{5}f_{15}(x_1)f_{11}(x_2) \quad (84)$$

$$g_6(x_1, x_2) = \frac{1}{5}f_8(x_1)f_5(x_2) \quad (85)$$

$$g_7(x_1, x_2) = 10\frac{\sin(10x_1^2 + 5x_2^2 - 6x_2)}{10x_1^2 + 5x_2^2 - 6x_2} \quad (86)$$

$$g_8(x_1, x_2) = \frac{1}{10}f_7(x_1)f_8(x_2) \quad (87)$$

$$g_9(x_1, x_2) = f_6(x_1)\tan^{-1}(10(x_2 + 0.8) \times (x_2 + 0.3)(x_2 - 0.4)(2x_2 - 0.7)). \quad (88)$$

Fig. 7 plots the surface of some of these samples of 2-D approximands.

C. 3-D Test Functions

We created 3-D test functions $f : X \subset R^3 \to R$ as products of 1-D test functions. We also define new 3-D set functions that were unfactorable. All test functions have as their domain the set $X = [-1, 1] \times [-1, 1] \times [-1, 1]$. Below we list some samples of the approximands that we tested

$$h_1(x_1, x_2, x_3) = 60x_1(x_1 - 0.5)(x_1 - 0.95)(x_1 + 0.35) \times (x_1 + 0.9)\left(3\sin(6x_2x_3) + 6\tan^{-1}(4x_2^2) \times \tan^{-1}(3x_3)\tan^{-1}(2x_2x_3^2) - 5x_2^2x_3\right) \quad (89)$$

$$h_2(x_1, x_2, x_3) = \frac{1}{200}f_{15}(x_1)f_9(x_2)f_{10}(x_3) \quad (90)$$

$$h_3(x_1, x_2, x_3) = \frac{1}{10}f_3(x_1)f_5(x_2)f_{12}(x_3) \quad (91)$$

$$h_4(x_1, x_2, x_3) = \left(1 + 10e^{-100(0.5x_1+0.3)^2}\frac{\sin\left(\frac{125}{0.5x_1+2}\right)}{5x_1 + 6}\right) \times f_6(x_2)f_3(x_3) \quad (92)$$

$$h_5(x_1, x_2, x_3) = e^{-(x_1x_2-0.7)(x_1x_3-0.5)}\frac{\sin(125/(x_1 + 1.5))}{x_2 + 1.1} + (5x_1x_2^2 - 6x_3^3)\tan^{-1}(10x_1x_2 + x_3^2). \quad (93)$$

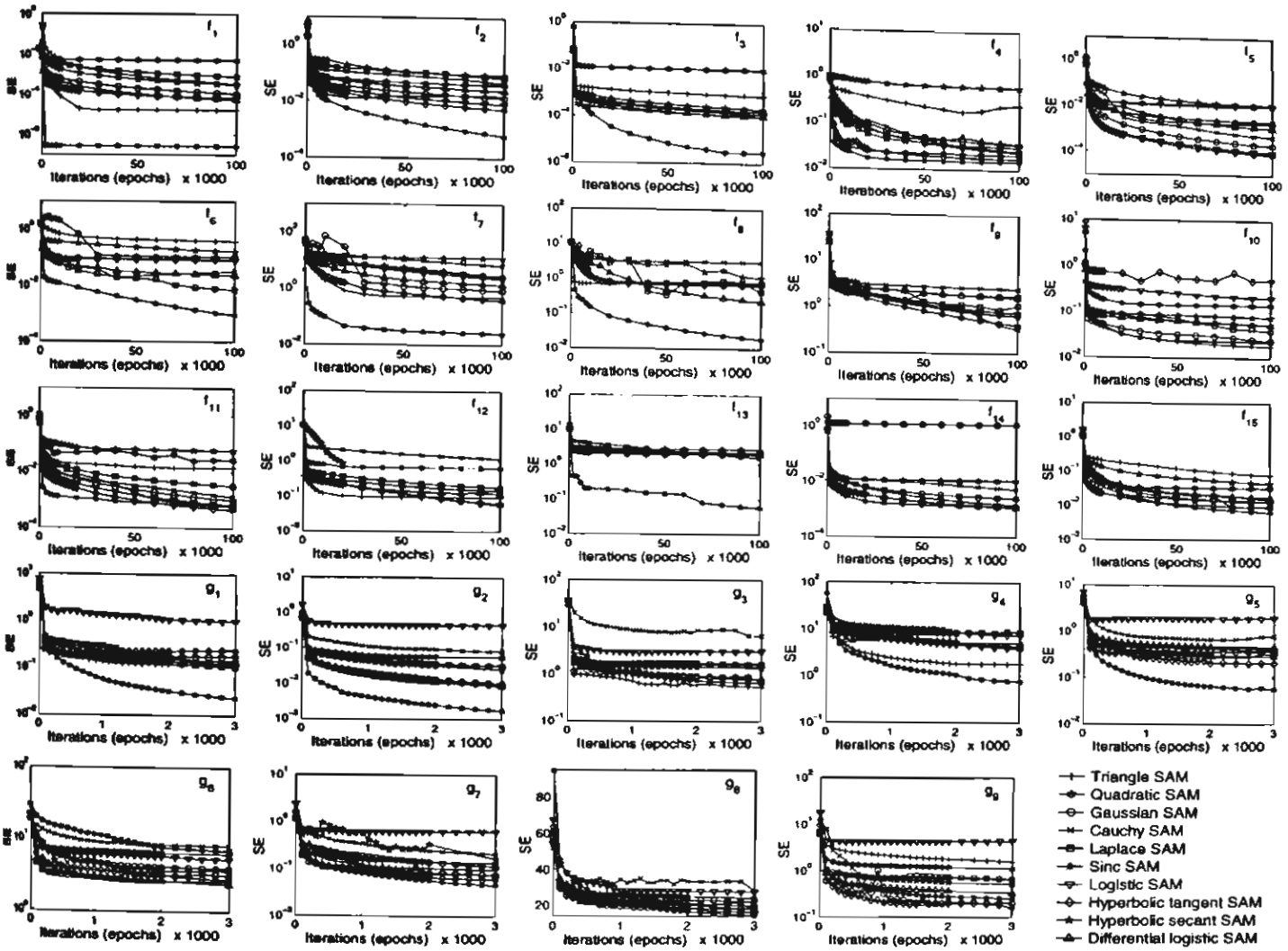

Fig. 8. Convergence plots of squared error versus iteration steps. We picked the best results from different learning rates from each set function. The approximands are 1-D and 2-D approximands in Fig. 7.

D. Results: Comparison of Squared Errors

We gave one point to the set function whose squared error (SE) was the lowest for each test approximand. In case of a tie (when their SEs are well within 20%) we gave a fraction of a point for each tying competitors. We also count as winners the set functions whose SEs lie within 20% of the lowest SE. We tested the learning laws with various learning rates (from $\mu = 10^{-8}$ to $\mu = 10^{-4}$) and also with different initial widths for set functions of bell-curve shape.

