A Dynamic Regularized Radial Basis Function Network for Nonlinear, Nonstationary Time Series Prediction

Paul Yee, *Member, IEEE,* and Simon Haykin, *Fellow, IEEE*

***Abstract*—In this paper, constructive approximation theorems are given which show that under certain conditions, the standard *Nadaraya-Watson regression estimate (NWRE)* can be considered a specially regularized form of *radial basis function networks (RBFN's)*. From this and another related result, we deduce that regularized RBFN's are m.s. consistent, like the NWRE for the one-step-ahead prediction of Markovian nonstationary, *nonlinear autoregressive* time series generated by i.i.d. noise processes. Additionally, choosing the regularization parameter to be *asymptotically optimal* gives regularized RBFN's the advantage of asymptotically realizing minimum m.s. prediction error. Two update algorithms (one with *augmented networks/infinite memory* and the other with *fixed-size networks/finite memory*) are then proposed to deal with nonstationarity induced by time-varying regression functions. For the latter algorithm, tests on several phonetically balanced male and female speech samples show an average 2.2-dB improvement in the predicted signal/noise (error) ratio over corresponding adaptive linear predictors using the exponentially-weighted RLS algorithm. Further RLS filtering of the predictions from an ensemble of three such RBFN's combined with the usual autoregressive inputs increases the improvement to 4.2 dB, on average, over the linear predictors.**

***Index Terms*—Neural networks, nonlinear, nonstationary, radial basis functions, time-series prediction.**

I. Introduction

ALONG with the multilayer perceptron (MLP), *radial basis function (RBF)* networks hold much interest in the current *neural network (NN)* literature [1]. Their *universal approximation property (UAP)* [2] and straightforward computation using a linearly weighted combination of single hidden-layer neurons have made RBFN's, particularly the *Gaussian RBF (GaRBF)* network, natural choices in such applications as *nonlinear* system identification [3] and time series prediction [4], [5]. In many approaches, the RBFN is trained once on a large example set taken from the unknown plant or times series and believed to capture the essential dynamics of the underlying system. Thereafter, the network is allowed to operate autonomously by sequentially generating outputs in response to newly arriving data. Clearly, such an approach is justifiable only when the dynamics of the plant or time series do not change appreciably over time, which is a condition that is often violated in practice. As a result, recent efforts have been directed toward incorporating some degree of time-adaptivity into the RBFN so that both nonstationary and stationary processes may be *tracked* on an ongoing basis. For example, in the weakly stationary case, one might assume *a priori* that the observed output time series is linear in a number of unknown *state variables* obtained by transforming the observable input time series through a given radial basis function (where the input vector is composed from delayed samples of the input time series). In such a case, if the observed output process is assumed to contain additive white Gaussian noise so that the optimal linear weights are posteriorly Gaussian distributed, we may apply the standard linear Kalman filter (which in this case reduces to the *recursive least-squares (RLS) algorithm*) to recursively estimate the required weights [6]. In [7], this approach is naturally extended to a nonstationary case by using an *extended* version of the RLS algorithm that allows the optimal state-space weights $\boldsymbol{w}^*(i)$ to drift according to a *random walk* model [8]. For modally nonstationary time series, i.e., time series generated by piecewise constant switching amongst a fixed number of state-space mappings and first-order Markovian transition between modes, they further use a *multiple model algorithm* to select (via Bayes inference) the "best" predictor from a number of candidate models running in parallel. Other applications of Bayesian inference in the nonstationary case can be found in [9] and [10]. In these works, however, *arbitrary* nonlinear state-space mappings, i.e., those not necessarily in the linear span of the chosen radial basis functions, are accommodated by extended (in the case of [9]) and iterated (in the case of [10]) extended Kalman filters of second and higher order which produce recursive Bayes estimates of the RBFN weights that best approximate (in mean-square) the nonlinear mapping. As with all methods, the success of these methods hinges on the validity of their accompanying assumptions.

Our interest in this paper centers on the principled design and application of regularized RBFN's to time series prediction. We begin by describing a class of RBFN's designed according to the principles of *regularized least-squares fitting (RLSF)* [11], [12]. With proper statistical considerations, the network class is shown to include asymptotically the well-known *Nadaraya–Watson regression estimate (NWRE)* found in kernel regression [13]–[15]. This relation, along with some

Manuscript received September 5, 1996; revised January 29, 1998. This work was supported in part by a Natural Sciences and Engineering Research Council post-graduate scholarship. The associate editor coordinating the review of this paper and approving it for publication was Dr. Shigera Katagiri.

P. Yee was with the Communications Research Laboratory, McMaster University, Hamilton, Ont., L8S 4K1 Canada. He is now with DATUM Telegraphic, Inc., Vancouver, B.C., V6H 3H8 Canada.

S. Haykin is with the Communications Research Laboratory, McMaster University, Hamilton, Ont., L8S 4K1 Canada.

Publisher Item Identifier S 1053-587X(99)06761-6.

additional results, allows us to prove the (global) mean-square (m.s.) consistency of the RBF class as a *plug-in* predictor for certain ergodic and mixing *nonlinear autoregressive (NLAR)* processes under the same conditions as is known for the NWRE. In particular, this result implies that the RBF class yields m.s. consistent predictors for Markovian NLAR time series generated by i.i.d. noise processes, which can be considered a first generalization of the usual linear AR processes. We also investigate the possibility of *dynamically* updating predictors in this RBF class by developing two *recursive* algorithms, where one gives the network infinite memory and the other finite memory, to deal with the nonstationarity generated by time-varying regression functions. As a practical application of the theory, experimental results for speech prediction are then given in which we also demonstrate how a number of dynamic regularized RBF networks can be linearly combined to improve overall prediction accuracy.

II. Kernel Regression and Regularized Radial Basis Function Networks

The application of kernel regression to the minimum m.s. error (m.m.s.e.) prediction of time series is a firmly established technique; for an overview, see [16] and [17]. In the following, the notation "~" means "is distributed according to," $P_{X,Y}$ denotes the (joint) measure or distribution governing random variables X and Y, and $p_{X,Y}$ denotes the corresponding density.[1] Random variables (r.v's) and processes are generally capitalized, whereas their realizations are indicated by the corresponding lowercase, e.g., T_n is the training set r.v., whereas t_n is a sample realization of T_n.

Assume that we are given a jointly random, discrete-time process $\{(\boldsymbol{Z}(i), Y(i)) \in \mathbb{R}^d \times \mathbb{R},\ i = 1, 2, \cdots\} \sim P_{\boldsymbol{Z}(i),Y(i)}$ with a sufficiently "smooth," time-invariant regression function

$$f(\boldsymbol{z}) \triangleq \mathbb{E}[Y(i)|\boldsymbol{Z}(i) = \boldsymbol{z}]\text{: } \mathbb{R}^d \to \mathbb{R}, \qquad i = 1, 2, \cdots \tag{1}$$

so that

$$Y(i) = f(\boldsymbol{Z}(i)) + B(i), \qquad i = 1, 2, \cdots \tag{2}$$

where $\{B(i)\}$ is a zero-mean random process with $\mathbb{E}[B(i)\boldsymbol{Z}(i)] = \boldsymbol{0}$ for all i. Note such an f exists whenever the joint process $\{(\boldsymbol{Z}(i), Y(i))\}$ is stationary, but the existence of f does not imply the stationarity of the joint process. For a trivial example, take $Z(i) \sim N(0, \sigma_i^2)$ to be a (generally) nonstationary Gaussian process, and $Y(i) \triangleq aZ(i) + b$; clearly, $f(z) = az + b$ independent of i. On the other hand, if f is time varying, then it is clear that the joint process is necessarily nonstationary. We shall have more to say on these matters further on.

The general structure of a kernel regression estimate (KRE) $\tilde{f}_n'$ of f based on a random sample $T_n \triangleq \{(\boldsymbol{Z}(i), Y(i))\}_{i=1}^n \sim P_{T_n}$ is

$$\tilde{f}_n'(\cdot) \triangleq \sum_{j=1}^n W_{n,j}(\cdot) Y(j) \tag{3}$$

where $W_{n,j}(\cdot)$ is a *weight function*. In the sequel, we shall consider weight functions of the form

$$W_{n,j}(\cdot) = \frac{K(||\cdot - \boldsymbol{Z}(j)||/h_n)}{\sum_{i=1}^n K(||\cdot - \boldsymbol{Z}(i)||/h_n)} \tag{4}$$

where K: $\mathbb{R}^+ \to R$ is usually a non-negative, Riemann integrable function rapidly decreasing to zero away from the origin, while $\{h_n\}$ is a sequence of positive *bandwidth* parameters. The resultant function estimate is an instance of the *Nadaraya–Watson estimate (NWRE)* or *normalized KRE* [13]–[15]. With the basic conditions

$$\lim_{n\to\infty} h_n = 0, \quad \lim_{n\to\infty} nh_n^d = \infty \tag{5}$$

on the bandwidth sequence, various modes of asymptotic consistency can be shown to hold for the NWRE in the cases where $\{(\boldsymbol{Z}(i), Y(i))\}$ is an independent, identically distributed (i.i.d.) process and (with slight modifications) a mixing (dependent) process [16], [17]. Of these modes, we shall be generally interested in the pointwise and m.s. modes.

Within the same regression framework, we now consider a particular variant of the regularized RBFN and show that it is a generalization of the NWRE when the two share a common radial kernel (up to a constant scaling factor). To allow a direct relation, we will use the so-called *strict interpolation (SI)* class of regularized RBFN's, where, as with the NWRE, one basis function is assigned to each input datum in the training set. Note that when regularization is present, the term "strict interpolation" refers to this one-to-one correspondence between basis functions (or *centres*) and the training input data and should not be taken to mean that the network is trained to generate a function estimate that agrees exactly with the training data. We shall generally omit the "SI" designation for the regularized RBFN's used in the sequel, except where necessary to emphasize some particular aspect of the SI construction.

Recall that for a regularized RBFN designed to solve the least-squares interpolation problem over a random sample T_n, the estimate of f is given in general form by the linear expansion

$$\tilde{f}_n(\cdot) \triangleq \boldsymbol{w}_n^\top \boldsymbol{g}_n(\cdot) \tag{6}$$

where

$$\boldsymbol{g}_n(\cdot) \triangleq [g_j(\cdot)]_{j=1}^n \tag{7}$$

$$g_j(\cdot) \triangleq K(||\cdot - \boldsymbol{Z}(j)||_{\boldsymbol{U}_n}) \tag{8}$$

$\{\boldsymbol{Z}(j)\}_{j=1}^n$ are the centres of the expansion, and the notation $||\cdot||_{\boldsymbol{U}_n}$ indicates the Euclidean norm in $\mathbb{R}^d$ weighted by a symmetric positive definite matrix $\boldsymbol{U}_n$. The linear weights $\boldsymbol{w}_n$ are then determined as the solution to

$$(\boldsymbol{G}_n + \lambda_n \boldsymbol{I})\boldsymbol{w}_n = \boldsymbol{Y}_n \tag{9}$$

[1] All densities in this paper are taken with respect to Lebesgue measure unless otherwise specified.

where

$$\boldsymbol{G}_n \triangleq \begin{bmatrix} \boldsymbol{g}_n^\top(\boldsymbol{Z}(1)) \\ \vdots \\ \boldsymbol{g}_n^\top(\boldsymbol{Z}(n)) \end{bmatrix} \tag{10}$$

is the symmetric, positive definite *interpolation matrix* and

$$\boldsymbol{Y}_n \triangleq [Y(j)]_{j=1}^n \tag{11}$$

is the vector of desired outputs or *targets* for the interpolation problem. The $\{\lambda_n \in \mathbb{R}^+\}$ is a sequence of *regularization* parameters that in the deterministic case, trades off the fidelity of the resultant interpolation over the sample data with the smoothness of the estimator $\tilde{f}_n$. From a deterministic point of view, the estimate (6) is optimal in the sense that it is the unique solution of the associated regularized variational interpolation problem

$$\tilde{f} \triangleq \arg\min_{f \in \mathcal{S}} \left(\sum_{i=1}^{N} (y(i) - f(\boldsymbol{z}(i)))^2 + \lambda ||Df||_2^2 \right) \tag{12}$$

where

$\mathcal{S}$	suitable space of "smooth" functions;
D	(pseudo) differential operator over $\mathcal{S}$;
$\|\|\cdot\|\|_2$	L_2 norm.

It is the choice of D that determines the kernel K for the regularized RBFN. For example, Gaussian kernels of the form $K(r) = \exp(-r^2/2)$ correspond to operators D defined by an infinite series of exponentially weighted iterated Laplacians with increasing order and oriented according to the input norm weighting matrix $\boldsymbol{U}_n$.

In this sense, the estimate $\tilde{f}$ constructed above is the "smoothest" function consistent (up to the regularization parameter λ) with the training data. For more details on the deterministic RBF interpolation problem, see [18].

To compare the two estimator structures, we may rewrite the NWRE general form as

$$\tilde{f}_n'(\cdot) = \boldsymbol{W}_n^\top(\cdot)\boldsymbol{Y}_n \tag{13}$$

where $\boldsymbol{W}_n(\cdot) \triangleq [W_{n,j}(\cdot)]_{j=1}^n$. Moreover, by substituting (9) into (6), the RBFN can be expressed as

$$\tilde{f}_n(\cdot) \triangleq \boldsymbol{g}_n^\top(\cdot)(\boldsymbol{G}_n + \lambda_n \boldsymbol{I})^{-1}\boldsymbol{Y}_n \tag{14}$$

thus showing that the RBFN is a KRE-type method with an *effective* weighting function $\boldsymbol{W}_n^\top(\cdot) \triangleq \boldsymbol{g}_n^\top(\cdot)(\boldsymbol{G}_n + \lambda_n \boldsymbol{I})^{-1}$. The similarity of the RBFN weighting function to that of the NWRE suggests that the two should be parametrically related (an intuition that is largely correct), as we shall see. We should mention that while there has been previous work relating RBFN's to the NWRE [19], that work considered only normalized, nonregularized RBFN's in which the parameters are explicitly chosen to approximate the form of a corresponding KRE.

Let us define a special class $F_{\boldsymbol{z}}$ of regularized RBFN's in which λ_n (and hence $\boldsymbol{w}_n$) is permitted to vary with its input $\boldsymbol{z} \in \mathbb{R}^d$. This class is a slight generalization of the usual class of regularized RBFN's in which λ_n (and hence $\boldsymbol{w}_n$) is set once on the basis of a realized training set t_n for all inputs $\boldsymbol{z}$. As will be explained further on, the generalization does not affect the overall tenor of the results. In the theorem and proofs, the related concept of the *Parzen window (density) estimate (PWE)* [20] also plays a central role.

Theorem 1: Assume that $\{\boldsymbol{Z}(i)\}$ has a stationary marginal measure P and density p. Let D be a compact subset of $\mathbb{R}^d$ with $p(\boldsymbol{z}) > 0$ for all $\boldsymbol{z} \in D$. Given an NWRE $\tilde{f}_n'$ with kernel K' and supremum $C \triangleq \sup_{\boldsymbol{z} \in \mathbb{R}^d} K'(\boldsymbol{z})$, define $\boldsymbol{g}_n'$ as $\boldsymbol{g}_n$ in (8) with K' in place of K and the associated PWE $\tilde{p}_n$: $\mathbb{R}^d \to \mathbb{R}^+$ of the input density as $nh_n^d \tilde{p}_n(\boldsymbol{z}) \triangleq \boldsymbol{1}_n^\top \boldsymbol{g}_n'(\boldsymbol{z})$, $\boldsymbol{z} \in \mathbb{R}^d$, where $\boldsymbol{1}_n$ is a constant vector of n ones. Then, we have the following.

1) If $|Y(i)| < M$ almost surely (a.s.) for all i and if K', $\{h_n\}$, and p are such that

$$\sup_{\boldsymbol{z} \in D} |\tilde{p}_n(\boldsymbol{z}) - p(\boldsymbol{z})| \overset{n \to \infty}{\to} 0 \tag{15}$$

then $\exists N = N(p, D, K', \{h_n\})$ such that for any $n > N$ and $\alpha > \max(2, \log(2C/(h_n^d m))/\log n)$, a regularized RBFN $\tilde{f}_{n,\infty} \in F_{\boldsymbol{z}}$ may be constructed such that

$$\sup_{\boldsymbol{z} \in D} |\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})| = \mathcal{O}(C^2 M n^{-\alpha} h_n^{-2d} m^{-2}) \overset{n \to \infty}{\to} 0 \text{ a.s.-}P_{T_n} \tag{16}$$

where $m = m(D) \triangleq \inf_{\boldsymbol{z} \in D} p(\boldsymbol{z})$.

2) If $\mathbb{E}[Y^2(i)] < M^2$ for all i and if K', $\{h_n\}$, and p are such that

$$\sup_{\boldsymbol{z} \in D} \mathbb{E}_{T_n}[|\tilde{p}_n(\boldsymbol{z}) - p(\boldsymbol{z})|^2] \overset{n \to \infty}{\to} 0 \tag{17}$$

and there exists positive constants R_1, R_2, R_3, and ν such that

$$\lim_{n \to \infty} \sup_{\boldsymbol{z} \in D} (|\tilde{f}_{n,\infty}(\boldsymbol{z})| + |\tilde{f}_n'(\boldsymbol{z})|) < R_1 \text{ a.s.-}P_{T_n} \tag{18}$$

$$\lim_{n \to \infty} \sup_{\substack{i,j=1,\cdots,n \\ (\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^d}} \frac{p_{ij}(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} < R_2 \tag{19}$$

$$\lim_{n \to \infty} \inf_{\boldsymbol{z} \in D} n^\nu n h_n^d \tilde{p}(\boldsymbol{z}) > R_3 > 0 \text{ a.s.-}P_{T_n} \tag{20}$$

where $p_{ij}(\cdot, \cdot)$: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ is the joint density for $\boldsymbol{Z}(i)$ and $\boldsymbol{Z}(j)$, then $\exists N = N(p, D, K', \{h_n\})$ such that for $n > N$ and $\alpha > \max(1, \nu, \log(2C/(h_n^d m))/\log n)$, a regularized RBFN $\tilde{f}_{n,\infty} \in F_{\boldsymbol{z}}$ may be constructed such that

$$\sup_{\boldsymbol{z} \in D} \mathbb{E}_{T_n}[|\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})|^2] = \mathcal{O}(C^2 M \sqrt{L} ||p||_2 ||K||_2^2 n^{-\alpha} h_n^{-d} m^{-2}) \overset{n \to \infty}{\to} 0 \tag{21}$$

where m is as before, and $L = L(D) \triangleq \sup_{\boldsymbol{z} \in D} p(\boldsymbol{z})$.

Proof: See Appendix A. □

As an aside, we may find in the literature numerous sets of conditions under which (15) and (17) hold. In particular, we refer to Lemma 2.1, Theorem 2.2, and Corollary 2.2 in the case of (15), and Theorem 2.1 and Corollary 2.1 in the case of (17), all from [17]. For the purposes of this paper, it suffices to

mention that the conditions include the case where the input process $\{\boldsymbol{Z}(i)\}$ is dependent, i.e., correlated, according to a *mixing* condition [21], [22]. The forms of mixing allowed in the cited theorems [2-α in the case of (17) and the stronger *geometrically strong mixing (GSM)* in the case of (15)] are less restrictive than other types of mixing conditions commonly assumed, e.g., ϕ and ρ-mixing, and include classical ARMA as well as i.i.d. processes.

From the construction of the approximating RBFN detailed in the proof of Theorem 1, we see that for sufficiently large training sets, the NWRE corresponds to a (specially) regularized RBFN for which $\lambda_n \overset{n\to\infty}{\to} \infty$ at an appropriate rate. From the RLSF theory, however, we know that by choosing the regularization parameter sequence $\{\lambda_n\}$ via an *asymptotically optimal (a.o.)* procedure, the resultant sequence of regularized RBFN's has an asymptotic *risk* (as defined below) which is minimum over all possible choices of regularization parameter sequences, including the ones with $\lambda_n \overset{n\to\infty}{\to} \infty$ present in the family $F_{\boldsymbol{z}}$ and (by approximation) the NWRE. If we can then determine conditions under which this "risk" converges (in n) to the desired global m.s.e., then by comparison, the regularized RBFN constructed with an a.o.-selected regularization parameter sequence should be m.s. consistent whenever the NWRE is. Indeed, this line of reasoning is pursued in the next section, where we prove the m.s. consistency of the plug-in predictor formed from the regularized RBFN for a Markovian NLAR time series generated by an i.i.d. noise process.

III. Prediction Using Regularized RBFNs

Define the *approximation error* for a regularized RBFN $\tilde{f}_n$ at time step i as

$$\delta_n(i,\tilde{\lambda}_n) \triangleq f(\boldsymbol{z}(i)) - \tilde{f}_n(\boldsymbol{z}(i),\tilde{\lambda}_n), \qquad i=1,2,\cdots \quad (22)$$

the *loss* of $\tilde{f}_n$ with respect to its training set t_n as

$$L_n(\tilde{\lambda}_n,t_n) \triangleq \frac{1}{n}\sum_{i=1}^{n}\delta_n^2(i,\tilde{\lambda}_n) \quad (23)$$

and the *risk* as

$$R_n(\tilde{\lambda}_n) \triangleq \mathbb{E}_{T_n}[L_n(\tilde{\lambda}_n,T_n)] \quad (24)$$

where we have indicated explicitly the dependence of $\tilde{f}_n$ (hence, δ_n and L_n) on the chosen regularization parameter $\tilde{\lambda}_n$; this dependence will be omitted when it is clear from context.

The main result that we shall exploit from RLSF theory is that the "optimal" regularization parameter λ_n^* that minimizes the risk[2] lies between zero and infinity, except in certain pathological cases [11], [23]. While this conclusion has some bearing on the quality of the *global estimate* $\tilde{f}_n$ of f, we are more interested in the corresponding implications for the (pointwise) *plug-in predictor* formed from the estimate of f as $\tilde{Y}(n+1)=\tilde{f}_n(\boldsymbol{Z}(n+1))$. In particular, we can show that the risk converges (in n) to the m.s. value of the *prediction error* $\epsilon_n(n+1)$ defined as

$$\epsilon_n(n+1) \triangleq Y(n+1)-\tilde{Y}(n+1). \quad (25)$$

With a view to the speech prediction experiments, we shall restrict our attention to the specific case of a Markovian *nonlinear autoregressive (NLAR)* process $\{X(i)\}$ of order p and delay k, i.e.,

$$X(i)=f(\boldsymbol{X}_p(i-k))+B(i), \qquad i=1,2,\cdots \quad (26)$$

for $k\geq 1$, where $\boldsymbol{X}_p(i) \triangleq [X(i),X(i-1),\cdots X(i-p+1)]^\top$, and $\{B(i)\}$ is an i.i.d. noise process with zero mean, bounded variance σ^2 and independent of the initial state vector $\boldsymbol{X}_0 \triangleq \boldsymbol{X}_p(-k)$. Thus, we have an instance of the general regression case with $Y(i)=X(i)$ and $\boldsymbol{Z}(i)=\boldsymbol{X}_p(i-k)$ or, equivalently, $Y(i)=X(i+k)$ and $\boldsymbol{Z}(i)=\boldsymbol{X}_p(i)$ (we shall use either notations as convenient). Note that for $k=1$, the vector input process $\{\boldsymbol{X}_p(i)\}$ satisfies a similar recurrence

$$\begin{aligned}\boldsymbol{X}_p(i) &= [f(\boldsymbol{X}_p(i-1)),X(i-1),X(i-2),\cdots \\ &\qquad X(i-p+1)]^\top+\epsilon(i)\boldsymbol{e}_1 \\ &\triangleq T(\boldsymbol{X}_p(i-1))+\epsilon(i)\boldsymbol{e}_1 \end{aligned} \quad (27)$$

where $\boldsymbol{e}_1 \triangleq [1,0,\cdots,0]^\top$ is the first unit vector in $\mathbb{R}^p$. Discussion of other more general processes, e.g., the case where $\{B(i)\}$ is a heteroskedastic (but still zero mean) noise process, can be found in [23]. For general f, the vector input process $\{\boldsymbol{X}_p(i)\}$ is clearly

a) dependent (by the autoregressive construction);
b) nonstationary (by the action of f).

To deal with these issues, we may impose conditions on f and the measure P_B for $\{B(i)\}$ such that

a) the dependence follows a mixing condition admissible under Theorem 1;
b) $\{\boldsymbol{X}_p(i)\}$ is "asymptotically stationary" in a sense to be explained below.

A sufficient set of conditions that meets both requirements for $k=1$ follow.

A.1) $\{B(i)\}$ satisfies $\mathbb{E}[|B(i)|]<\infty$ and has an everywhere continuous and positive density with respect to Lebesgue measure.
A.2) f is bounded and Lipschitz in $\mathbb{R}^p$, has $f(\mathbf{0})=0$ (so that $T(\mathbf{0})=\mathbf{0}$), and is *exponentially asymptotically stable in the large*, i.e., $\exists A$, $c>0$ such that $\forall n\in\mathbb{Z}^+$ and $\boldsymbol{x}(0)\in\mathbb{R}^p$, $||\boldsymbol{x}(n)|| \leq A\exp(-cn)||\boldsymbol{x}(0)||$, where $\boldsymbol{x}(n) \triangleq T^n(\boldsymbol{x}(0))$ is the n-fold composition of T applied to $\boldsymbol{x}(0)$.

Of the two conditions, the second is obviously the more restrictive one because it requires that the underlying mapping f satisfy a rather strong contractivity condition (although it does allow the stable point of the map T to be other than $\mathbf{0}$ by applying a suitable translation). Exponential decay in transiently driven physical systems is quite plausible, how-

[2] Reference [11] gives this result for the usual case of the input-conditioned version of the risk, whereas [23] extends this result to the (unconditional) risk defined above.

ever, which implies that the exponentially asymptotic stability condition may hold at least locally within a given time series.

Under these chosen conditions, it can be shown for $k = 1$ that the vector input process $\{\boldsymbol{Z}(i)\} \triangleq \{\boldsymbol{X}_p(i)\}$ is

a) *geometrically ϕ-mixing (GPM)* [16], hence, GSM (since ϕ-mixing implies α-mixing);
b) *geometrically ergodic*, i.e., the sequence of marginal measures $\{P_{\boldsymbol{Z}(i)}\}$ converges at geometric rate (in total variation norm[3] $||\cdot||_V$ as $i \to \infty$) to a common measure $P_{\boldsymbol{Z}}$ [21], [24].

The first consequence implies that the dependence created in (26) is compatible with the mixing conditions supported in Theorem 1, whereas the second consequence essentially states that the marginal input measures (and densities) for the r.v.s $\boldsymbol{X}_p(i)$ approach a common (stationary) measure geometrically fast as i increases. We should mention that we have chosen this rather weak form of nonstationarity primarily to simplify the exposition; other conditions can be chosen to permit stronger forms of nonstationarity [23]. The main point to be demonstrated here is that with these selected conditions, the NWRE is an appropriate, i.e., consistent, predictor that can be approximated according to the Theorem 1.

Returning to the analysis of the m.s. prediction error for (26) in the case $k = 1$, elementary expansions yield

$$\begin{aligned}
&\mathbb{E}[\epsilon_n^2(n+1)] \\
&\quad = \mathbb{E}[|f(\boldsymbol{Z}(n+1)) + B(n+1) - \tilde{f}(\boldsymbol{Z}(n+1))|^2] \\
&\quad = \mathbb{E}[|f(\boldsymbol{Z}(n+1)) - \tilde{f}(\boldsymbol{Z}(n+1))|^2] \\
&\qquad + 2\mathbb{E}[B(n+1)\{f(\boldsymbol{Z}(n+1)) - \tilde{f}(\boldsymbol{Z}(n+1))\}] + \sigma^2 \\
&\quad \triangleq \mathbb{E}[\delta_n^2(n+1)] + \sigma^2
\end{aligned}$$

where the cross-term in the second line vanishes by the independence of $B(n+1)$ from $\{B(i)\}_{i=1}^n$, and hence, $\{\boldsymbol{Z}(i)\}_{i=1}^n$. Clearly, it is sufficient to relate the risk of $\tilde{f}_n$ to its m.s. approximation error at time step $n+1$, as we do in the following.

Theorem 2: Assume that conditions A.1) and A.2) hold. If, in addition, a) the density of the stationary measure for $\{\boldsymbol{Z}(i)\}$ is bounded and b) the sequence of estimators $\{\tilde{f}_n\}$ is uniformly bounded a.s.-P_{T_n} with a correspondingly bounded sequence of Lipschitz constants, then

$$|\mathbb{E}[\delta_n^2(n+1)] - R_n(\tilde{\lambda}_n)| \overset{n\to\infty}{\to} 0 \text{ a.s.-}P_{T_n}. \tag{28}$$

Proof: See Appendix B. □

The ramifications of this result are two fold:

a) that RBFN training procedures aimed at minimizing (asymptotically) the risk, such as the a.o. parameter selection methods for the regularization parameter sequence described earlier, are also sensible from a m.m.s.e. prediction point-of-view;
b) that the m.s. consistency of such an a.o.-trained RBFN follows from that of the corresponding[4] NWRE whenever conditions admit the approximation results of Theorem 1.

On the latter point, we note that if the initial state r.v. $\boldsymbol{Z}(0)$ is a.s. bounded in norm by a constant R, then the an appropriate compact set for the application of Theorem 1 is $D = \{\boldsymbol{z} \in \mathbb{R}^p\colon\ ||\boldsymbol{z}|| \le A \cdot R\}$, where A is as defined in A.2. Furthermore, it is not difficult to see that Theorem 1 also holds for geometrically ergodic input processes by replacing the common measure P, density p, and joint densities p_{ij} in the proofs with the invariant measure π, density p_π, and joint densities $p_{\pi,ij}$, respectively (e.g., see the discussion regarding pointwise convergence of the marginal input densities to the invariant density in the proof of Theorem 2). Therefore, Theorem 1 remains valid under our chosen NLAR process conditions.

By the argument stated at the end of Section II, Theorem 1 (with the indicated modifications) and Theorem 2 allow us to conclude that the regularized RBFN predictor is m.s. consistent for the NLAR processes considered. While the NWRE predictor is also consistent, we know from the discussion of asymptotic optimality at the end of Section II that only the regularized RBFN has the flexibility of selecting a sequence $\{\lambda_n\}$ that yields near-minimal risk once n is sufficiently large; the NWRE, with its effectively unbounded regularization parameter sequence, will generally have greater asymptotic risk and, hence, m.s. prediction error. We should add that although the particular NLAR process conditions we have chosen are somewhat restrictive, they do allow the use of the *generalized cross-validation (GCV)* procedure for calculating such an a.o. sequence of regularization parameters [12]. It can be shown that the regularization parameter sequence produced by the GCV procedure is invariant to rotations of the data axes in (2). Under the more general condition of independent but heteroskedastic $B(i)$, only the *leave-out-one* or *ordinary cross-validation (OCV)* procedure is currently known to guarantee a.o. estimates of the true risk-minimizing regularization parameter sequence $\{\lambda_n^*\}$ [25], [26], but this procedure does not share the rotational invariance property of the GCV procedure.