Fig. 8 plots the SEs against the number of learning cycles. The simulation results show that the sinc set function often converged faster and more accurately than did the other set functions. The 2-D and 3-D cases with factored set functions showed like patterns. The pie charts in Fig. 9 show the frequency with which each set function performed best in the test cases for the scalar sets and factorable (product) sets. Note that the sinc shape wins in one and two dimensions while it loses to Gaussian and hyperbolic tangent shapes in three dimensions.

A joint set function $a_j : R^n \rightarrow [0, 1]$ measures the degree to which input $x \in R^n$ belong to the fuzzy or multivalued set $A_j \subset R^n : a_j(x) = \text{Degree}(x \in A_j)$. Most fuzzy systems factor the joint set function though some use distance to maintain the joint structure and thus to maintain the correlation among input components [5]. We further examine how factorable and unfactorable joint set functions affect function approximation.

VII. Joint Unfactorable Fuzzy Sets: Transformed Metrics

This section considers a class of joint set functions $a_j : R^n \rightarrow [0, 1]$ that do not factor. We focus on a small class of metrical joint set functions: $a_j[x] = g(d(x; m_j, K_j)) = g(d_j(x))$ for some metric d_j and some scalar function g such as a Gaussian, triangle, or sinc set function.

We first define the metric $d_j(x) = d_j(x; m_j, K_j)$ as a quadratic form with positive definite matrix K_j

$$d_j(x)^2 = (x - m_j)^T K_j (x - m_j). \tag{94}$$

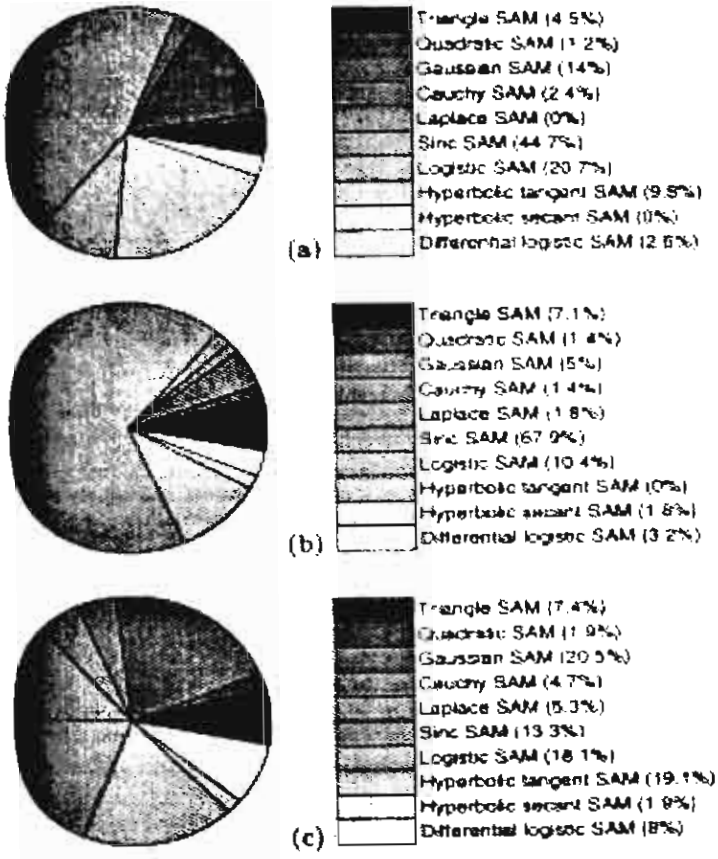

Fig. 9. Proportions of test cases where each function performed best. Multidimensional sets are factorable (product) sets of the scalar ones. The winners in each case are from the best learning rates from $\mu = 10^{-6}$ to $\mu = 10^{-4}$. (a) 1-D, (b) 2-D, and (c) 3-D test cases.

Then we can create metrical joint set functions a_j from this metric d_j and the scalar set functions g: $a_j[x] = g(d_j(x))$. Below we show the cases when g takes the form of a piecewise linear function $g(x) = ax + b$ (this gives a metrical triangle), parabolic function $g(x) = ax^2 + bx + c$, Cauchy function $g(x) = 1/(1 + x^2)$, Gaussian function $g(x) = e^{-x^2}$, Laplace function $g(x) = e^{-|x|}$, sinc function $g(x) = \sin x/x$, hyperbolic tangent $g(x) = 1 + \tanh(-x^2)$, logistic function $g(x) = 2S(-x^2)$ where $S(x) = 1/(1 + e^{-x})$, hyperbolic secant $g(x) = \operatorname{sech} x$, or the derivative of logistic function $g(x) = S'(x)$.

1) *Symmetric metrical triangle set function.* This set function defines the degree to which an input vector $x \in R^n$ belongs to set A_j with linear function

$$a_j[x] \equiv a_j(d_j(x)) = \begin{cases} 1 - d_j(x), & \text{if } d_j(x) < 1 \\ 0, & \text{else} \end{cases} \tag{95}$$

2) *Joint Gaussian set function.* This set function derives from the probability density function of a jointly normal random vector [23]

$$a_j[x] = e^{-d_j(x)^2} = e^{-(x-m_j)^T K_j (x-m_j)}. \tag{96}$$

So K_j is analogous to the inverse covariance matrix $(1/2)K^{-1}$ and m_j is analogous to the mean vector in the normalized joint Gaussian probability density [23]. The joint Gaussian set factors when the positive definite matrix K_j is diagonal.

The joint Gaussian set function has the Mahalanobis distance as its exponent if K_j^{-1} is a covariance matrix. We apply this method to scalar set functions to create metrical joint set functions below.

3) *Metrical parabolic set function.* The set value linearly falls as the square of the distance d_j grows

$$a_j[x] = \begin{cases} 1 - d_j(x)^2, & \text{if } d_j(x) < 1 \\ 0, & \text{else.} \end{cases} \tag{97}$$

4) *Joint Cauchy.* The joint Cauchy set function derives from the probability density function of joint Cauchy random variables [25]. We discard the constant that normalizes the density function to a unit integral and obtain the joint Cauchy set function

$$a_j[x] = \frac{1}{[1 + (x - m_j)^T K_j (x - m_j)]^{(n+1)/2}}. \tag{98}$$

5) *Metrical Cauchy set function.* This set function differs from the actual joint Cauchy density in (98). It has a simpler form

$$a_j[x] = \frac{1}{1 + d_j(x)^2} = \frac{1}{1 + (x - m_j)^T K_j (x - m_j)}. \tag{99}$$

6) *Metrical Laplace set function.* The scalar Laplace function forms the metrical set function as

$$a_j[x] = \exp\{-d_j(x)\} = e^{-\sqrt{(x-m_j)^T K_j (x-m_j)}}. \tag{100}$$

This metrical set function reduces to the factorable product set if the positive definite matrix K_j is diagonal.

7) *Metrical sinc set function.* The scalar sinc function forms a joint metrical set from a metric d_j as

$$a_j[x] = \frac{\sin(d_j(x))}{d_j(x)}. \tag{101}$$

8) *Metrical logistic set function.* The logistic function defines this metrical joint set as

$$a_j[x] = \frac{2}{1 + \exp\{d_j(x)^2\}}. \tag{102}$$

9) *Metrical hyperbolic tangent set function.* This metrical joint set has the form

$$a_j[x] = 1 + \tanh(-d_j(x)^2). \tag{103}$$

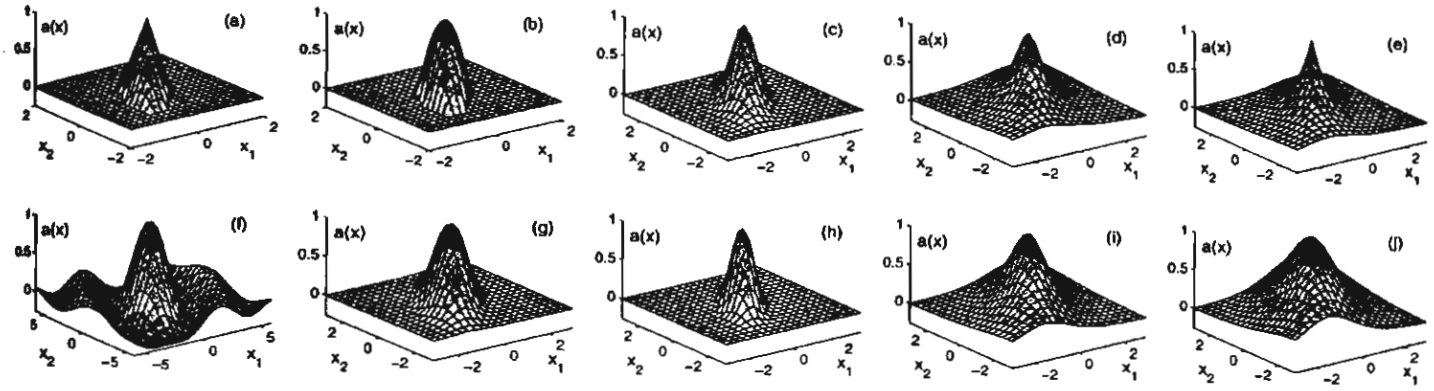

Fig. 10. Metrical joint set functions with $m = 0$ and $K = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$ and l^2 distance. (a) Symmetric metrical triangle set function. (b) Metrical parabola set function. (c) Joint (metrical) Gaussian set function. (d) Metrical Cauchy set function. (e) Metrical Laplace set function. (f) Metrical sinc set function. (g) Metrical logistic set function. (h) Metrical hyperbolic tangent set function. (i) Metrical hyperbolic secant set function. (j) Metrical differential logistic set function.

10) *Metrical hyperbolic secant set function.* We form the metrical joint set from the hyperbolic secant function as

$$a_j[x] = \operatorname{sech}(d_j(x)). \tag{104}$$

11) *Metrical differential logistic set function.* The derivative of the [illegible] function also defines a metrical joint set

$$a_j[x] = 4S(d_j(x))(1 - S(d_j(x))). \tag{105}$$

Fig. 10 shows some of the above joint set functions with centers at $m_j = 0$ and with $K_j = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$.

The metric d_j reduces to the weighted l^2 metric for the diagonal matrix $K_j = \operatorname{diag}((\kappa_j^1)^2, \ldots, (\kappa_j^n)^2)$ $d_j(x) = \sqrt{\sum_{i=1}^n |\kappa_j^i(x_i - m_j^i)|^2}$. So we can generalize this metrical measure to the weighted l^p metric

$$d_j^p(x) = \left[\sum_{i=1}^n \left|\kappa_j^i \left(x_i - m_j^i\right)\right|^p\right]^{\frac{1}{p}} \tag{106}$$

for $p > 0$ and use it to create joint metrical set functions. We replaced the weights κ_j^i from the diagonal matrix K with scales $1/\sigma_j^i$. So we replaced $|\kappa_j^i(x_i - m_j^i)|^p$ with $|(x_i - m_j^i)/\sigma_j^i|^p$ to conform with the form of factorable sets in Section IV-B. The p metrical distance has the form

$$d_j^p(x) = \left[\sum_{i=1}^n \left|\frac{x_i - m_j^i}{\sigma_j^i}\right|^p\right]^{\frac{1}{p}}. \tag{107}$$

So the l^p metrical set function a_j^p follows as

$$a_j^p(x) = g\left(d_j^p(x)\right) \tag{108}$$

for some scalar function $g : R \to R$ and for d_j^p as in (107). This gives a general form for l^p metrical sets. The real function $g : R \to R$ can be any generalized scalar set function. Popular examples of g are triangle and Gaussian functions.

We also tested the metrical sets with the l^1 or "city block" metric

$$d_j^1(x) = \sum_{i=1}^n \left|\frac{x_i - m_j^i}{\sigma_j^i}\right| \tag{109}$$

where $\sigma_j^i > 0$ in (107). The l^1 set function a_j^1 has the form

$$a_j^1[x] = g\left(d_j^1(x)\right) = g\left(\sum_{i=1}^n \left|\frac{x_i - m_j^i}{\sigma_j^i}\right|\right). \tag{110}$$

Fig. 11 shows some of the l^1 metrical sets with $m = 0$ and $\sigma = [2\ 1]$ for the 2-D input case. The function g takes the form of a symmetrical triangle, parabola, Gaussian, Cauchy, Laplace, sinc, logistic, hyperbolic tangent, or differential logistic function.