IV. Recursive Updating for Regularized RBFN Predictors

As there is no substantial difficulty in doing so, we shall, where possible, develop the subsequent algorithms for a general pair of input/output processes $\{\boldsymbol{Z}(i), Y(i)\}$ rather than specifically for the autoregressive case $Y(i) \triangleq X(i)$ and $\boldsymbol{Z}(i) \triangleq \boldsymbol{X}_p(i-1)$. Thus far, both the NWRE and regularized RBFN assume that the process to be predicted admits a time-invariant regression function; in practice, as our speech prediction experiment will show, this condition does not always hold. If the regression function f drifts slowly with time

[3] The total variation norm $||\cdot||_V$ for the space $\mathcal{L}$ of probability measures over $\mathcal{B}(\mathbb{R}^d)$ is defined as $||P - Q||_V \triangleq \sup_{B\in\mathcal{B}(\mathbb{R}^d)} |P(B) - Q(B)|$, where $P, Q \in \mathcal{L}$.

[4] By "corresponding NWRE," we mean the NWRE trained with the same data and sharing the same kernel (up to a constant scaling factor) and bandwidth sequence as a given RBFN; see the proof of Lemma 1 in Appendix A.

TABLE I
NWRE BASIC FIXED-SIZE PREDICTION UPDATE ALGORITHM

Initialization: assume the NWRE has been generated from $t_n(i)$ in the usual way, i.e., via equations (3) and (4) with $t_n(i)$ in place of T_n.

Updating: when the new datum $(\boldsymbol{z}(i+1), y(i+1))$ becomes available,

1. replace the basis function $K(\|\bullet - \boldsymbol{z}(i-n+1)\|/h_n)$ with $K(\|\bullet - \boldsymbol{z}(i+1)\|/h_n)$ in (3).
2. replace the corresponding prediction target $y(i-n+1)$ with $y(i+1)$ in (3).

Prediction: for the NLAR case $y(i) \triangleq x(i)$ and $\boldsymbol{z}(i) \triangleq \boldsymbol{x}_p(i-1)$, set $\tilde{y}(i+2) = \tilde{f}_{n,i+1}(\boldsymbol{x}_p(i+1))$.

Iteration: $i \rightarrow i+1$ and repeat from Updating step.

index i as f_i, i.e., exhibits a form of *local stationarity*, the idea of updating the regression function parameters periodically, say, every l time steps, as new data arrive is intuitively appealing, particularly when it can be performed *efficiently* in a *recursive* fashion. The basis of comparison will be the standard adaptive linear estimation procedures such as the recursive least-squares (RLS) algorithm. Let us consider the limiting case $l = 1$ and assume for now that n, which is the size of the training set and, hence, the number of basis functions in the estimate for f_i, is fixed. Before continuing, let us set the notations for the following discussion.

Subscripts: For *vector* and (square) *matrix quantities*, the *first subscript* refers to its *dimension*, whereas for a *scalar quantity*, it refers to the dimension of the associated vector or matrix quantity being indexed. The *second subscript*, if present, refers to *either* the *time index* of the training set from which the quantity is constructed (in the case of a *scalar* or *vector function*) or a *particular element* of that quantity (in the case of an *ordinary vector*). If a vector quantity's second subscript consists of the notation $a{:}b$, then we are referring to the subvector formed from the ath element to the bth element inclusive.

Parenthesized Arguments: For *nonfunctional quantities*, a parenthesized argument indicates *time dependence*, i.e., $\cdot(i)$ mean quantity $\cdot$ uses data up to and including time step i. For functions, it indicates the usual argument.

As an example, $t_n(i) \triangleq \{(\boldsymbol{z}(j), y(j))\}_{j=i-n+1}^{i}$ denotes the realized training set for the network at time step i, where in the NLAR case, this training set is formed from the time series segment $\{x(j)\}_{j=i-n-p+1}^{i}$. Then, $\boldsymbol{g}_{n,i}(\cdot)$ corresponds to $\boldsymbol{g}(\cdot)$ in (8), and $\boldsymbol{w}_{n,j}(i)$ corresponds to the jth element of $\boldsymbol{w}$ in (9) when $t_n(i)$ is used in place of t_n.

Given $t_n(i)$, which is a realized set of input/output examples for f_i, and $\tilde{f}_{n,i}$, which is the corresponding regression function estimate, the problem is to recursively compute $\tilde{f}_{n,i+1}$, which is the estimate associated with $t_n(i+1)$, from $\tilde{f}_{n,i}$. For the NWRE, this network updating and subsequent prediction are simple, as shown in Table I. If we are using some data-based method of selecting the bandwidth, it may also be advantageous to adjust the bandwidth from $h_n = h_n(i)$ to $h_n(i+1)$ at the same time. The basic order of the updating, excluding the cost of computing an updated bandwidth parameter, for the NWRE is $\mathcal{O}(1)$, and that of computing the prediction $\tilde{y}(i+1)$ is $\mathcal{O}(n)$.

For the regularized RBFN, we shall analyze the effect of the one-step updating in two stages and thereby find interesting parallels to the standard RLS estimation algorithm. In the first stage, we allow the size of the RBFN to grow with incoming data so that one weight is added per update, leading to an *augmented* network with *infinite memory* (cf. for linear adaptive filters, this growth is usually called *order recursion* [e.g., see [27, ch. 15]). The second stage is to simultaneously add one (new) weight and truncate the oldest weight per update, leading to a network of *fixed size* with *finite memory*.

This idea of augmenting a RBFN with incoming data was previously introduced in [28] and later in [9]. Compared with the latter work, our approach is developed as an optimal recursive solution to a local interpolation problem and is thus solidly grounded in the theory of RLSF, which deals with noise in principled and explicit fashion. In contrast, the sequential function estimation (s.f.e.) approach of the latter work assumes that the training data are noise-free, which may not be realistic in many applications. To ameliorate the influence of noise and to limit the network growth with their s.f.e. approach, the latter work then proposes a growth criterion based on Hilbert function space geometry according to both prediction error and distance criteria. While such criteria may be intuitively appealing, no theoretical guidance is provided on the proper selection of the criteria parameters, nor are the conditions required for their effective application characterized. By building on the significant body of knowledge surrounding RLSF and KRE for time series estimation, we are able to provide analyses of our algorithmic choices and their effect on prediction performance.

A. Augmented (Infinite Memory) Case

We begin by decomposing the $(n+1)\times(n+1)$ regularized SI equation for the *combined* realized training set $t_{n+1}(i+1) = t_n(i) \cup t_n(i+1)$ as

$$\left(\begin{bmatrix} \boldsymbol{G}_n(i) & \boldsymbol{\gamma}_n(i+1) \\ \boldsymbol{\gamma}_n^\top(i+1) & K(0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Lambda}_n(i) & \boldsymbol{0} \\ \boldsymbol{0}^\top & \lambda_{n+1}(i+1) \end{bmatrix}\right) \cdot \left(\begin{bmatrix} \boldsymbol{w}_n(i) \\ 0 \end{bmatrix} + \begin{bmatrix} \Delta\boldsymbol{w}_n(i) \\ w_{n+1,n+1}(i+1) \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{y}_n(i) \\ y(i+1) \end{bmatrix} \tag{29}$$

which we may write more compactly as

$$\begin{bmatrix} \boldsymbol{F}_n(i) & \boldsymbol{\gamma}_n(i+1) \\ \boldsymbol{\gamma}_n^\top(i+1) & K(0) + \lambda_{n+1}(i+1) \end{bmatrix} \cdot \left(\begin{bmatrix} \boldsymbol{w}_n(i) \\ 0 \end{bmatrix} + \begin{bmatrix} \Delta\boldsymbol{w}_n(i) \\ w_{n+1,n+1}(i+1) \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{y}_n(i) \\ y(i+1) \end{bmatrix}$$
$$\boldsymbol{F}_{n+1}(i+1) \cdot \boldsymbol{w}_{n+1}(i+1) = \boldsymbol{y}_{n+1}(i+1) \tag{30}$$

where $\boldsymbol{F}_n(i) \triangleq \boldsymbol{G}_n(i) + \boldsymbol{\Lambda}_n(i)$, and $\boldsymbol{\gamma}_n(i)$ is the vector formed from the first n elements of the last column of $\boldsymbol{G}_{n+1}(i)$, i.e., $\boldsymbol{\gamma}_n(i) \triangleq [\boldsymbol{g}_{n,i}(\boldsymbol{z}(i))]_{1:n}$ (the notation $i{:}j$ means indices i to j inclusive). Here, as a slight generalization, $\boldsymbol{\Lambda}_n(i) \triangleq \operatorname{diag}(\lambda_n(i-j), j = n-1, n-2, \cdots, 0)$ is the diagonal weighting matrix formed from the most recent n regularization parameters up to and including time step i. Let $\boldsymbol{w}_n(i)$ be the previously computed solution to the regularized SI equation $(\boldsymbol{G}_n(i) + \boldsymbol{\Lambda}_n(i))\boldsymbol{w}_n(i) = \boldsymbol{y}_n(i)$ over $t_n(i)$. We

TABLE II
REGULARIZED RBFN AUGMENTED PREDICTION UPDATE ALGORITHM

Initialization: assume the regularized RBFN has been generated from $t_n(i)$ in the usual way, i.e., via equations (6) to (9) with $t_n(i)$ in place of T_n, and assume that $\boldsymbol{F}_n^{-1}(i)$ is known.

Updating: when the new datum $(\boldsymbol{z}(i+1), y(i+1))$ becomes available,

1. select the new regularization parameter $\lambda_{n+1}(i+1)$ and the norm weighting matrix $\boldsymbol{U}_{n+1}(i+1)$, typically from $t_{n+1}(i+1)$.
2. compute the new basis function vector $\boldsymbol{\gamma}_n(i+1)$.
3. compute $\left(\boldsymbol{F}_n(i) - \frac{\boldsymbol{\gamma}_n(i+1)\boldsymbol{\gamma}_n^\top(i+1)}{\lambda_{n+1}(i+1)+K(0)}\right)^{-1}$. Note the complexity of this calculation may be reduced to $\mathcal{O}(n^2)$ if $\boldsymbol{F}_n^{-1}(i)$ is optionally propagated from time step to time step as indicated below, since the Sherman-Woodbury-Morrison formula [40] (or *matrix inversion lemma* in the statistical signal processing field [27]) for the inverse of the sum of a given matrix and a low rank perturbation may be applied.
4. compute the weight change vector $\Delta\boldsymbol{w}_n(i)$ according to (32).
5. add the weight change vector $\Delta\boldsymbol{w}_n(i)$ to the existing network weight vector.
6. compute the new weight $w_{n+1,n+1}(i+1)$ via (32).
7. add the new basis function $K\left(\|\bullet - \boldsymbol{z}(i+1)\|_{\boldsymbol{U}_{n+1}(i+1)}\right)$ with weight $w_{n+1,n+1}(i+1)$ to the network
8. (optional) compute $\boldsymbol{F}_{n+1}^{-1}(i+1)$ from $\boldsymbol{F}_n^{-1}(i)$ with complexity $\mathcal{O}(n^2)$ via a partitioned matrix inverse formula applied to the decomposition (30).

Prediction: for the NLAR case $y(i) \triangleq x(i)$ and $\boldsymbol{z}(i) \triangleq \boldsymbol{x}_d(i-1)$, set $\widetilde{x}(i+2) = \widetilde{f}_{n+1,i+1}(\boldsymbol{x}_d(i+1))$.

Iteration: $i \to i+1$, $n \to n+1$ and repeat from Updating step.

assume that the new regularization parameter $\lambda_{n+1}(i+1)$ has been chosen on the basis of $t_{n+1}(i+1)$. The objective is to find the new weight $w_{n+1,n+1}(i+1)$ and the weight change vector $\Delta\boldsymbol{w}_n(i)$ to be applied to $\boldsymbol{w}_n(i)$ such that the *augmented* regularized SI equation (29) is satisfied. The solution is

$$w_{n+1,n+1}(i+1) = \frac{y(i+1) - (\boldsymbol{w}_n(i) + \Delta\boldsymbol{w}_n(i))^\top \boldsymbol{\gamma}_n(i+1)}{\lambda_{n+1}(i+1) + K(0)} \tag{31}$$

$$\Delta\boldsymbol{w}_n(i) = -\left(\boldsymbol{F}_n(i) - \frac{\boldsymbol{\gamma}_n(i+1)\boldsymbol{\gamma}_n^\top(i+1)}{\lambda_{n+1}(i+1)+K(0)}\right)^{-1} \cdot \frac{\boldsymbol{\gamma}_n(i+1)}{\lambda_{n+1}(i+1)+K(0)}(y(i+1) - \boldsymbol{w}_n^\top(i)\boldsymbol{\gamma}_n(i+1)). \tag{32}$$

The resultant prediction update algorithm is listed in Table II. Because $\boldsymbol{\gamma}_n(i+1)$ is also the vector of basis function outputs of the previous network from time step i in response to the newly available input $\boldsymbol{z}(i+1)$, we see that the new weight $w_{n+1,n+1}(i+1)$ is merely a scaled version of the *a posteriori* estimation error, i.e., the estimation error that would have been obtained had the previous weight vector $\boldsymbol{w}_n(i)$ been updated to $\boldsymbol{w}_n(i) + \Delta\boldsymbol{w}_n(i)$. In contrast, the weight change vector $\Delta\boldsymbol{w}_n(i)$ is proportional to the *a priori* estimation error, i.e., the actual estimation error using the previous weight vector $\boldsymbol{w}_n(i)$ prior to any updating, which is similar to what occurs in the RLS algorithm. This partitioning of roles between $w_{n+1,n+1}(i+1)$ and $\Delta\boldsymbol{w}_n(i)$ is intuitively satisfying; the change $\Delta\boldsymbol{w}_n(i)$ applied to the existing weight vector attempts to account for estimation error incurred by the existing (nonupdated) network, whereas the new weight element $w_{n+1,n+1}(i+1)$ attempts to account for the estimation error remaining after the existing network has been updated. Analogous to the RLS algorithm, we may also expect the ratio of the m.s. *a priori* and the m.s. *a posteriori* estimation errors to converge to unity as $n \to \infty$ if the regression function being estimated is not significantly time varying. If the ratio is nonconvergent, it may be an indication that old training samples are no longer representative of the regression function behavior currently being estimated. For this situation, the effective *memory* of the RBFN can be limited by fixing its size to n weights/basis functions computed from the most recent n training data available, which leads us to the second stage of updating described next.