We now consider the extreme case of the l^p metrical set functions when $p = \infty$. This gives the "max" metric. The l^∞ set function has the form

$$a_j^\infty[x] = g\left(d_j^\infty(x)\right) \tag{111}$$

$$= g\left(\lim_{p\to\infty}\left(\sum_{i=1}^n \left|\frac{x_i - m_j^i}{\sigma_j^i}\right|^p\right)^{\frac{1}{p}}\right) \tag{112}$$

$$= g\left(\max_{1\le i\le n}\left|\frac{x_i - m_j^i}{\sigma_j^i}\right|\right). \tag{113}$$

Note that $|(x_i - m_j^i)/\sigma_j^i|$ is never negative. So if $g(x)$ is monotone decreasing for $x \geq 0$ (such as for a triangle or Gaussian function or any unimodal function where a_j peaks at $x = m_j$) then

$$a_j^\infty[x] = g\left(\max_{1\le i\le n}\left|\frac{x_i - m_j^i}{\sigma_j^i}\right|\right) \tag{114}$$

$$= \min_{1\le i\le n} g\left(\left|\frac{x_i - m_j^i}{\sigma_j^i}\right|\right) \tag{115}$$

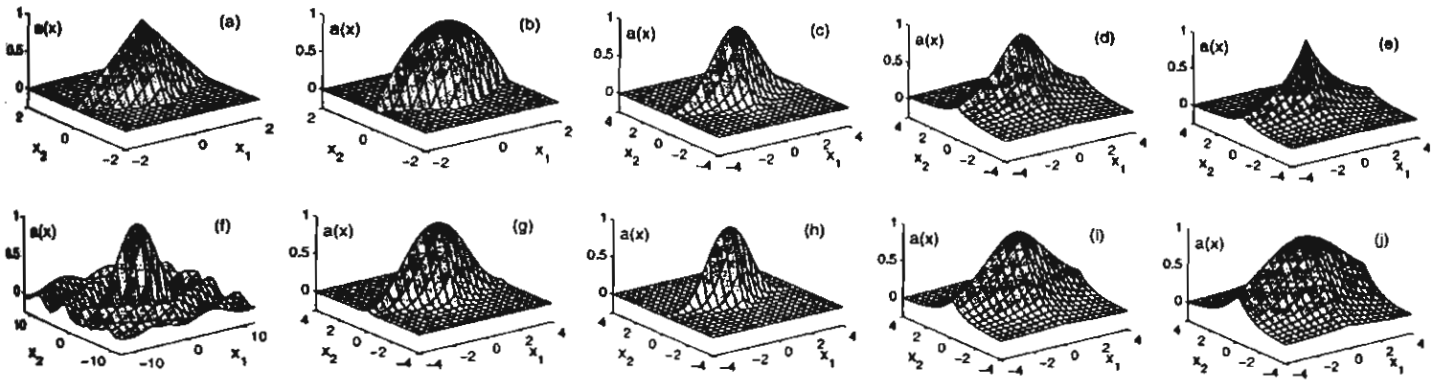

Fig. 11. Metrical joint set functions with $m = 0$, $\sigma = [2\ 1]$, and l^1 distance. (a) Symmetric metrical triangle set function. (b) Metrical parabola set function. (c) Metrical Gaussian set function. (d) Metrical Cauchy set function. (e) Metrical Laplace set function. (f) Metrical sinc set function. (g) Metrical logistic set function. (h) Metrical hyperbolic tangent set function. (i) Metrical hyperbolic secant set function. (j) Metrical differential logistic set function.

$$= \min_{1 \le i \le n} a_{ji} \left(\left| \frac{x_i - m_j^i}{\sigma_j^i} \right| \right) \tag{116}$$

holds for a scalar set function $a_{ji}(x_i) = g(|(x_i - m_j^i)/\sigma_j^i|)$. So the l^∞ metrical joint sets a_j^∞ with g monotone decreasing are equivalent to the factorable sets with the min conjunctive operator. Fig. 12 shows the sets of points that give the same distance from the origin with l^p metric for $p = 1, 2$, and ∞. So factorable set functions with min bound the metrical set functions in (108) through the l^p metric in (107).

The shape and orientation of the "hills" of if-part fuzzy sets may help fuzzy systems better approximate certain functions in that region. So we transform the translated input vector $(x - m_j) \in R^n$ to $A_j(x - m_j)$ where $A_j : R^n \to R^n$ is any linear or nonlinear operator [16]. We transform the translated vector $x - m_j$ instead of the input vector x because it is easier to keep track of the "center" vector m_j (if we use a unimodal set function such as the Gaussian and some mapping A_j such that $A_j(x) = 0$ if and only if $x = 0$).

Here we show the simple case of a linear transformation. Say A_j is an $n \times n$ matrix $A_j \in R^{n \times n}$. Then define the norm (or distance with the vector m_j) as

$$d_j^p(x) = \|A_j(x - m_j)\|_p \tag{117}$$

for the jth metrical set function $a_j[x] = g(d_j^p(x))$ as above. The p-norm $\|x\|_p$ of a vector $x = [x_1, \ldots, x_n]^T \in R^n$ has the form

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}. \tag{118}$$

So we can rewrite the quadratic distance $d_j^2(x) = \sqrt{(x - m_j)^T K_j (x - m_j)}$ in (94) in the form of (117). The matrix K_j is symmetrical nonnegative definite: $K_j = K_j^T \ge 0$. So $K_j = \sqrt{K_j}\sqrt{K_j}$ and $\sqrt{K_j} = (\sqrt{K_j})^T$ [29]. This implies that

$$d_j^2(x) = \sqrt{(x - m_j)^T K_j (x - m_j)} \tag{119}$$

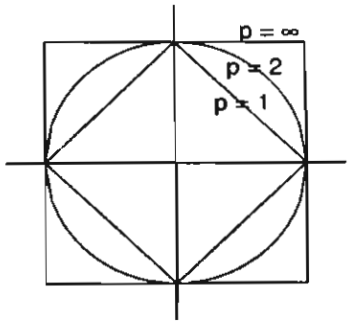

Fig. 12. Spheres in different metric spaces.

$$= \sqrt{(x - m_j)^T \sqrt{K_j^T}\sqrt{K_j}(x - m_j)} \tag{120}$$

$$= \|\sqrt{K_j}(x - m_j)\|_2. \tag{121}$$

This has the form $\|A_j(x - m_j)\|_p$ where $A_j = \sqrt{K_j}$ and $p = 2$.

Users may encode more useful information in the nonlinear operator A_j to reduce the number of fuzzy rules and perhaps lessen the rule explosion. Finding good combinations of nonlinear maps A_j and metrics d_j^p and functional form g remains an open research problem.

VIII. Supervised Learning in SAMs: Metrical Sets

The learning laws for the then-part set centroids c_j and volumes V_j remain the same for any if-part fuzzy sets. Only the learning laws for if-part set parameters have new forms. The joint metrical set functions depend on the metric d_j. So we tune the parameters that define the metric d_j. For the quadratic metric $d_j(x)^2 = (x - m_j)^T K_j (x - m_j)$ we tune the vector m_j and the matrix K_j

$$m_j(t+1) = m_j(t) - \mu_t \nabla_{m_j} E \tag{122}$$

$$K_j(t+1) = K_j(t) - \mu_t \nabla_{K_j} E. \tag{123}$$

The partial derivatives (or gradients in the vector-matrix cases) follow from (24) in like manner

$$\nabla_{m_j} E = \frac{\partial E}{\partial F}\frac{\partial F}{\partial a_j}\frac{\partial a_j}{\partial d_j}\nabla_{m_j} d_j \tag{124}$$

$$\nabla_{K_j} E = \frac{\partial E}{\partial F}\frac{\partial F}{\partial a_j}\frac{\partial a_j}{\partial d_j}\nabla_{K_j} d_j. \tag{125}$$

We have derived the first two partial derivatives in (26) and (27). The partial derivative $(\partial a_j)/(\partial d_j)$ depends on which scalar set function we use to create the joint set function.