B. Fixed-Size (Finite Memory) Case

Let us return to the original task and assume that the size of the RBFN is fixed at n weights/basis functions. The desire is to relate $\boldsymbol{w}_n(i+1)$, which are the weights satisfying the regularized SI equation over $t_n(i+1)$, to the previously computed weights $\boldsymbol{w}_n(i)$, which do the same for $t_n(i)$. Before we do so, let us establish the notations. Decompose the $n \times n$ regularized SI equation for the previous training set $t_n(i)$ as

$$\begin{bmatrix} \lambda_n(i-n+1)+K(0) & \boldsymbol{\beta}_{n-1}^\top(i) \\ \boldsymbol{\beta}_{n-1}(i) & \boldsymbol{F}_{n-1}(i) \end{bmatrix} \begin{bmatrix} w_{n,1}(i) \\ \boldsymbol{w}_{n,2:n}(i) \end{bmatrix} = \begin{bmatrix} y(i-n+1) \\ \boldsymbol{y}_{n,2:n}(i) \end{bmatrix}$$
$$\boldsymbol{F}_n(i) \cdot \boldsymbol{w}_n(i) = \boldsymbol{y}_n(i) \tag{33}$$

where $\boldsymbol{\beta}_{n-1}(i)$ is the vector of the last $n-1$ elements of the first column of the previous interpolation matrix $\boldsymbol{G}_n(i)$, i.e., $\boldsymbol{\beta}_{n-1}(i) \triangleq [\boldsymbol{g}_{n,i}(\boldsymbol{z}(i-n+1))]_{2:n}$. This time, the objective is to find $\Delta\boldsymbol{w}_{n,2:n}(i)$ and $w_{n,n}(i+1)$ satisfying

$$\begin{bmatrix} \boldsymbol{F}_{n-1}(i) & \boldsymbol{\gamma}_{n-1}(i+1) \\ \boldsymbol{\gamma}_{n-1}^\top(i+1) & \lambda_n(i+1)+K(0) \end{bmatrix} \cdot \left(\begin{bmatrix} \boldsymbol{w}_{n,2:n}(i) \\ 0 \end{bmatrix} + \begin{bmatrix} \Delta\boldsymbol{w}_{n,2:n}(i) \\ w_{n,n}(i+1) \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{y}_{n,2:n}(i) \\ y(i+1) \end{bmatrix}$$
$$\boldsymbol{F}_n(i+1) \cdot \boldsymbol{w}_n(i+1) = \boldsymbol{y}_n(i+1). \tag{34}$$

In other words, the new weight vector for the updated network can be considered the result of

i) shifting the last $n-1$ weights in the old weight vector $\boldsymbol{w}_n(i)$ which are associated with the most recent $n-1$ data in $t_n(i)$ upwards into positions 1 to $n-1$ and setting the nth element to zero;
ii) adding a perturbation $\Delta\boldsymbol{w}_{n,2:n}(i)$ to the shifted vector
iii) adding a new weight $w_{n,n}(i+1)$ in the nth position.

It is not difficult to show that the resultant update equations become

$$w_{n,n}(i+1) = \frac{y(i+1) - (\boldsymbol{w}_{n,2:n}(i) + \Delta\boldsymbol{w}_{n,2:n}(i))^\top \boldsymbol{\gamma}_{n-1}(i+1)}{\lambda_n(i+1)+K(0)} \tag{35}$$

$$\Delta\boldsymbol{w}_{n,2:n}(i) = \left(\boldsymbol{F}_{n-1}(i) - \frac{\boldsymbol{\gamma}_{n-1}(i+1)\boldsymbol{\gamma}_{n-1}^\top(i+1)}{\lambda_n(i+1)+K(0)}\right)^{-1} \cdot \left[w_{n,1}(i)\boldsymbol{\beta}_{n-1}(i) - \frac{\boldsymbol{\gamma}_{n-1}(i+1)}{\lambda_n(i+1)+K(0)} \cdot (y(i+1) - \boldsymbol{w}_{n,2:n}^\top(i)\boldsymbol{\gamma}_{n-1}(i+1)) \right]. \tag{36}$$

TABLE III
REGULARIZED RBFN FIXED-SIZE PREDICTION UPDATE ALGORITHM

Initialization: assume the regularized RBFN has been generated from $t_n(i)$ in the usual way, i.e., via equations (6) to (9) with $t_n(i)$ in place of T_n, and assume that $\boldsymbol{F}_{n-1}^{-1}(i)$ is known.

Updating: when the new datum $(\boldsymbol{z}(i+1), y(i+1))$ becomes available,

1. select the new regularization parameter $\lambda_n(i+1)$ and the norm weighting matrix $\boldsymbol{U}_n(i+1)$, typically from $t_n(i+1)$.
2. compute the new basis function vector $\boldsymbol{\gamma}_{n-1}(i+1)$.
3. compute $\left(\boldsymbol{F}_{n-1}(i) - \frac{\gamma_{n-1}(i+1)\boldsymbol{\gamma}_{n-1}^{\top}(i+1)}{\lambda_{n+1}(i+1)+K(0)}\right)^{-1}$. Complexity can be reduced to $\mathcal{O}(n^2)$ if $\boldsymbol{F}_{n-1}^{-1}(i)$ is optionally propagated from time step to time step as in step 3 of the Updating procedure in Table 2.
4. compute the shifted weight change vector $\Delta\boldsymbol{w}_{n,2:n}(i)$ according to (36).
5. compute the new weight $w_{n,n}(i+1)$ via (36).
6. delete the basis function $K\left(\|\bullet - \boldsymbol{z}(i-n+1)\|_{U_n(i)}\right)$ and its weight $w_{n,1}$ associated with the oldest data in $t_n(i)$ from the network.
7. add the shifted weight change vector $\Delta\boldsymbol{w}_n(i)$ to the remaining $n-1$ network weights.
8. add the new basis function $K\left(\|\bullet - \boldsymbol{z}(i+1)\|_{U_n(i+1)}\right)$ with weight $w_{n,n}(i+1)$ to the network
9. (optional) compute $\boldsymbol{F}_n^{-1}(i+1)$ from $\boldsymbol{F}_{n-1}^{-1}(i)$ with complexity $\mathcal{O}(n^2)$ via a partitioned matrix inverse formula applied to the decomposition (33). Hence compute $\boldsymbol{F}_{n-1}^{-1}(i+1)$ with complexity $\mathcal{O}(n^2)$ via

$$\boldsymbol{F}_{n-1}^{-1}(i+1) = \left(\boldsymbol{I} - \boldsymbol{h}_{n-1}(i+1)\boldsymbol{\beta}_{n-1}^{\top}(i+1)\right)^{-1}\boldsymbol{H}_{n-1}(i+1) \tag{81}$$

where $\boldsymbol{h}_{n-1}(i+1)$ is the vector formed from the last $n-1$ elements of the first column of $\boldsymbol{F}_n^{-1}(i+1)$ and $\boldsymbol{H}_{n-1}(i+1)$ is the lower right $(n-1)\times(n-1)$ submatrix of $\boldsymbol{F}_n^{-1}(i+1)$.

Prediction: for the NLAR case $y(i) \triangleq x(i)$ and $\boldsymbol{z}(i) \triangleq \boldsymbol{x}_d(i-1)$, set $\tilde{x}(i+2) = \tilde{f}_{n,i+1}(\boldsymbol{x}_d(i+1))$.

Iteration: $i \to i+1$ and repeat from Updating step.

Except for the additional term $w_{n,1}(i)\boldsymbol{\beta}_{n-1}(i)$ in (36), the forms of the update equations for this fixed-size case are identical to those for the augmented case. The additional term can be regarded as embodying the effect of weight vector augmentation from size n to $n+1$ followed by truncation to the weights computed from the most recent n training data. We summarize the prediction update algorithm for the fixed-size case in Table III. Note that the formula (81) in updating step 9 follows the identity in (37), shown at the bottom of the page.

Although the parallels between the recursive update algorithms described here and those in the RLS algorithm are interesting in their own right, we must be careful not to conclude that the algorithms presented are merely expressions of the RLS algorithm after a nonlinear mapping $\boldsymbol{z}(i) \in \mathbb{R}^p \mapsto \boldsymbol{g}_{n,i-1}(\boldsymbol{z}(i)) \in \mathbb{R}^n$. We can see this difference clearly in the fact that infinite memory regularized RBFN's require an infinite number of weights/basis functions; fixed-size regularized RBFN's can only have a finite memory of the same size. This condition stands in contrast to the situation with the RLS filter where a fixed number of weights are updated to reflect all the past history of the input data. Of course, the exponentially weighted variant of the RLS algorithm is commonly used in practice, and we can argue that its memory is, for all practical purposes, limited. Indeed, the introduction of the exponentially weighted variant of the RLS algorithm was motivated by the heuristic that decaying memory would improve estimation when the input/output processes are nonstationary, although it has now been established that this notion is, in fact, generally incorrect [29]. In this respect, the fixed-size regularized RBFN is somewhat more explicit in the way it deals with nonstationarity.

With both the augmented and fixed-size update algorithms, their computational efficiency is derived from the low rank of the perturbation applied to the existing interpolation matrix at a given time step through augmentation and addition, respectively. Exploiting the matrix inversion lemma can then reduce the update complexity to $\mathcal{O}(n^2)$ (for n basis functions) per time step. As may be expected, the experimental results for speech prediction show that these *partial update* algorithms can result in loss of tracking and degraded performance compared with a *full update* algorithm in which the the bandwidth and/or regularization parameter is updated for *all* entries of the regularized interpolation matrix $\boldsymbol{F}_n(i)$ and not just those involving the new basis function vectors $\boldsymbol{\gamma}_n(i+1)$ (in the case of the augmented updates) and $\boldsymbol{\gamma}_{n-1}(i+1)$ (in the case of the fixed-size updates). The update complexity per time step in this full update case is naturally greater at $\mathcal{O}(n^3)$ compared with the partial update case. Nevertheless, the recursive update algorithms for both cases provide useful insight into the essential character and operation of the dynamic regularized RBFN as a time series estimator.

V. APPLICATION TO SPEECH PREDICTION

For a benchmark problem with real-world data, we turn to speech prediction. That the human speech signal is generally nonlinear and nonstationary is well-known; even so, the linear prediction of speech with *analytic* methods such as the LMS/RLS/Kalman algorithms [27] and *synthetic* methods such as CELP [30] has been met with surprising success. Of course, these results are achieved after significant prior knowledge regarding the characteristics of human speech have been carefully embedded into the corresponding methods to realize maximum performance. In contrast, we should emphasize that our interest in speech as the test signal for the proposed algorithms is limited to the characterization of the gains possible from nonlinear and nonstationary processing and should not be taken to imply that the proposed predictors (in their current form) are either practical or optimally tuned for actual speech prediction applications such as speech coding. Further, speech-specific research and evaluation would clearly be necessary to reach that state. That said, the results of the following experiments in which both the partial and full update algorithms for the fixed-size network case are evaluated (albeit with different motivations) do offer evidence of the performance gains possible when the nonlinearity and nonstationarity of speech signals are addressed.

$$\begin{bmatrix} 1 & \boldsymbol{0}^{\top} \\ \boldsymbol{0} & \boldsymbol{F}_{n-1}^{-1}(i+1) \end{bmatrix} = \boldsymbol{F}_n^{-1}(i+1)\begin{bmatrix} \lambda_n(i-n+2)+K(0) & \boldsymbol{\beta}_n^{\top}(i+1)\boldsymbol{F}_{n-1}^{-1}(i+1) \\ \boldsymbol{\beta}_n(i+1) & \boldsymbol{I} \end{bmatrix} \tag{37}$$

A. Experiment 1: Partial Update Algorithm for Fixed-Sized Networks

We begin by giving some results for the fixed-sized update algorithm of Table III. At this stage of development, we focus our attention on the practical issues of predictor tracking stability and performance versus the fixed-size full update algorithm.

1) Description of Speech Data: We use a 10 000-point speech sample of a male voice recorded at 8 kHz and 8 b/sample while speaking the sentence fragment "When recording audio data ...." The speech data, which appear to have no discernible noise, are approximately zero-mean and normalized to unit total amplitude range. Applying the Mann–Whitney rank-sum test as described in Section V-B1 rejects the null hypothesis that the speech sample is that of a stationary linear process with a maximum sample Z statistic of less than -13 (a Z statistic of less than -3 is considered grounds for strong rejection), hence indicating a high probability of nonlinearity in the speech sample.

2) Approach Using Regularized RBFN's: In the main, we follow same the approach as in the full-update case discussed in Section V-B2, except for the following modifications.

Input Order: A common input order of $p = 50$ is used for each network (unless otherwise indicated).

Regularization Parameter: For a given network, fixed for the duration of prediction over the input signal, i.e., $\lambda_n(i+1) = \lambda_n(i)$ for all i.

Update Algorithm: Except during reset (see the following), we follow Table III, where the updated norm weighting matrix $\boldsymbol{U}_n(i)$ is computed according to the input data covariance formula described in the corresponding section below for the full update case. The updated norm weighting matrix, however, is applied only to the new basis functions in the updated column $\boldsymbol{\gamma}_{n-1}(i+1)$ in (34) to maintain consistency with the usual SI fitting relation $\tilde{\boldsymbol{y}}_n(i+1) = \boldsymbol{G}_n(i+1)\boldsymbol{w}_n(i+1)$, where $\tilde{\boldsymbol{y}}_n(i+1)$ is the estimate of $\boldsymbol{y}_n(i+1)$ produced by the network at time step $i+1$.

Reset Algorithm: As can be seen, the partial updating algorithm implies that the networks produced no longer exactly solve the interpolation problem (12) [since with partial updates the interpolation matrix $\boldsymbol{G}_n(i)$ is not identical to the one specified by the interpolation problem over $t_n(i)$]. The accumulation of these partial updates to the interpolation matrix over many consecutive time steps can lead to a loss of tracking and instability. To counteract this problem, we monitor the prediction error $\epsilon_n(i+1)$ of the dynamic network at each time step i and *reset* the network, i.e., restart the partial update algorithm from the initialization step 1 of Table III, when one of two possible conditions, denoted (RC.1) and (RC.2), are met:

$$|\epsilon_n(i+1) - \mu(\epsilon_n(i), m)| > \kappa\sigma(\epsilon_n(i), m) \qquad \text{(RC.1)}$$

$$|\epsilon_n(i+1)| - \mu(|\epsilon_n(i)|, m) > \kappa\sigma(|\epsilon_n(i)|, m) \qquad \text{(RC.2)}$$

where for a sequence $\{a(i)\}$, $\mu(a(i), m)$ is the sample mean, and $\sigma(a(i), m)$ is the sample standard deviation of $\{a(j)\}_{j=i-m+1}^{i}$. Thus, a predictor reset occurs when a probable large deviation (as set by the *window* parameter m and *threshold* parameter κ) has occurred in either the two-sided (RC.1) or one-sided (RC.2) prediction error. For our experiment, we use reset condition (RC.1) with window $m = n = 100$ and threshold $\kappa = 4$ as there appears to be no substantial difference in performance compared with condition (RC.2). In the ideal case that prediction error is a white Gaussian process, the choice of κ corresponds to a large deviation probability of approximately 0.0063%. Not unexpectedly, the actual reset rate in the experiment is quite a bit greater due to heavy tails in the prediction error density.

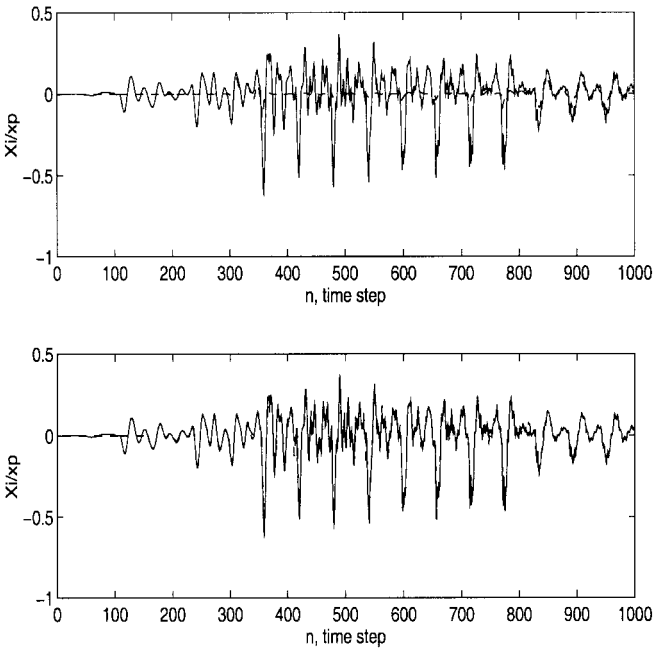


Fig. 1. Tracking ability of static (upper) versus dynamic (lower) predictors (solid is actual, dashed is predicted).

These design decisions yield networks with moderate computational complexity and reasonable performance that suit the basic purpose of demonstrating the partial update algorithm for fixed-size networks. Further optimization of the design choices with their concomitant increased computational load are no doubt possible but will not be pursued here.

3) Dynamic Updating and Regularization for Speech Prediction: Using Figs. 1 and 2, we can briefly argue for the practical utility of dynamic updating and regularization for speech prediction. In the former figure, we compare the initial predictions of a dynamic predictor trained according to the partial update algorithm (without reset) for a $n = 100$, $\lambda = 0.01$ fixed-size network with those from a static $n = 250$, $\lambda = 0.01$ predictor whose network parameters are frozen after the initial training. Not surprisingly, even with more than twice the number of basis functions, the static predictor quickly loses track of speech signal in transition from a quickly to a slowly varying portion of the input signal, as shown in the figure. The dynamic predictor, however, is able to adapt and maintain its prediction performance. Regarding regularization, although RLSF theory implies that $\lambda = 0$ is a consistent choice when no noise is present, in practice, some regularization is necessary because the likelihood of a singular/ill-conditioned interpolation matrix $\boldsymbol{G}_n(i)$ increases

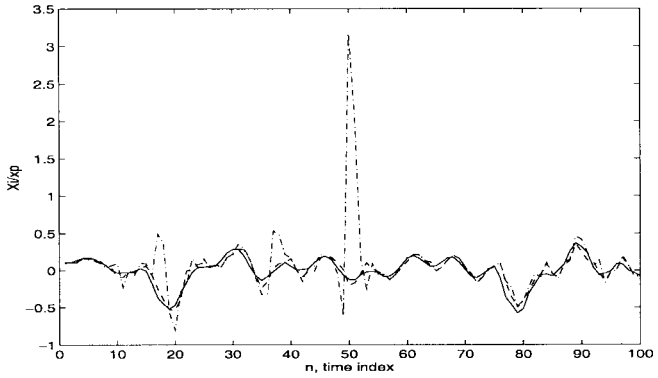


Fig. 2. Regularized versus nonregularized predictors, $n = 100, p = 2$ (solid is actual, dashed is predicted for $\lambda = 0.1$, dash-dot is predicted for $\lambda = 0.01$).

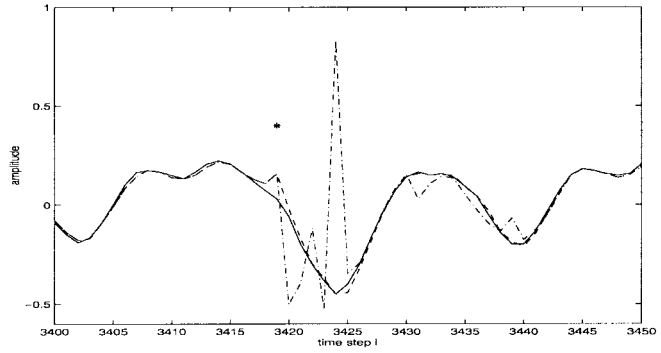


Fig. 3. Partial update algorithm, fixed-size case with reset versus without reset (solid is actual, dashed is predicted with reset, dash-dot is predicted without reset, star indicates reset point).

as n increases. Empirically, this effect appears especially pronounced for small values of p in which the predictor output is more sensitive to individual inputs in the input vector. An example of this phenomenon can be seen in Fig. 2, where we contrast the predictions for two partially updated (without reset) fixed-sized predictors, one of which is trained with a fixed $\lambda = 0.1$ and the other with a fixed $\lambda = 0.01$. Again, it is evident that sufficient regularization is useful from a numerical point of view to combat instability.

4) Comparison of Partial Update Algorithm with and Without Reset: Fig. 3 gives an example of the efficacy of the reset criterion (RC.1). After detecting a relatively large deviation in the prediction error at the starred point (time step 3419), the partially updated fixed-size predictor with reset reinitializes to avoid the obvious stability problem exhibited by the same predictor without reset. Since reset is triggered at approximately 1% of all prediction time steps for the $\lambda = 0.001$ case shown, the example shown is by no means isolated, although the magnitude of tracking loss displayed is among the largest observed for that case.

5) Comparison to Full Update Algorithm: Ultimately, we would like to compare the performance of the fixed-size dynamic network algorithm using partial updating and reset [according to RC.1)] to the same with full updating. As the performance measure, we use the PSNR as described for the full update case in Section V-B.3. Table IV shows that the overall performance loss for the networks using partial

TABLE IV
PREDICTION PERFORMANCE OF FIXED-SIZE ALGORITHM WITH PARTIAL UPDATE VERSUS FULL UPDATE

RBF reg. parameter	% of pred. resets	PSNR (dB)		
		partial update	full update	partial − full
$\lambda = 0.001$	0.94	14.37	14.73	−0.36
$\lambda = 0.0001$	0.92	13.71	14.45	−0.74
$\lambda = 0.00001$	0.87	13.91	14.19	−0.28

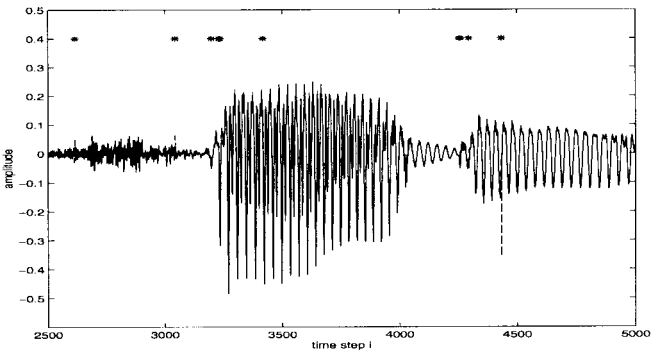


Fig. 4. Reset points in second 2500 predicted points for partial update algorithm, fixed-size case (solid is actual, dashed is predicted, stars indicate reset positions).

updating compared with those using full updating varies from a relatively minimal 0.28 dB for the $\lambda = 0.000\,01$ predictor to a more substantial 0.74 dB for the $\lambda = 0.0001$ predictor. From a computational standpoint, the figures for the percentage of points at which reset is triggered indicate that the partially updated fixed-size dynamic predictor has only 1% of the computational complexity of the corresponding fully updated fixed-size dynamic predictor. While this reset rate is two orders of magnitude larger than expected in the ideal case of white Gaussian prediction errors, it still easily satisfies the basic distribution-free upper bound of 1/16 = 6.25% implied by the Chebyshev inequality, viz.

$$\text{Prob}\left\{|\epsilon_n(i+1) - \mu^*(\epsilon_n(i))| > \kappa\sigma^*(\epsilon_n(i))\right\} < \frac{1}{\kappa^2}$$

where $\mu^*(\cdot)$ and $\sigma^*(\cdot)$ are the true, i.e., distributional, mean and standard deviation of a process $\cdot$.

Fig. 4 shows the points of reset for the $\lambda = 0.001$ partially updated fixed-size dynamic predictor from time steps 2500–5000. It is interesting to note how in this segment the predictor resets occur at points of a regime shift within the speech sample. Because the performance/computational tradeoff between the two update techniques is influenced by several factors such as the length of prediction, the speech segment being predicted, etc., further characterization is necessary to make more definitive statements; nonetheless, the results can be considered encouraging.

6) Comparison to Previous Work: The same speech signal was also used as part of two previous studies, both of which are based on *pipelined recurrent neural networks (PRNN's)* followed by standard linear adaptive filters [31], [32]. PRNN's represent another method of modeling nonstationary dynamics based on the use of explicit feedback between modular network elements, each of which is itself a (recurrent) neural network. While considerably different in the details of their

architectures and training methods, they do share the common principle of continuously adapting their network parameters to minimize their squared prediction error and, thus, track nonstationary signal characteristics. Comparing our results in Table IV with those of [32, Table III], we see that even our worst-performing case of partial updating yields a PSNR of 13.71 dB, which is 0.12 dB better than the best PSNR of 13.59 dB listed in Table II for a hybrid extended RLS (ERLS)-trained PRNN followed by a 12th-order RLS filter. To be fair, however, the $\mathcal{O}(n^2)$ computational complexity of our partial updating method with $n = 100$ centers is most likely somewhat greater than that of the ERLS PRNN with three eight-input single-neuron modules used in [32]. On the other hand, their predictor has the benefit of an additional level of RLS linear prediction not (yet) present in our scheme, and their performance figures are reported using the variance rather than mean-squared value of the prediction error, both which should bias their results upwards compared with our PSNR figures (see the discussion in Section V-B3). Of course, it would be premature to draw any substantial conclusions on the basis of a single speech signal, which leads us to consider the more comprehensive suite of longer, phonetically balanced male and female speech signals in the next experiment. Over this test suite, we shall also show the performance increase possible from employing a similar final level of RLS linear prediction.

B. Experiment 2: Full Update Algorithm for Fixed-Size Networks

As we previously mentioned, our objective in this set of experiments is to demonstrate that even without significant tuning, the dynamic regularized RBFN can provide a nontrivial improvement in prediction SNR over the standard LMS/RLS algorithm-based predictors. We also indicate the further improvement possible in exploiting the residual correlations between the predictions of several dynamic regularized RBFN's and the predicted, i.e., desired, speech signal by way of an additional stage of RLS estimation.

1) Description of Speech Data: The speech data to be predicted consist of samples from ten different male and ten different female speakers, each reading a distinct phonetically balanced sentence. In their original format, the continuous speech signals were 16-b linear PCM and sampled at 16 kHz rate with 8 kHz bandwidth. These samples were subsequently filtered by a third-order Butterworth filter with a cutoff frequency of 3.2 kHz, decimated to 8 kHz rate, and recentered to zero-mean. Both the original and final speech signals are of high quality with little discernible background noise. The sentence samples and some of their key characteristics as discrete time series are summarized in Tables V and VI. As can be seen from these tables, the total length of a speech signal being tested varies from approximately 2.5–4 s.

Before beginning, it is useful to quantify the degree of nonlinearity in the speech samples, as this factor will ultimately determine the gains possible in our approach. Using some software for chaotic time series analysis developed by the chemical reactor engineering group at Delft University of Technology in the Netherlands [33], the method of surrogate data analysis [34] with a Mann–Whitney rank-sum test rejects the null hypothesis that each speech sample is that of a stationary linear process with a maximum sample Z statistic of less than -13 in each case (a Z statistic of less than -3 is considered grounds for strong rejection). This result indicates that significant benefit from nonlinear processing should be possible.

TABLE V
MALE SPEECH SAMPLE PARAMETERS

ID	No. of samples	Sentence
m130	20215	Type out three lists of orders.
m131	23525	The harder he tried, the less he got done.
m132	25128	The boss ran the show with a watchful eye.
m133	25129	The cup cracked and spilled its contents.
m134	23260	Paste can cleanse the most dirty brass.
m135	29073	The slang word for raw whiskey is booze.
m136	27746	It caught its hind paw in a rusty trap.
m137	26724	The wharf could be seen at the farthest shore.
m138	22435	Feel the heat of the weak dying flame?
m139	20877	The tiny girl took off her hat.

TABLE VI
FEMALE SPEECH SAMPLES PARAMETERS

ID	No. of samples	Sentence
f150	21530	The young kid jumped the rusty gate.
f151	25471	Guess the results from the first scores.
f152	25064	A salt pickle taste fine with ham.
f153	24674	The just claim got the right verdict.
f154	24361	These thistles bend in a high wind.
f155	22098	Pure bred poodles have curls.
f156	26954	The tree top waved in a graceful way.
f157	32522	The spot on the blotter was made by green ink.
f158	27220	Mud was spattered on the front of his white shirt.
f159	23306	The cigar burned a hole in the desk top.

2) Approach Using Regularized RBFN's: The particular approach taken is to treat each speech sample as a realization of a discrete-time Markov process of order p obeying (26). For one-step-ahead (1-SA) prediction $k = 1$, we consider the limiting case of per time step updating, i.e., $l = 1$. Key design issues to consider are the following.

Input Order: Preliminary experiments showed that for a given speech sample, the prediction performance of the dynamic regularized RBFN varied with the order p, depending on the local characteristics of the speech over which the network was operating. For example, in the transition periods between voiced, unvoiced, and silent segments, networks with small p, e.g., $p = 10$, were generally found to perform better than those with large p, e.g., $p = 50$. Conversely, within a given type of speech segment, the networks with larger p tended to be the better predictors. While techniques for estimating the order of NLAR processes have been recently proposed [35], for computational simplicity, three fixed-sized networks with $p = 10$, 30, and 50 are run in parallel for each speech sample and, as we shall see later, linearly combined.

RBFN Parameters: Based on some previous work [36], each of the networks is chosen to have the following.

network size: A fixed-size of $n = 100$ basis functions is used. This fixed-size corresponds to the assumption that a useful memory for the networks is 12.5 ms, which is the average length of the 5–20-ms window of stationarity usually associated with speech.

basis function: The "smooth" [in the sense of satisfying (12)] Gaussian basis function $K(r) = \exp(-r^2/2)$ is used.

norm weighting matrix: Common to all basis functions is

TABLE VII
GCV CRITERION FUNCTION EVALUATION LIMITS

Network no.	p	λ_{min}	λ_{max}
1	50	0.0001	0.001
2	30	0.00001	0.01
3	10	0.0001	0.01

a diagonal norm weighting matrix $\boldsymbol{U}_n(i)$ whose inverse, at time step i, is set to p times the diagonal of the empirical covariance matrix for the input samples in $t_n(i)$. This particular form of the norm weighting matrix allows the multidimensional network basis function to be decomposed into a p-fold product of one-dimensional (1-D) (Gaussian) kernels, each with bandwidth parameter equal to the variance of a particular window over $t_n(i)$. In the 1-D i.i.d. density estimation setting, such a form of bandwidth has been shown to be consistent in the L_1 sense [37].

Regularization Parameter: For each of the three networks ($p = 10$, 30, and 50), the regularization parameter for each time step is selected as the value that minimizes the GCV criterion function evaluated over 1000 logarithmically spaced points from $\lambda_{\min}$ to $\lambda_{\max}$ for that network as given in Table VII. Since the speech signals are largely noise free, the upper bound on $\lambda_n(i+1)$ prevents undue over-regularization, whereas the lower bound is necessary to ensure the numerical nonsingularity of the regularized SI matrix at each time step. The slight differences in the evaluation limits account for the varying degrees of sensitivity of each network to these two conditions.

Update Algorithm: Because the norm weighting matrix $\boldsymbol{U}_n(i)$ is updated for *all* network basis functions when new data arrive, the update from $\boldsymbol{F}_n(i)$ to $\boldsymbol{F}_n(i+1)$ is full rank, and hence, (34) must be solved directly without using the recursion aids (35) and (36). It was found in previous experiments [36] that the speech samples were sufficiently nonstationary so that without careful choice of the update parameters indicated in first Updating step of each algorithm, the recursively updated fixed-size network outputs would frequently loose track of the speech samples within an order of n time steps from the last full-rank update. Notwithstanding the results of the previous section, the issue of how best to select the update parameters in the recursive fixed-size update algorithm to minimize performance loss from partial updating remains an open question.