1) Symmetric metrical triangle set function

$$\frac{\partial a_j}{\partial d_j} = \begin{cases} -1, & \text{if } d_j(x) < 1 \\ 0, & \text{else,} \end{cases} \tag{126}$$

2) Metrical parabola set function

$$\frac{\partial a_j}{\partial d_j} = \begin{cases} -2d_j(x), & \text{if } d_j(x) < 1 \\ 0, & \text{else.} \end{cases} \tag{127}$$

3) Joint Gaussian set function

$$\frac{\partial a_j}{\partial d_j} = -2d_j(x)a_j[x]. \tag{128}$$

4) Joint Cauchy set function

$$\frac{\partial a_j}{\partial d_j} = -(n+1)\frac{d_j(x)}{1+d_j(x)^2}a_j[x]. \tag{129}$$

5) Metrical Cauchy set function

$$\frac{\partial a_j}{\partial d_j} = -2d_j(x)a_j[x]^2. \tag{130}$$

6) Metrical Laplace set function

$$\frac{\partial a_j}{\partial d_j} = -a_j[x]. \tag{131}$$

7) Metrical sinc set function

$$\frac{\partial a_j}{\partial d_j} = \frac{1}{d_j(x)}(\cos(d_j(x)) - a_j[x]). \tag{132}$$

8) Metrical logistic set function

$$\frac{\partial a_j}{\partial d_j} = -d_j(x)(2 - a_j[x])a_j[x]. \tag{133}$$

9) Metrical hyperbolic tangent set function

$$\frac{\partial a_j}{\partial d_j} = -2d_j(x)(2 - a_j[x])a_j[x]. \tag{134}$$

10) Metrical hyperbolic secant set function

$$\frac{\partial a_j}{\partial d_j} = -\tanh(d_j(x))a_j[x]. \tag{135}$$

11) Metrical differential logistic set function

$$\frac{\partial a_j}{\partial d_j} = -S(d_j(x))a_j[x]. \tag{136}$$

These partial derivatives $(\partial a_j)/(\partial d_j)$ hold for any metric d_j that users might choose. They are independent of the function d_j that we use to transform the input vector x into the scalar $d_j(x)$.

We now derive the gradients of the metric d_j with respect to the vector m_j and matrix K_j for the quadratic case $d_j(x)^2 = (x - m_j)^T K_j (x - m_j)$. The gradients have the form

$$\nabla_{m_j} d_j = -\frac{1}{d_j(x)}K_j(x - m_j) \tag{137}$$

$$\nabla_{K_j} d_j = \frac{1}{2d_j(x)}(x - m_j)(x - m_j)^T \tag{138}$$

since $K_j = K_j^T$.

We might use diagonal matrices K_j to reduce the computation. This reduces the quadratic form of d_j to a weighted l^2 norm. We can also use any l^p norm to compute $d_j^p(x)$ as mentioned earlier. We also examine set functions from the l^1 norm as in (109). The partial derivatives have the form

$$\frac{\partial d_j^1}{\partial m_j^k} = -\operatorname{sgn}(x_k - m_j^k)\frac{1}{|\sigma_j^k|} \tag{139}$$

$$\frac{\partial d_j^1}{\partial \sigma_j^k} = -\frac{|x_k - m_j^k|}{(\sigma_j^k)^2} \tag{140}$$

for $\sigma_j^k > 0$. The learning laws for the set functions that use the l^p metric in (106) follow in like manner. We now derive the learning laws for the metrical set function $a_j[x] = g(d_j^p(x))$ where d_j^p takes the form in (117) and A_j is a matrix $A_j \in R^{n \times n}$. Let $[A_j]_i$ denote the ith row of an $n \times n$ matrix A_j, and put $m_j = [m_j^1, \ldots, m_j^n]^T$. We can rewrite the norm $d_j^p(x)$ as

$$d_j^p(x) = \|A_j(x - m_j)\|_p \tag{141}$$

$$= \left(\sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p\right)^{\frac{1}{p}}. \tag{142}$$

So the gradient (in row vector notation) for the kth row of A_j is

$$\nabla_{[A_j]_k} d_j^p = \frac{1}{p}\left(\sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p\right)^{\frac{1}{p}-1} \times \nabla_{[A_j]_k} \sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p \tag{143}$$

$$= \frac{1}{p}\left(d_j^p(x)\right)^{-1} \nabla_{[A_j]_k} |[A_j]_k(x - m_j)|^p \quad (144)$$

$$= \frac{1}{d_j^p(x)} |[A_j]_k(x - m_j)|^{p-1} \operatorname{sgn}([A_j]_k(x - m_j)) \times \nabla_{[A_j]_k}([A_j]_k(x - m_j)) \quad (145)$$

$$= \frac{1}{d_j^p(x)} |[A_j]_k(x - m_j)|^{p-1} \times \operatorname{sgn}([A_j]_k(x - m_j))(x - m_j)^T. \quad (146)$$

The gradient of the metric d_j^p with respect to m_j (in column vector notation) follows in like manner

$$\nabla_{m_j} d_j^p = \frac{1}{p}\left(\sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p\right)^{\frac{1}{p}-1} \times \nabla_{m_j} \sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p \quad (147)$$

$$= \frac{1}{p}\left(d_j^p(x)\right)^{-1} \sum_{i=1}^{n} \nabla_{m_j} |[A_j]_i(x - m_j)|^p \quad (148)$$

$$= \frac{p}{d_j^p(x)} \sum_{i=1}^{n} p|[A_j]_i(x - m_j)|^{p-1} \times \nabla_{m_j} |[A_j]_i(x - m_j)| \quad (149)$$

$$= \frac{1}{d_j^p(x)} \sum_{i=1}^{n} |[A_j]_i(x - m_j)|^{p-1} \times \operatorname{sgn}([A_j]_i(x - m_j))\left(-[A_j]_i^T\right) \quad (150)$$

$$= -\frac{1}{d_j^p(x)} \sum_{i=1}^{n} |[A_j]_i(x - m_j)|^{p-1} \times \operatorname{sgn}([A_j]_i(x - m_j))[A_j]_i^T. \quad (151)$$

We can further tune the parameter p in the l^p metric in (106)

$$\frac{\partial d_j^p}{\partial p} = -\frac{1}{p} d_j^p(x) \ln d_j^p(x) + \frac{1}{p}\left(\sum_{i=1}^{n} \left|\kappa_j^i\left(x_i - m_j^i\right)\right|^p\right)^{\frac{1}{p}-1} \times \left(\sum_{i=1}^{n} \left|\kappa_j^i\left(x_i - m_j^i\right)\right|^p \ln\left|\kappa_j^i\left(x_i - m_j^i\right)\right|\right) \quad (152)$$

$$= -\frac{1}{p} d_j^p(x) \ln d_j^p(x) + \frac{1}{p} \frac{1}{\left(d_j^p(x)\right)^{p-1}} \times \left(\sum_{i=1}^{n} \left|\kappa_j^i\left(x_i - m_j^i\right)\right|^p \ln\left|\kappa_j^i\left(x_i - m_j^i\right)\right|\right). \quad (153)$$