3) Comparison to Linear RLS Algorithm and Previous Work: The performance measure we shall use is the *predicted signal-to-noise ratio (PSNR)* defined for an actual or input signal sequence $\{y(i)\}_{i=1}^{N}$ by

$$\text{PSNR (dB)} \triangleq 10\log_{10}(\tilde{\sigma_y^2}/\tilde{\sigma_\epsilon^2}) \tag{38}$$

where $\tilde{\sigma_y^2}$ and $\tilde{\sigma_\epsilon^2}$ are the actual and error signal *powers* estimated by

$$\tilde{\sigma_y^2} \triangleq \frac{1}{N}\sum_{i=1}^{N} y^2(i)$$

$$\tilde{\sigma_\epsilon^2} \triangleq \frac{1}{N}\sum_{i=1}^{N} \epsilon^2(i) \quad \text{where} \quad \epsilon(i) \triangleq y(i) - \tilde{y}(i) \tag{39}$$

and $\tilde{y}(i)$ is the network prediction for actual signal $y(i)$. The PSNR can be considered a measure of the *generalization* performance of the dynamic network since in our NLAR case, each prediction $\tilde{y}(i) = \tilde{x}(i+1)$ at time step i is for the first time series point *outside* the window $\{x(j)\}_{j=i-(n+p-1)}^{i}$ of data effectively seen during training [$n+p$ sequential data are needed to form $t_n(i)$]. This effective training window, along with the predicted point, shift forward in time as the dynamic network advances through the entire input signal sequence. Although, strictly speaking, the test set per time step is a single (out-of-sample) point, by iterating the training/prediction cycle over the available input time series (the number of samples in each speech signal listed in Tables V and VI less $n+p$ samples for initialization), this pointwise prediction performance can be averaged to gauge the generality of our method. For example, the PSNR figure in Table VIII for network 1 operating on signal m130 is computed according to (38) and (39) with $N = 20\,215 - 100 - 50 = 20\,065$ effective test data.

TABLE VIII
OVERALL EXPERIMENTAL RESULTS FOR SPEECH PREDICTION, SAMPLES m130–m134 (ALL PSNR IN dB)

Network no.	m130	m131	m132	m133	m134
1	14.02	13.76	15.30	9.93	12.79
2	14.91	13.83	15.53	9.91	12.51
3	14.04	13.49	15.24	11.27	13.40
NL avg.	14.32	13.69	15.36	10.37	12.90
RLS (auto)	11.76 (14, 0.99)	11.77 (50, 0.999)	14.58 (50, 0.99)	10.85 (50, 0.999)	11.83 (50, 0.999)
RLS (NL)	16.07 (1, 0.98)	15.09 (1, 0.99)	17.08 (1, 0.98)	12.36 (1, 0.99)	14.75 (1, 0.999)
RLS (NL+auto)	16.43 (1+6, 0.99)	15.58 (4+14, 0.999)	17.73 (1+10, 0.99)	13.14 (3+6, 0.999)	15.31 (3+14, 0.999)

TABLE IX
OVERALL EXPERIMENTAL RESULTS FOR SPEECH PREDICTION EXAMPLE, SAMPLES m135–m139 (ALL PSNR IN DECIBELS)

Network no.	m135	m136	m137	m138	m139
1	17.37	12.22	15.41	15.09	15.41
2	17.30	12.81	15.43	15.13	17.30
3	16.69	13.22	15.74	15.52	16.69
NL avg.	17.12	12.75	15.53	15.25	15.53
RLS (auto)	15.48 (46, 0.99)	10.92 (10, 0.99)	13.24 (50, 0.99)	12.48 (50, 0.999)	13.43 (50, 0.999)
RLS (NL)	18.63 (1, 0.99)	14.42 (1, 0.99)	17.05 (1, 0.999)	16.92 (1, 0.99)	16.20 (1, 0.999)
RLS (NL+auto)	19.35 (1+6, 0.99)	14.75 (1+4, 0.99)	17.42 (2+6, 0.999)	17.11 (5+10, 0.999)	16.86 (1+14, 0.999)

Note that in our case of zero-mean input signals, because we use estimated signal powers rather than estimated signal variances, as is sometimes the case in defining the PSNR, the following performance figures are somewhat conservative (for example, a nonzero mean level of error will degrade performance by the former definition but not by the latter definition). That said, the PSNR results of the three RBFN predictors individually and jointly (as will be explained) over the complete speech samples can be found in Tables VIII–XII. Summary tables of minimum, average, and maximum performance gains are listed in Tables X, XIII, and XIV for the male only, female only, and joint male/female samples, respectively. The first four lines of each table list the individual predictor performances along with their arithmetic average. We see an average gain of 1.65 dB of the basic regularized RBFN predictors over the RLS predictor for the male speech samples while the average gain for the female speech samples is somewhat better at 2.67 dB. Over both the male and female speech samples, the average gain is 2.2 dB. The

TABLE X
SUMMARY OF GAINS IN EXPERIMENTAL RESULTS FOR SPEECH PREDICTION, MALE SPEECH SAMPLES (ALL FIGURES IN DECIBELS)

Gain	min.	avg.	max.
NL avg. over RLS(auto)	−0.48	1.65	2.77
RLS(NL) over NL avg.	0.67	1.58	1.99
RLS(NL+auto) over RLS(NL)	0.19	0.51	0.78
Total over RLS(auto)	2.29	3.73	4.67

TABLE XI
OVERALL EXPERIMENTAL RESULTS FOR SPEECH PREDICTION, SAMPLES f150–f154 (ALL PSNR IN DECIBELS)

Network no.	f150	f151	f152	f153	f154
1	15.29	15.35	14.72	13.76	15.49
2	15.58	15.67	14.91	12.90	15.07
3	14.46	15.55	14.55	13.78	15.15
NL avg.	15.11	15.52	15.36	13.48	15.24
RLS (auto)	11.05 (50, 0.999)	13.10 (44, 0.99)	12.29 (44, 0.99)	9.869 (50, 0.999)	13.68 (46, 0.99)
RLS (NL)	16.74 (3, 0.999)	17.37 (1, 0.99)	16.42 (1, 0.98)	15.37 (3, 0.999)	16.93 (1, 0.99)
RLS (NL+auto)	16.93 (4+6, 0.999)	17.43 (1+4, 0.99)	16.47 (2+6, 0.999)	15.50 (4+8, 0.999)	17.16 (2+48, 0.999)

TABLE XII
OVERALL EXPERIMENTAL RESULTS FOR SPEECH PREDICTION EXAMPLE, SAMPLES f155–f159 (ALL PSNR IN DECIBELS)

Network no.	f155	f156	f157	f158	f159
1	18.02	15.60	15.42	11.74	15.38
2	18.14	15.97	15.95	12.40	14.39
3	17.78	15.89	16.19	13.06	16.15
NL avg.	17.98	15.82	15.85	12.40	15.31
RLS (auto)	16.08 (42, 0.99)	11.85 (50, 0.999)	13.15 (50, 0.999)	10.09 (50, 0.999)	14.26 (50, 0.999)
RLS (NL)	19.86 (1, 0.99)	17.45 (3, 0.999)	17.51 (1, 0.99)	13.96 (1, 0.999)	17.48 (1, 0.999)
RLS (NL+auto)	20.60 (1+4, 0.99)	17.70 (4+8, 0.999)	17.84 (4+32, 0.999)	14.29 (3+8, 0.999)	17.83 (3+8, 0.999)

TABLE XIII
SUMMARY OF GAINS IN EXPERIMENTAL RESULTS FOR SPEECH PREDICTION, FEMALE SPEECH SAMPLES (ALL FIGURES IN DECIBELS)

Gain	min.	avg.	max.
NL avg. over RLS(auto)	1.05	2.67	4.06
RLS(NL) over NL avg.	1.06	1.70	2.17
RLS(NL+auto) over RLS(NL)	0.05	0.27	0.74
Total over RLS(auto)	3.48	4.63	5.88

TABLE XIV
SUMMARY OF GAINS IN EXPERIMENTAL RESULTS FOR SPEECH PREDICTION, MALE AND FEMALE SPEECH SAMPLES (ALL FIGURES IN DECIBELS)

Gain	min.	avg.	max.
NL avg. over RLS(auto)	−0.48	2.16	4.06
RLS(NL) over NL avg.	0.67	1.64	2.17
RLS(NL+auto) over RLS(NL)	0.05	0.39	0.78
Total over RLS(auto)	2.29	4.18	5.88

RLS predictor performance reported in the fifth line (with the corresponding autoregressive input order and exponential weight in parentheses) is the best one observed in a series of experiments for which the parameters vary as in Table XV. To allow a fair assessment of the gains possible from nonlinear versus linear prediction, the maximum order p of the linear predictor is set to 50, which is the same as for the RBFN. With regards to nonlinear speech predictors, these figures are in general agreement with those in previously published work [32], [38], [39]. In particular, [38] reported an increase in prediction gain of 2.8 dB when a nonlinear predictor is trained on the residuals of a time-varying LPC predictor, which may be considered a *linear-nonlinear* processing scheme.

TABLE XV
TRIAL PARAMETERS FOR REFERENCE ADAPTIVE LINEAR PREDICTOR (a:h:b DENOTES SEQUENCE FROM a TO b INCLUSIVE SAMPLED AT h, $\boldsymbol{P}(0)$ IS INITIAL INVERSE OF INPUT CORRELATION MATRIX, ρ IS EXPONENTIAL WEIGHT)

RLS parameter	Trial range/setting
$\boldsymbol{P}(0)$	$100\boldsymbol{I}$
$1-\rho$	0, 10^{-6}, 10^{-4}, 10^{-3}, 0.01:0.01:0.20
p	2:2:50

As previously mentioned in Section V-A.6, [31] and [32] considered enhancing their nonlinear predictor performance by including a final stage of adaptive linear prediction, with the latter work showing gains of between 1.6 and 2.0 dB over the final linear stage by itself. We follow this point of view to improve our nonlinear predictor performance by linearly combining the three predictor outputs, resulting in the *nonlinear-linear* processing scheme described below.

4) Linearly Combining Predictor Outputs for Improved Performance: During the course of the experiment, we noted that the error sequences produced by an ensemble of nonlinear predictor outputs trained on a given speech sample with different parameters exhibit some residual correlation with the desired prediction. This observation suggested that by standard properties of least-squares estimators, some further improvement in prediction performance should be possible when the predictor outputs are used as inputs in an additional level of regression on the desired (actual) speech signal. In selecting a compatible structure for this subsequent processing, it was desirable to retain as much as possible the recursive on-line nature of the algorithm without significantly increasing the computational burden. Thus, the sixth line of the overall result tables shows the best observed performance for each speech sample when the three RBFN predictor outputs $\tilde{Y}_1(i)$, $\tilde{Y}_2(i)$, and $\tilde{Y}_3(i)$ at each time step i are formed into 3-tuples $\tilde{\boldsymbol{Y}}(i) = [\tilde{Y}_1(i), \tilde{Y}_2(i), \tilde{Y}_3(i)]^\top$ and taken as regressive vector inputs into another exponentially-weighted RLS predictor or *linear combiner* (to avoid confusion with the reference adaptive linear predictor). As before, the regressive orders and weights of the best such RLS linear combiners are given in parentheses following their performance figures and are chosen from trials conducted over the parameter ranges specified in Table XVI. In most cases, only the most recent RBFN predictor outputs are necessary to provide a further nontrivial performance gain averaging 1.64 dB over both the male and female speech samples. Augmenting the RBFN predictor output 3-tuples with autoregressive inputs drawn directly from the speech samples gives an additional small improvement of 0.51 dB for the male speech samples and 0.27 dB for the female speech samples, on average, for the best observed linear combiners. The exact performance figures for this nonlinear-linear input configuration are given in the seventh line of the tables, where the notation in parentheses is (*nonlinear 3-tuple order* + *linear autoregressive order, RLS weight*). Table XVII lists the trial parameter ranges in this final configuration for which the average performance gain over the RLS predictor for both male and female speech samples is 4.18 dB. We note that this average performance gain is approximately 2 dB greater than that reported in the relevant rows of [32, Tables II–IV], although that study was limited to three speech signals. This

TABLE XVI
TRIAL PARAMETERS FOR RLS LINEAR COMBINER ON RBFN OUTPUTS ONLY (a:h:b DENOTES SEQUENCE FROM a TO b INCLUSIVE SAMPLED AT h, $\boldsymbol{P}(0)$ IS INITIAL INVERSE OF INPUT CORRELATION MATRIX, ρ IS EXPONENTIAL WEIGHT)

RLS parameter	Trial range/setting
$\boldsymbol{P}(0)$	$100\boldsymbol{I}$
$1-\rho$	0, 10^{-6}, 10^{-4}, 10^{-3}, 0.01, 0.02
p	1:1:6

TABLE XVII
TRIAL PARAMETERS FOR RLS LINEAR COMBINER ON BOTH RBFN OUTPUTS AND AUTOREGRESSIVE INPUTS (a:h:b DENOTES SEQUENCE FROM a TO b INCLUSIVE SAMPLED AT h, $\boldsymbol{P}(0)$ IS INITIAL INVERSE OF INTPUT CORRELATION MATRIX), ρ IS EXPONENTIAL WEIGHT)

RLS parameter	Trial range/setting
$\boldsymbol{P}(0)$	$100\boldsymbol{I}$
$1-\rho$	0, 10^{-6}, 10^{-4}, 10^{-3}, 0.01, 0.02
p_{NL}	1:1:6
p_{auto}	2:2:50

gain naturally comes at the price of increased computational complexity, namely, $\mathcal{O}(n^3)$ per time step, where n is the number of basis function, versus $\mathcal{O}(p^2)$ for the linear RLS predictor, where p is the linear autoregressive order. Whether the increased computational complexity of the regularized RBFN predictor over a linear one such as the RLS predictor is acceptable depends on the intended application, but we should note that further gains in the nonlinear predictor's performance over the linear one should (at least in principle) still be possible since not all network parameters were fully optimized, e.g., the bandwidth parameters.

VI. CONCLUSION

We have presented two theorems relating the NWRE to the regularized RBFN that justify its application to nonlinear time series prediction. In the case of certain NLAR processes, we show that minimizing the risk over the training set is asymptotically optimal in the global m.s. prediction error, thereby demonstrating the key role regularization plays in the RBFN. To deal with the nonstationarity induced by a multimodal time-varying regression function, recursive algorithms for the periodic updating of RBFN parameters have been developed for both the infinite and finite memory cases that exhibit significant resemblance to the standard RLS algorithms and allow for similar interpretations. Experiments conducted on a suite of phonetically balanced male and female speech samples demonstrate the nontrivial gains over linear techniques possible when the nonlinear processing of the regularized RBFN is applied to the one-step-ahead prediction of NLAR processes. We also describe how a simple linear combination of an ensemble of nonlinear predictor outputs via the RLS algorithm can yield further improvements in prediction performance with little added computational complexity while alleviating the difficulty of optimal model parameter estimation.

APPENDIX A
PROOF OF THEOREM 1

In the proof, the following lemma for NWRE approximation with regularized RBFN's in the deterministic case will be useful.

Lemma 1: Let $\tilde{f}_n'$ be an NWRE with radial kernel $K' = C \cdot K$, where $C \triangleq \sup_{\boldsymbol{z}\in\mathbb{R}^d} K'(\boldsymbol{z})$, and with bandwidth parameter h_n designed from a given training set t_n. Then, for any $n>1$, $\alpha > \log(C/(h_n^d \tilde{p}_n(\boldsymbol{z})))/\log n$, and $\boldsymbol{z}\in\mathbb{R}^d$ such that the denominator $nh_n^d\tilde{p}_n(\boldsymbol{z}) \triangleq \mathbf{1}_n^\top \boldsymbol{g}_n'(\boldsymbol{z})$ of $\tilde{f}_n'$ is not zero, a regularized RBFN $\tilde{f}_n$ with kernel K may be constructed such that

$$|\tilde{f}_n(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})| \leq \frac{C^2 M}{n^{\alpha/2}h_n^d\tilde{p}_n(\boldsymbol{z})[n^{\alpha/2}h_n^d\tilde{p}_n(\boldsymbol{z}) - Cn^{-\alpha/2}]} \tag{40}$$

where $||\boldsymbol{y}||_\infty \leq M$, i.e., $\boldsymbol{y}$ is element-wise bounded by M, and where $\mathbf{1}_n$ is a constant vector of n ones.