The partial derivative when the metric d_j^p has the form (117) has the similar form

$$\frac{\partial d_j^p}{\partial p} = -\frac{1}{p} d_j^p(x) \ln d_j^p(x) + \frac{1}{p} \frac{1}{\left(d_j^p(x)\right)^{p-1}} \times \left(\sum_{i=1}^{n} |[A_j]_i(x - m_j)|^p \ln |[A_j]_i(x - m_j)|\right). \quad (154)$$

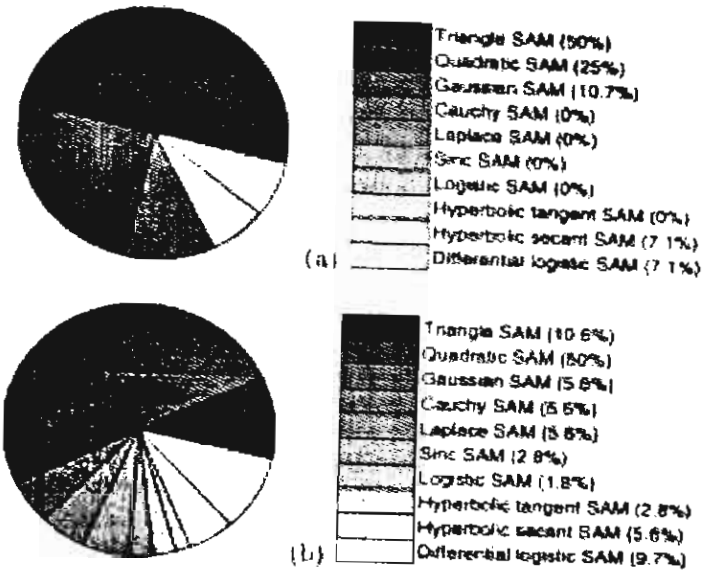

Fig. 13. l^2 metrical sets. Proportions of test cases where each metrical set function performed best. (a) 2-D test cases. (b) 3-D test cases. Note that the metrical triangle and the metrical quadratic switch from first and second place for the 2-D test cases to second and first place for the 3-D test cases.

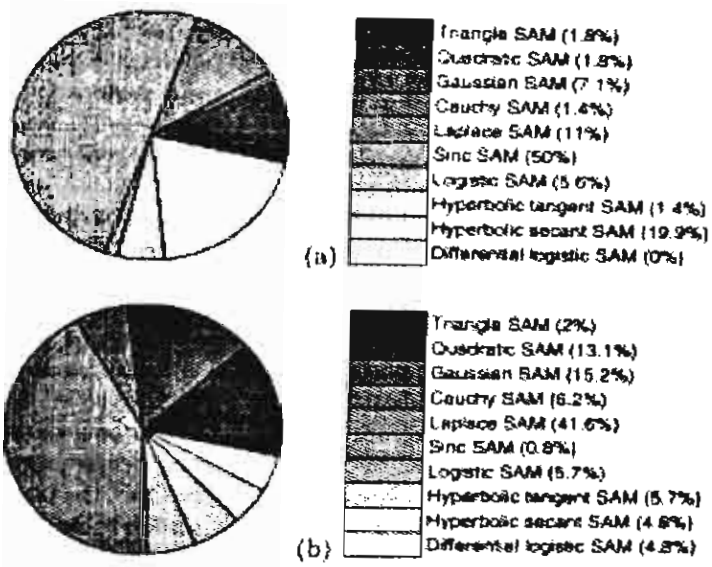

Fig. 14. l^1 metrical sets. Proportions of test cases where each metrical set function performed best. (a) 2-D test cases. (b) 3-D test cases. The l^1-metrical sinc goes from winner for the 2-D test cases to loser for the 3-D test cases. The l^1-metrical Laplacian emerges as the winner for the first time in the 3-D case.

IX. Simulation Results II: Joint Metrical Sets

Figs. 13 and 14 show the second results of quadratic l^2 and l^1 metrical sets in 2-D and 3-D test cases. Fig. 13 shows that the metrical triangle performs best in the 2-D experiments while the l^2-metrical quadratic performs second best. This outcome reverses in the 3-D experiments. There the l^2-metrical quadratic if-part set performs best while the metrical triangle performs second best. Fig. 14 shows that the l^1-metrical sinc wins for the 2-D test cases but loses for the 3-D test cases (when the l^1-metrical Laplace wins).

A. The Second Curse of Dimensionality and Unfactorable Metrical Sets

Our final result is negative: even unfactorable joint set functions can suffer the second curse of dimensionality of spikiness in high dimensions. The following theorem illustrates this claim for metrical set functions that depend on diagonal matrices. The result may also hold for many nondiagonal matrices.

Theorem: Suppose that a metrical set function a_j^p has the form

$$a_j^p(x) = g\left(d_j^p(x)\right) \tag{155}$$

for the l^p metric $d_j^p(x) = \|A_j(x - m_j)\|_p$. Here A_j is an $n \times n$ positive-definite diagonal matrix and $g : R_+ \to [0, 1]$ is a monotone decreasing function such that $g(x) \to 0$ as $x \to \infty$. Then a_j^p suffers the second curse of dimensionality: it collapses to a spike in high dimension as n grows to ∞.

Proof: Recall that factorable set functions with min conjunction $a_j(x) = \min_i g(|(x_i - m_j^i)/d_j^i|)$ collapse to spikes in high dimensions for monotone decreasing g such that $g(x) \to 0$ as $x \to \infty$ (see Section II). So we need show only that for a given metrical set function a_j^p in (155) there exists a *factorable* set function $\tilde{a}_j$ (generated from the same function g) that bounds a_j: $a_j(x) \leq \tilde{a}_j(x)$. Then the metrical set a_j^p collapses to a spiky surface in high dimensions.