Proof: Letting $\boldsymbol{y}_n' \triangleq n^\alpha \boldsymbol{y}_n$, where $\alpha \geq 0$ is an exponent to be determined later, we may equivalently write the NWRE output as

$$\tilde{f}_n'(\boldsymbol{z}) = \boldsymbol{g}_n^\top(\boldsymbol{z})(\lambda_n(\boldsymbol{z})\boldsymbol{I})^{-1}\boldsymbol{y}_n'. \tag{41}$$

Consider the regularized RBFN (with kernel K) constructed from t_n using the scaled outputs $\boldsymbol{y}_n'$ in place of $\tilde{y}_n$ and with $\boldsymbol{U}_n \triangleq \boldsymbol{I}/h_n$, $\lambda_n = \lambda_n(\boldsymbol{z}) = n^\alpha \boldsymbol{g}_n^\top(\boldsymbol{z})\mathbf{1}_n = n^{\alpha+1}h_n^d\tilde{p}_n(\boldsymbol{z})/C$. Comparing the NWRE output to the output of this regularized RBFN, we find that the difference can be bounded (by Cauchy–Schwarz) as

$$\begin{aligned}
&|\tilde{f}_n(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})| \\
&\quad = |\langle \boldsymbol{g}_n(\boldsymbol{z}), (\boldsymbol{G}_n + \lambda_n(\boldsymbol{z})\boldsymbol{I})^{-1}\boldsymbol{y}_n' - (\lambda_n(\boldsymbol{z})\boldsymbol{I})^{-1}\boldsymbol{y}_n'\rangle| \\
&\quad \leq ||\boldsymbol{g}_n(\boldsymbol{z})||\,||(\boldsymbol{G}_n + \lambda_n(\boldsymbol{z})\boldsymbol{I})^{-1} - (\lambda_n(\boldsymbol{z})\boldsymbol{I})^{-1}||\,||\boldsymbol{y}_n'|| \\
&\quad \leq ||\boldsymbol{g}_n(\boldsymbol{z})||\frac{||\boldsymbol{G}_n||\,||\boldsymbol{I}||}{\lambda_n(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - ||\boldsymbol{G}_n||)}||\boldsymbol{y}_n'|| \\
&\qquad \lambda_n(\boldsymbol{z}) > ||\boldsymbol{G}_n||.
\end{aligned} \tag{42}$$

Using the Euclidean norm as an upper bound for all quantities except for $\boldsymbol{G}_n$, which we bound in Fröbenius norm as $||\boldsymbol{G}_n|| \leq n$, we obtain

$$\begin{aligned}
|\tilde{f}_n(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})| &\leq \sqrt{n}\frac{n}{\lambda_n(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - n)}n^\alpha\sqrt{n}M \\
&\leq \frac{n^2 n^\alpha M}{\lambda_n(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - n)}
\end{aligned}$$

which can be written for our choice of $\lambda_n(\boldsymbol{z})$ as

$$\begin{aligned}
&|\tilde{f}_n(\boldsymbol{z}) - \tilde{f}_n'(\boldsymbol{z})| \\
&\quad \leq \sqrt{n}\frac{n^{\alpha+2}C^2M}{n^{\alpha+1}h_n^d\tilde{p}_n(\boldsymbol{z})(n^{\alpha+1}h_n^d\tilde{p}_n(\boldsymbol{z}) - nC)} \\
&\quad \leq \frac{C^2M}{n^{\alpha/2}h_n^d\tilde{p}_n(\boldsymbol{z})[n^{\alpha/2}h_n^d\tilde{p}_n(\boldsymbol{z}) - Cn^{-\alpha/2}]}.
\end{aligned} \tag{43}$$

The condition on $\lambda_n(\boldsymbol{z})$ in (42) can be satisfied by choosing

$$\begin{aligned}
\lambda_n(\boldsymbol{z}) > n \Rightarrow n^{\alpha+1}h_n^d\tilde{p}_n(\boldsymbol{z})/C > n \Rightarrow \alpha \\
> \log(C/(h_n^d\tilde{p}_n(\boldsymbol{z})))/\log n.
\end{aligned} \tag{44}$$

□

Main Proof: Building on Lemma 1, we treat each case separately.

1) It is easy to show that (15) implies that by choosing N to satisfy

$$n > N \Rightarrow \sup_{\boldsymbol{z}\in D} |\tilde{p}_n(\boldsymbol{z}) - p(\boldsymbol{z})| < \frac{m}{2} \tag{45}$$

we have $n > N \Rightarrow \tilde{p}_n(\boldsymbol{z}) \geq m/2$ for all $\boldsymbol{z} \in D$. Hence, for $n > N$, we may replace $\tilde{p}_n(\boldsymbol{z})$ with $m/2$ in the denominator of the upper bound, and the term $Cn^{-\alpha/2}$ can be dominated by selecting a sufficiently large constant to multiply the numerator of the order bound, i.e., $\exists L > 0$ such that for $n > N$

$$\frac{L \cdot C^2 M}{n^{\alpha} h_n^{2d} m^2} > \frac{C^2 M}{n^{\alpha/2} h_n^d (m/2)[n^{\alpha/2} h_n^d (m/2) - Cn^{-\alpha/2}]}. \tag{46}$$

From the basic KDE consistency condition $nh_n^d \overset{n\to\infty}{\to} \infty$, requiring $\alpha > \max(2, \log(2C/(h_n^d m))/\log n)$ ensures that the approximation error vanishes with increasing n.

2) While the convergence rates for this case must be at least as rapid as for the a.s. uniform case [by squaring and taking expectations on both sides of (42) before computing the sup on the left-hand side], we can obtain slightly better convergence rates with tighter m.s. estimates of the terms in (42). We begin be noting that it is sufficient to demonstrate the corresponding result in absolute value since

$$\begin{aligned}
&\mathbb{E}_{T_n}[|\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}'_n(\boldsymbol{z})|^2] \\
&\quad = \mathbb{E}_{T_n}[(\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}'_n(\boldsymbol{z}))(\tilde{f}_{n,\infty}(\boldsymbol{z}) + \tilde{f}'_n(\boldsymbol{z})) \\
&\qquad + 2\tilde{f}'_n(\boldsymbol{z})(\tilde{f}'_n(\boldsymbol{z}) - \tilde{f}_{n,\infty}(\boldsymbol{z}))] \\
&\quad \leq \sup_{\boldsymbol{z}\in D}(|\tilde{f}_{n,\infty}(\boldsymbol{z}) + \tilde{f}'_n(\boldsymbol{z})| + 2|\tilde{f}'_n(\boldsymbol{z})|) \\
&\qquad \cdot \mathbb{E}_{T_n}[|\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}'_n(\boldsymbol{z})|]
\end{aligned} \tag{47}$$

where the supremum is $\mathcal{O}(R_1)$ for n sufficiently large by assumption (18). Returning to the expectation term, taking expectations with respect to P_{T_n} on both sides of (42), and applying Cauchy–Schwarz gives

$$\begin{aligned}
&\mathbb{E}_{T_n}[|\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}'_n(\boldsymbol{z})|] \\
&\quad \leq \mathbb{E}^{1/2}_{T_n}[||\boldsymbol{g}_n(\boldsymbol{z})||^2]\mathbb{E}^{1/2}_{T_n}[||\boldsymbol{G}_n||^2] \\
&\qquad \cdot \mathbb{E}^{1/2}_{T_n}[\lambda_n^{-2}(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - ||\boldsymbol{G}_n||)^{-2}]\mathbb{E}^{1/2}_{T_n}[||\boldsymbol{y}'_n||^2].
\end{aligned} \tag{48}$$

The first term squared $\mathbb{E}_{T_n}[||\boldsymbol{g}_n(\boldsymbol{z})||^2]$ can be asymptotically bounded in Euclidean norm as

$$\begin{aligned}
\mathbb{E}_{T_n}[||\boldsymbol{g}_n(\boldsymbol{z})||^2] &= \sum_{i=1}^{n} \mathbb{E}_{T_n}\left[K^2\left(\frac{\boldsymbol{z} - \boldsymbol{Z}(i)}{h_n}\right)\right] \\
&= \mathcal{O}(nh_n^d p(\boldsymbol{z})||K||_2^2) \quad \text{a.e.}
\end{aligned} \tag{49}$$

where $||\cdot||_2$ is the standard L_2 norm with respect to Lebesgue measure, and we have used the fact that (see in [17, eq. (2.10)])

$$\left|\int_{\mathbb{R}^d} K^2\left(\frac{\boldsymbol{x} - \boldsymbol{u}}{h_n}\right) p(\boldsymbol{u})\, d\boldsymbol{u} - h_n^d p(\boldsymbol{x})||K||_2^2\right| \overset{n\to\infty}{\to} 0 \text{ a.e.} \tag{50}$$

where $||\cdot||_2$ is the usual $L_2(\mathbb{R}^d)$ norm with respect to Lebesgue measure. Similarly, we bound the second term squared $\mathbb{E}_{T_n}[||\boldsymbol{G}_n||^2]$ in Fröbenius norm and apply (50) with Lebesgue dominated convergence to obtain

$$\begin{aligned}
\mathbb{E}_{T_n}[||\boldsymbol{G}_n||^2] &\leq \sum_{i,j=1}^{n} \mathbb{E}_{\boldsymbol{Z}(\boldsymbol{i}),\boldsymbol{Z}(\boldsymbol{j})}\left[K^2\left(\frac{\boldsymbol{Z}(i) - \boldsymbol{Z}(j)}{h_n}\right)\right] \\
&= \mathcal{O}\left(n^2 \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} K^2\left(\frac{\boldsymbol{z}(i) - \boldsymbol{z}(j)}{h_n}\right) p(\boldsymbol{z}(i)) \right. \\
&\qquad \left. \cdot\, p(\boldsymbol{z}(j))\, d\boldsymbol{z}(i)\, d\boldsymbol{z}(j)\right) \\
&= \mathcal{O}(n^2 h_n^d ||p||_2^2 ||K||_2^2)
\end{aligned} \tag{51}$$

where we have applied (19). For the square of the middle term, we may again apply the majorization $||\boldsymbol{G}_n|| < n$ and use the same argument as for (46) to obtain the estimate

$$\begin{aligned}
&\mathbb{E}_{T_n}[\lambda_n^{-2}(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - ||\boldsymbol{G}_n||)^{-2}] \\
&\quad \leq L \cdot \mathbb{E}_{T_n}[\lambda_n^{-4}(\boldsymbol{z})], \quad \forall \boldsymbol{z} \in D \quad \text{when} \quad n > N_1
\end{aligned} \tag{52}$$

for some $L > 0$ and $N_1 \in \mathbb{N}$. Next, we may substitute p for $\tilde{p}_n$ in the expectation with error bounded by

$$\begin{aligned}
&|\mathbb{E}_{T_n}[\lambda_n^{-4}(\boldsymbol{z})] - \lambda^{-4}(\boldsymbol{z})| \\
&\quad \leq \mathbb{E}_{T_n}[|\lambda_n^{-4}(\boldsymbol{z}) - \lambda^{-4}(\boldsymbol{z})|] \\
&\quad \leq \mathbb{E}_{T_n}\left[\left|\frac{\lambda_n^4(\boldsymbol{z}) - \lambda^4(\boldsymbol{z})}{\lambda_n^4(\boldsymbol{z}) \cdot \lambda^4(\boldsymbol{z})}\right|\right] \\
&\quad \leq \mathbb{E}_{T_n}\left[\left|\frac{(\lambda_n^2(\boldsymbol{z}) + \lambda^2(\boldsymbol{z}))(\lambda_n(\boldsymbol{z}) + \lambda\boldsymbol{z}))(\lambda_n(\boldsymbol{z}) - \lambda\boldsymbol{z}))}{\lambda_n^4(\boldsymbol{z}) \cdot \lambda^4(\boldsymbol{z})}\right|\right]
\end{aligned} \tag{53}$$

where $\lambda(\boldsymbol{z}) \triangleq n^{\alpha+1} h_n^d p(\boldsymbol{z})/C$, whence, by Cauchy–Schwarz

$$\begin{aligned}
&\sup_{\boldsymbol{z}\in D} |\mathbb{E}_{T_n}[\lambda_n^{-4}(\boldsymbol{z})] - \lambda^{-4}(\boldsymbol{z})| \\
&\qquad \leq \sup_{\boldsymbol{z}\in D} \mathbb{E}^{1/2}_{T_n}\left[\left|\frac{(\tilde{p}_n^2(\boldsymbol{z}) + p^2(\boldsymbol{z}))(\tilde{p}_n(\boldsymbol{z}) + p(\boldsymbol{z}))}{(n^{\alpha+1} h_n^d \tilde{p}_n(\boldsymbol{z})/C)^4 \cdot p^4(\boldsymbol{z})}\right|^2\right] \\
&\qquad \cdot \sqrt{\sup_{\boldsymbol{z}\in D} \mathbb{E}_{T_n}[|\tilde{p}_n(\boldsymbol{z}) - p(\boldsymbol{z})|^2]}.
\end{aligned} \tag{54}$$

By (18) and (20), the first sup term is (at least) bounded for α sufficiently large, whereas the second term vanishes by (17). Therefore, we have that for n sufficiently large

$$\begin{aligned}
&\sup_{\boldsymbol{z}\in D} \mathbb{E}_{T_n}[\lambda_n^{-2}(\boldsymbol{z})(\lambda_n(\boldsymbol{z}) - ||\boldsymbol{G}_n||)^{-2}] \\
&\qquad = \mathcal{O}(\sup_{\boldsymbol{z}\in D} p^{-4}(\boldsymbol{z})) = \mathcal{O}((n^{\alpha+1} h_n^d m/C)^{-4}).
\end{aligned} \tag{55}$$

The square of the last term $\mathbb{E}_{T_n}[||\boldsymbol{y}'_n||^2]$ is bounded (by assumption) by $nn^{2\alpha}M^2$ so that combining the square roots of the previous terms leaves us with the conclusion that for sufficiently large n, we have (56), shown at the bottom of the next page. In order for (48) to be valid,

we require that (42) hold uniformly over D, leading to the previous condition of $\alpha > \log(2C/(h_n^d m))/\log n$, where m is as defined in (45). As an aside, if (17) is weakened to the corresponding mean-input case, then (42) need only hold a.s.-P. Finally, the lower limits of 1 and ν imposed on α by the maximum function ensures that the upper bound on the m.s. approximation error in (21) decreases as $n \to \infty$ by the consistency condition $nh_n^d \overset{n\to\infty}{\to} \infty$. □

The implications of these approximation theorems are discussed in greater length in [23, Sec. 2.1]. Here, we merely note that while the introduction of $F_{\boldsymbol{z}}$ is motivated by its utility in the proofs, the arguments contained therein imply, nonetheless, that over any given compact set $D \subset \mathbb{R}^d$, the approximating regularized RBFN's have λ_n growing asymptotically at rate *at least* $\Omega(n^{\alpha+4/(d+4)} \log^d n)$[5] for $\alpha > 2$ in the uniform case and $\Omega(n^{\alpha+4/(d+4)})$ for $\alpha > 1$ in the m.s. case, i.e., at least roughly $\Omega(n)$ in both cases. Thus, for our purposes of comparison with regularized RBFN's trained in the "usual" way, i.e., with a single regularization parameter determined once from a realized training set t_n and used thereafter over the entire network domain, it suffices to consider the NWRE as (roughly speaking) "infinitely" regularized RBFN's.

Appendix B
Proof of Theorem 2

Before proceeding, we shall need the following elementary lemma concerning the convergence in probability of one-nearest neighbor distances.