For a matrix A_j it follows that

$$\begin{aligned}
a_j^p(x) &= g(\|A_j(x - m_j)\|_p) && (156)\\
&= g(\|A_j x - A_j m_j\|_p) && (157)\\
&= g(\|\tilde{x} - \tilde{m}_j\|_p) && (158)\\
&\leq g(\|\tilde{x} - \tilde{m}_j\|_\infty) = a_j^\infty(x) \\
&\quad \text{since } \|x\|_p \geq \|x\|_\infty \text{ (see Lemma below)} \\
&\quad \text{and } g : R_+ \to [0, 1] \text{ is monotone decreasing} && (159)\\
&= g\left(\max_i |\tilde{x}_i - \tilde{m}_j^i|\right) = \min_i g\left(|\tilde{x}_i - \tilde{m}_j^i|\right) \\
&\quad \text{since } g \text{ is monotone decreasing} && (160)\\
&= \min_i \tilde{a}_j^i(x) = \tilde{a}_j(x) && (161)
\end{aligned}$$

where $\tilde{a}_j^i(x) = g(|[A_j]_i x - [A_j]_i m_j|)$ and $[A_j]_i$ is the ith row of A_j. So $\min_i \tilde{a}_j^i(x)$ bounds $a_j^p(x)$. Q.E.D.

Lemma: $\|x\|_p \geq \|x\|_\infty$ if $x = [x_1, \ldots, x_n] \in R^n$.

Proof: Consider $x \in R^n$. Then

$$\sum_{i=1}^n |x_i|^p \geq |x_j|^p \quad \text{for all } j = 1, \ldots, n \tag{162}$$

$$\Leftrightarrow \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} \geq |x_j| \quad \text{for all } j = 1, \ldots, n \tag{163}$$

$$\Leftrightarrow \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} \geq \max_j |x_j| = \lim_{r \to \infty} \left(\sum_{i=1}^n |x_i|^r\right)^{1/r}. \tag{164}$$

So $\|x\|_p \geq \|x\|_\infty$. Q.E.D.

Note that the set function $\tilde{a}_j$ may not count as a factorable set function since each component $\tilde{a}_j^i$ takes as input the whole vector $x \in R^n$. Then the ith row of A_j transforms the input vector x into a scalar $\tilde{x}_i$. Therefore $\{\tilde{x}_i\}_{i=1}^n$ may not be independent and so the theorem (Borel–Cantelli lemma) [9] need not apply. The theorem does apply if A_j is diagonal.

X. Conclusion: The Search Goes On

At least three main conclusions follow from the above if-part fuzzy-set definitions, learning laws, and simulations of how these if-part sets affect adaptive fuzzy function approximation. The first conclusion is that curses of dimensionality alone will impose tight limits on empirical searches for the best shape of parametrized if-part fuzzy sets. The complexity of the learning laws further compounds this computational burden. It limited our simulation experiments to no more than three dimensions. The sets that performed well in these smaller dimensions may not do so in higher dimensions. The winner histograms even changed dramatically when going from one to two to three dimensions. The second dimensionality curse of set spikiness will also have greater force for searches through the spaces of four- and higher dimensional set functions.

The second conclusion is that common sense or even expert intuition may offer little guidance for picking good if-part sets in higher dimensions. Indeed, they may mislead even in the scalar case. The frequent winning status of the sinc set in the simulations shows that. This seems to be the first time anyone has used the sinc function as a fuzzy set and yet such sets may well have improved the performance of many real fuzzy systems. Surely there are many more scalar if-part sets that would perform even better for these and other test functions and that would appear even less intuitive or have less linguistic meaning than does the sinc function. Again, the engineering goal of accurate function approximation will tend to lead the search for the best if-part set far beyond where the earlier goal of accurate linguistic modeling would take it. And the success of the sinc set and the hyperbolic-tangent bell curve further suggest that the familiar Gaussian or Cauchy or other familiar unimodal curves will not emerge as optimal set functions in other searches.

The third conclusion follows from the other two: The search for the best shape of if-part (and then-part) sets will continue. There are as many continuous if-part fuzzy subsets of the real line as there are real numbers. The set of all if-part fuzzy subsets of the real line has the higher cardinality of the set of all subsets of the real line. Fuzzy theorists will never exhaust this search space. Each theorist can draw different lines through the space to form set taxonomies or to focus the search or to pose narrow or broad optimality problems. We suspect that many such searches will take care to distinguish factorable from unfactorable sets though they may well ignore our distinction of parametrized versus nonparametrized sets. The unfactorable sets hold the promise that they may lessen if not defeat exponential rule explosion even if they may still suffer from set spikiness. These searches may be endless in principle but that itself does not mean that they are not worthwhile. They can on occasion produce new tools.

APPENDIX
THE STANDARD ADDITIVE MODEL (SAM) THEOREM

This Appendix derives the basic ratio structure (6) of a standard additive model fuzzy system and review the local structure of optimal fuzzy rules.

SAM Theorem: Suppose the fuzzy system $F : R^n \to R^p$ is a standard additive model: $F(x) = \text{Centroid}(B(x)) = \text{Centroid}(\sum_{j=1}^m w_j a_j(x) B_j)$ for if-part joint set function $a_j : R^n \to [0, 1]$, rule weights $w_j \geq 0$, and then-part fuzzy set $B_j \subset R^p$. Then $F(x)$ is a convex sum of the m then-part set centroids

$$F(x) = \frac{\sum_{j=1}^m w_j a_j(x) V_j c_j}{\sum_{j=1}^m w_j a_j(x) V_j} = \sum_{j=1}^m p_j(x) c_j. \tag{165}$$

The convex coefficients or discrete probability weights $p_1(x), \ldots, p_m(x)$ depend on the input x through

$$p_j(x) = \frac{w_j a_j(x) V_j}{\sum_{i=1}^m w_i a_i(x) V_i}. \tag{166}$$

V_j is the finite positive volume (or area if $p = 1$) and c_j is the centroid of then-part set B_j

$$V_j = \int_{R^p} b_j(y_1, \ldots, y_p)\, dy_1 \ldots dy_p > 0 \tag{167}$$

$$c_j = \frac{\int_{R^p} y b_j(y_1, \ldots, y_p)\, dy_1 \ldots dy_p}{\int_{R^p} b_j(y_1, \ldots, y_p)\, dy_1 \ldots dy_p}. \tag{168}$$

Proof: There is no loss of generality to prove the theorem for the scalar-output case $p = 1$ when $F : R^n \to R^p$. This simplifies the notation. We need but replace the scalar integrals over R with the p-multiple or volume integrals over R^p in the proof to prove the general case. The scalar case $p = 1$ gives (167) and (168) as

$$V_j = \int_{-\infty}^{\infty} b_j(y)\, dy \tag{169}$$

$$c_j = \frac{\int_{-\infty}^{\infty} y b_j(y)\, dy}{\int_{-\infty}^{\infty} b_j(y)\, dy}. \tag{170}$$

Then the theorem follows if we expand the centroid of B and invoke the SAM assumption $F(x) = \text{Centroid}(B(x)) = \text{Centroid}(\sum_{j=1}^m w_j a_j(x) B_j)$ to rearrange terms

$$F(x) = \text{Centroid}(B(x)) = \frac{\int_{-\infty}^{\infty} y b(y)\, dy}{\int_{-\infty}^{\infty} b(y)\, dy} \tag{171}$$

$$= \frac{\int_{-\infty}^{\infty} y \sum_{j=1}^m w_j b'_j(y)\, dy}{\int_{-\infty}^{\infty} \sum_{j=1}^m w_j b'_j(y)\, dy} \tag{172}$$

$$= \frac{\int_{-\infty}^{\infty} y \sum_{j=1}^m w_j a_j(x) b_j(y)\, dy}{\int_{-\infty}^{\infty} \sum_{j=1}^m w_j a_j(x) b_j(y)\, dy} \tag{173}$$

$$= \frac{\sum_{j=1}^m w_j a_j(x) \int_{-\infty}^{\infty} y b_j(y)\, dy}{\sum_{j=1}^m w_j a_j(x) \int_{-\infty}^{\infty} b_j(y)\, dy} \tag{174}$$

$$= \frac{\sum_{j=1}^m w_j a_j(x) V_j \frac{\int_{-\infty}^{\infty} y b_j(y)\, dy}{V_j}}{\sum_{j=1}^m w_j a_j(x) V_j} \tag{175}$$

$$= \frac{\sum_{j=1}^m w_j a_j(x) V_j c_j}{\sum_{j=1}^m w_j a_j(x) V_j}. \tag{176}$$

Now we give a simple *local* description of optimal lone fuzzy rules [13], [14]. We move a fuzzy rule patch so that it most reduces an error. We look (locally) at a minimal fuzzy system $F : R \to R$ of just one rule. So the fuzzy system is constant in that region: $F = c$. Suppose that $f(x) \neq c$ for $x \in [a, b]$ and define the error

$$e(x) = (f(x) - F(x))^2 = (f(x) - c)^2. \tag{177}$$

We want to find the best place $\hat{x}$. So the first-order condition gives $\nabla e = \mathbf{0}$ or

$$0 = \frac{\partial e(x)}{\partial x} = 2(f(x) - c)\frac{\partial f(x)}{\partial x}. \tag{178}$$

Then $f(x) \neq c$ implies that

$$\frac{\partial e(x)}{\partial x} = 0 \Leftrightarrow \frac{\partial f(x)}{\partial x} = 0 \tag{179}$$

at $x = \hat{x}$. So the extrema of e and f coincide in this case. Fig. 4 shows how fuzzy rule patches can "patch the bumps" and so help minimize the error of approximation. □

REFERENCES

[1] J. A. Dickerson and B. Kosko, "Fuzzy function approximation with supervised ellipsoidal learning," in *Proc. World Congr. Neural Networks (WCNN-93)*, vol. 2, July 1993, pp. 9–17.

[2] ——, "Fuzzy function approximation with ellipsoidal rules," *IEEE Trans. Syst., Man, Cybern.*, pt. B, vol. 26, pp. 542–560, Aug. 1996.

[3] J. J.-S. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 156–159, Jan. 1993.

[4] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neurofuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Englewood Cliffs: Prentice-Hall, 1996.

[5] H. M. Kim and B. Kosko, "Fuzzy prediction and filtering in impulsive noise," *Fuzzy Sets Syst.*, vol. 77, no. 1, pp. 15–33, Jan. 15, 1996.

[6] ——, "Neural fuzzy motion estimation and compensation," *IEEE Trans Signal Processing*, vol. 45, pp. 2515–2532, Oct. 1997.

[7] H. M. Kim and J. M. Mendel, "Fuzzy basis functions: Comparison with other basis functions," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 158–168, May 1995.

[8] G. J. Klir, U. H. St. Clair, and B. Yuan, *Fuzzy Set Theory: Foundations and Applications*. Englewood Cliffs: Prentice-Hall, 1997.

[9] B. Kosko, "Fuzzy knowledge combination," *Int. J. Intell. Syst*, vol. 1, pp. 293–320, 1986.

[10] ——, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs: Prentice-Hall, 1991.

[11] ——, "Fuzzy systems as universal approximators," *IEEE Trans. Computers*, vol. 43, no. 11, pp. 1329–1333, Nov. 1994.

[12] ——, "Combining fuzzy systems," *Proc. IEEE Int. Conf. Fuzzy Systems (IEEE FUZZ-95)*, pp. 1855–1863, Mar. 1995.

[13] ——, "Optimal fuzzy rules cover extrema," *Int. J. Intell. Syst.*, vol. 10, no. 2, pp. 249–255, Feb. 1995.

[14] ——, *Fuzzy Engineering*: Prentice Hall, 1996.

[15] ——, "Global stability of generalized additive fuzzy systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp. 441–452, Aug. 1998.

[16] E. Kreyszig, *Introductory Functional Analysis with Applications*. New York: Wiley, 1978.

[17] S. Lee and R. M. Kil, "Multilayer feedforward potential function network," *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, pp. 141–152, 1988.

[18] S. Mitaim and B. Kosko, "What is the best shape for a fuzzy set in function approximation?," *Proc. 5th IEEE Int. Conf. Fuzzy Systems (FUZZ-96)*, vol. 2, pp. 1237–1243, Sept. 1996.

[19] ——, "Adaptive joint fuzzy sets for function approximation," *Proc. 1997 IEEE Int. Conf. Neural Networks (ICNN-97)*, vol. 1, pp. 537–542, June 1997.
[20] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing unit," *Neural Computat.*, vol. 1, no. 2, pp. 281–294, 1989.
[21] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[22] P. J. Pacini and B. Kosko, "Adaptive fuzzy frequency hopper," *IEEE Trans. Commun.*, vol. 43, pp. 2111–2117, June 1995.
[23] A. Papoulis, *Probability and Statistics*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[24] W. Pedrycz, "Why triangular membership functions?," *Fuzzy Sets Syst.*, vol. 64, pp. 21–30, 1994.
[25] S. J. Press, "Multivariate stable distributions," *J. Multivariate Anal.*, vol. 2, pp. 444–462, 1972.
[26] T. A. Runkler and J. C. Bezdek, "Function approximation with polynomial membership functions and alternating cluster estimation," *Fuzzy Sets Syst.*, vol. 101, pp. 207–218, 1999.
[27] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, July 1971.
[28] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 549–557, 1991.
[29] G. Strang, *Linear Algebra and Its Applications*, 3rd ed. New York: Harcourt Brace Jovanovich, 1988.
[30] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Networks*, vol. 3, pp. 807–814, Sept. 1992.
[31] F. A. Watkins, "The representation problem for additive fuzzy systems," *Proc. IEEE Int. Conf. Fuzzy Systems (IEEE FUZZ-95)*, Mar. 1995.
[32] ——, "The trouble with triangles," in *Proc. INNS World Congr. Neural Networks (WCNN-96)*, San Diego, CA, Sept. 1996, pp. 1123–1126.
[33] J. Zhang and A. Knoll, "Designing fuzzy controllers by rapid learning," *Fuzzy Sets Syst.*, vol. 101, no. 2, pp. 287–301, Jan. 1999.