Lemma 2: Let $\{\boldsymbol{Z}(1), \boldsymbol{Z}(2), \cdots, \boldsymbol{Z}(n), \boldsymbol{Z}\} \triangleq \{\boldsymbol{Z}_n, \boldsymbol{Z}\} \sim P_{\boldsymbol{Z},\boldsymbol{Z}_n}$ be $n+1$ consecutive samples from a geometrically ergodic process $\{\boldsymbol{Z}(i)\}$ with stationary (marginal) measure $P_{\boldsymbol{Z}}$. Then, for each $\epsilon > 0$

$$P_{\boldsymbol{Z},\boldsymbol{Z}_n}\{\boldsymbol{z}, \boldsymbol{z}_n\colon \min_{j=1,2,\cdots,n} ||\boldsymbol{z} - \boldsymbol{z}(j)|| > \epsilon\} \overset{n\to\infty}{\to} 0 \tag{57}$$

where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^d$.

Proof: Let $\epsilon > 0$ be given. Set $A_{n,j}(\epsilon) \triangleq \{\boldsymbol{z}, \boldsymbol{z}_n\colon ||\boldsymbol{z} - \boldsymbol{z}(j)|| > \epsilon\}$. We use the independence bound implied by Cauchy–Schwarz for the intersection of a finite collection of events $\{F_i\}_{i=1}^n$ defined with respect to a common probability measure P

$$P\left(\bigcap_{i=1}^n F_i\right) = \mathbb{E}\left[\prod_{i=1}^n I(F_i)\right] \le \prod_{i=1}^n P^{1/2}(F_i) \tag{58}$$

where $I(\cdot)$ is the indicator function for event $\cdot$ so that

$$P_{\boldsymbol{Z},\boldsymbol{Z}_n}\left(\bigcap_{j=1}^n A_{n,j}(\epsilon)\right) \le \prod_{j=1}^n P^{1/2}_{\boldsymbol{Z},\boldsymbol{Z}(j)}(A_{n,j}) \triangleq \prod_{j=1}^n q_j^{1/2}(\epsilon). \tag{59}$$

Let p and p_j be the marginal densities of $\boldsymbol{Z}$ and $\boldsymbol{Z}(j)$ with supports S and S_j, respectively. Then

$$\begin{aligned} q_j(\epsilon) &= \int_{\boldsymbol{x}\in S_j} P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| > \epsilon\} p_j(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in S_j \Delta S} P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| > \epsilon\} p_j(\boldsymbol{x})\, d\boldsymbol{x} \\ &\quad + \int_{\boldsymbol{x}\in S_j \cap S} P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| > \epsilon\} p_j(\boldsymbol{x})\, d\boldsymbol{x}. \end{aligned} \tag{60}$$

By the geometric ergodicity of $\{\boldsymbol{Z}(i)\}$, the first integral can be made arbitrarily small for j sufficiently large (since $P_{\boldsymbol{Z}(j)}(\boldsymbol{z}(j) \in S_j \Delta S) \overset{j\to\infty}{\to} 0$ by the triangle inequality $||P_{\boldsymbol{Z}} - P_{\boldsymbol{Z}(j)}||_V \le ||P_{\boldsymbol{Z}} - \pi||_V + ||\pi - P_{\boldsymbol{Z}(j)}||_V$, where π is the stationary measure for $\{\boldsymbol{Z}(i)\}$), whereas the second integral can be expressed as

$$\begin{aligned} &\int_{\boldsymbol{x}\in S_j\cap S} P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| > \epsilon\} p_j(\boldsymbol{x})\, d\boldsymbol{x} \\ &\quad = \int_{\boldsymbol{x}\in S_j\cap S} (1 - P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| < \epsilon\}) p_j(\boldsymbol{x})\, d\boldsymbol{x} \\ &\quad = P_{\boldsymbol{Z}(j)}(\boldsymbol{z}(j) \in S_j \cap S) \\ &\qquad - \int_{\boldsymbol{x}\in S_j\cap S} P_{\boldsymbol{Z}}\{\boldsymbol{z} \in S\colon ||\boldsymbol{z} - \boldsymbol{x}|| < \epsilon\} p_j(\boldsymbol{x})\, d\boldsymbol{x} \end{aligned} \tag{61}$$

in which the first term is no greater than unity, whereas the strict positivity of the second term given $\epsilon > 0$ follows from that of the integrand. To see this last fact, note that given any $\epsilon > 0$, when $\boldsymbol{x} \in S$, the integrand must be strictly positive because a) $P_{\boldsymbol{Z}}$ is assumed absolutely continuous with respect to Lebesgue measure, hence, S must have nonzero Lebesgue measure, and b) p is almost everywhere (Lebesgue) continuous; therefore, an arbitrary radius open ball centered at almost all points of S must have nonzero $P_{\boldsymbol{Z}}$ measure. The proof can now be completed: Given $\delta > 0$, take N sufficiently large so that $q_j^{1/2}(\epsilon) < 1/2$ for all $j > N$. Then, for all $n > N - \log_2 \delta$, we have $\Pi_{j=1}^n q_j^{1/2}(\epsilon) < \delta$, as required. □

We generalize slightly the definitions of loss and risk from Section III. Define the (squared-error) loss $\tilde{L}_2$ of an estimate $\tilde{f}_n$ with respect to a true function f given t_n as

$$\tilde{L}_2(\tilde{f}_n, f, t_n) \triangleq \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{z}(i)) - \tilde{f}_n(\boldsymbol{z}(i)))^2 \tag{62}$$

[5] $f(n) = \Omega(g(n))$ means $\exists C > 0$ such that $|f(n)| \ge C|g(n)|$ for all n sufficiently large.

$$\begin{aligned} \sup_{\boldsymbol{z}\in D} \mathbb{E}_{T_n}[|\tilde{f}_{n,\infty}(\boldsymbol{z}) - \tilde{f}'_n(\boldsymbol{z})|^2] &= \mathcal{O}\left(\frac{(nh_n^d L||K||_2^2)^{1/2} \cdot (n^2 h_n^d ||p||_2^2 ||K||_2^2)^{1/2} \cdot \sqrt{n} n^\alpha M}{(n^{\alpha+1} h_n^d m/C)^2}\right) \\ &= \mathcal{O}(C^2 M \sqrt{L} ||p||_2 ||K||_2^2 n^{-\alpha} h_n^{-d} m^{-2}) \end{aligned} \tag{56}$$

and the risk $\tilde{R}_2$, i.e.,

$$\tilde{R}_2(\tilde{f}_n, f) \triangleq \mathbb{E}_{T_n}[\tilde{L}_2(\tilde{f}_n, f, T_n)]. \tag{63}$$

Similarly, define the global m.s.e. or risk as

$$R_2(f, \tilde{f}_n) \triangleq \mathbb{E}[(f(\boldsymbol{Z}) - \tilde{f}_n(\boldsymbol{Z}))^2]. \tag{64}$$

By identifying $\tilde{R}_2(f, \tilde{f}_n)$ with $R_n(\tilde{\lambda}_n)$ and $R_2(f, \tilde{f}_n)$ with $\mathbb{E}[\delta_n^2(n+1)]$ [by letting $\boldsymbol{Z} \triangleq \boldsymbol{Z}(n+1)$], we can prove Theorem 2 as a slightly modified version a corresponding theorem from [23]. Conditions a) and b) in the preamble of Theorem 2 correspond to conditions 1 and 2 stated below.

Theorem 3 [23]: Assume that f is bounded as $|f| < L_f$ and Lipschitz with constant K_f over $\mathbb{R}^d$. If $\{\boldsymbol{Z}(i)\} \triangleq \{\boldsymbol{X}_p(i)\}$ is a geometrically ergodic process, with stationary measure π absolutely continuous with respect to Lebesgue measure via density p_π, and satisfying the following.

1) There exists a positive constant L_π satisfying

$$\sup_{\boldsymbol{z} \in \mathbb{R}^d} |p_\pi(\boldsymbol{z})| < L_\pi. \tag{65}$$

2) There exist positive constants L and K for the regularized RBFN estimate $\tilde{f}_n$ constructed from T_n satisfying for $n = 1, 2, \cdots$

$$\sup_{\boldsymbol{z} \in \mathbb{R}^d} |\tilde{f}_n(\boldsymbol{z}, T_n)| \le L \quad \text{a.s.-}P_{T_n} \tag{66}$$

$$K_n \le K \quad \text{a.s.-}P_{T_n} \tag{67}$$

where K_n is a Lipschitz constant for $\tilde{f}_n$.

Then

$$|R_2(f, \tilde{f}_n) - \tilde{R}_2(f, \tilde{f}_n)| \overset{n\to\infty}{\to} 0 \quad \text{a.s.-}P_{T_n}. \tag{68}$$

Proof: First, we note that when the process $\{\boldsymbol{Z}(i)\}$ is geometrically ergodic as assumed, (65) implies that the corresponding sequence of marginal densities $\{p_j\}$ is also bounded, i.e., there exists $L_p > 0$ such that

$$\sup_{j \in \mathbb{N}} \sup_{\boldsymbol{z} \in \mathbb{R}^d} |p_j(\boldsymbol{z})| < L_p \tag{69}$$

where p_j is the marginal density for $\boldsymbol{Z}(j)$. The geometric ergodicity condition implies the (Lebesgue) a.e. pointwise-convergence of p_j to p_π as $j \to \infty$ (by choosing B to be a point set when applying the definition of total variation norm in footnote 3). Thus, p_j can be bounded either by L_π, when $j > N$ for some $N \in \mathbb{N}$ sufficiently large, or by $\sup_{j=1,2,\cdots,N} \sup_{\boldsymbol{z} \in \mathbb{R}^d} |p_j(\boldsymbol{z})|$ for $1 \le j \le N$.

For convenience of notation, let $v_{t_n}(\cdot) \triangleq (f(\cdot) - \tilde{f}_n(\cdot, t_n))^2$. Consider the ϵ-cover $B_n(\epsilon)$ induced by a realized training sequence t_n, i.e., $B_n(\epsilon) \triangleq \cup_{i=1}^n B_{n,i}(\epsilon)$, $B_{n,i}(\epsilon) \triangleq \{\boldsymbol{z}, t_n\colon \|\boldsymbol{z} - \boldsymbol{z}(i)\| < \epsilon\}$. An equivalent disjoint cover may be obtained by replacing $B_{n,i}(\epsilon)$ in the definition of $B_n(\epsilon)$ with $D_{n,i}(\epsilon) \triangleq B_{n,i}(\epsilon) \cap V_{\boldsymbol{z}_n}(\boldsymbol{z}(i))$, where $V_{\boldsymbol{z}_n}(\boldsymbol{z}(i))$ is the Voronoi cell centred at $\boldsymbol{z}(i)$ of the partition induced by the input sequence $\boldsymbol{z}_n$ contained in t_n. Decompose R_2 with respect to $B_n(\epsilon)$, where $\epsilon = \epsilon(n)$ (as will be explained later) so that

$$\begin{aligned} R_2(f, \tilde{f}_n) &\triangleq \int_{\boldsymbol{z}, t_n} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n) \\ &= \int_{(\boldsymbol{z}, t_n) \in B_n(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n) \\ &\quad + \int_{(\boldsymbol{z}, t_n) \in B_n^c(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n). \end{aligned} \tag{70}$$

By the assumed boundedness of f and condition (66) on $\tilde{f}_n$, Lemma 2 implies that the latter integral can be made arbitrarily small for n sufficiently large since for any $\delta > 0$

$$\int_{(\boldsymbol{z}, t_n) \in B_n^c(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n) \le L_v P_{\boldsymbol{Z}, \boldsymbol{Z}_n}(B_n^c(\epsilon)) \le \delta \tag{71}$$

when n satisfies

$$\log\left(\frac{\delta}{L_v}\right) \Big/ \sum_{j=1}^n \log q_j(\epsilon) < 1/2 \tag{72}$$

where $L_v = (L_f + L)^2$ is a global upper bound on v, and q_j is as defined in Lemma 2. For the former integral, we may write

$$\begin{aligned} &\int_{(\boldsymbol{z}, t_n) \in B_n(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n) \\ &\quad = \sum_{i=1}^n \int_{(\boldsymbol{z}, t_n) \in D_{n,i}(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z}, t_n) \\ &\quad \overset{+}{\lessgtr} \sum_{i=1}^n \int_{(\boldsymbol{z}, t_n) \in D_{n,i}(\epsilon)} (v_{t_n}(\boldsymbol{z}(i)) \mp K_v \epsilon)\, dP(\boldsymbol{z}, t_n) \\ &\quad \overset{+}{\lessgtr} \int_{t_n} \left[\sum_{i=1}^n (v_{t_n}(\boldsymbol{z}(i)) \mp K_v \epsilon) P_{\boldsymbol{Z}|T_n}(D_{n,i}(\epsilon) | T_n = t_n) \right] \\ &\qquad \cdot dP(t_n) \end{aligned} \tag{73}$$

and by the definition of $D_{n,i}(\epsilon)$ and the fact that f and $\tilde{f}_n$ are bounded and Lipschitz implies the same for v with Lipschitz constant not greater than $K_v \triangleq 2(L_f + L)(K_f + K)$. Here, the notation $a \overset{+}{\lessgtr} f(b \mp c)$ is shorthand for the double inequality $f(b - c) \le a \le f(b + c)$, where f is an expression containing $b \mp c$. The remainder term containing $K_v \epsilon$ can be bounded uniformly over all possible training realizations t_n (equivalently, over all possible training input realizations $\boldsymbol{z}_n$) since

$$\begin{aligned} &P_{\boldsymbol{Z}|T_n}(D_{n,i}(\epsilon) | T_n = t_n) \\ &\qquad \le L_p (2\epsilon)^d, \quad \forall t_n \quad \text{and} \quad i = 1, 2, \cdots, n \end{aligned} \tag{74}$$

by (69) and where we have used the (Euclidean) volume of a d-dimensional cube in $\mathbb{R}^d$ with edge 2ϵ to upper-bound the

volume of corresponding closed ball of radius ϵ. Hence, we have the deviation bound

$$\left| \int_{\boldsymbol{z},t_n \in D_n(\epsilon)} v_{t_n}(\boldsymbol{z})\, dP(\boldsymbol{z},t_n) - \int_{t_n} \sum_{i=1}^{n} v_{t_n}(\boldsymbol{z}(i)) \cdot P_{\boldsymbol{Z}|T_n}(D_{n,i}(\epsilon)|T_n = t_n)\, dP(t_n) \right| \leq nK_v L_p \epsilon (2\epsilon)^d. \tag{75}$$

For the remainder term $r(n) \stackrel{\Delta}{=} nK_v L_p \epsilon (2\epsilon)^d$ to vanish as $n \to \infty$, we require that $\epsilon^{d+1} = \mathcal{O}(1/n^{1+\beta})$ for some $\beta > 0$. At the same time, the inequality

$$q_i(\epsilon) = 1 - P_{\boldsymbol{Z},\boldsymbol{Z}(i)}(B_{n,i}(\epsilon)) \geq 1 - L_p(2\epsilon)^d \quad i = 1, 2, \cdots, n \tag{76}$$

implies that we cannot let ϵ decrease too quickly as $n \to \infty$ if (72) is to be satisfiable for $\delta = \delta(n) = \mathcal{O}(1/n^{\alpha})$ with $\alpha > 0$ since for x small, $\log(1-x) \approx -x$. In other words, for (72) to hold with $L_v > \delta(n) \stackrel{n\to\infty}{\to} 0$, it is necessary that $\epsilon^d = \Omega(1/n^{1-\gamma})$ for some $\gamma \in (0,1)$. Equating the two exponents gives the relationship between β and γ as

$$0 < \beta < 1/d, \quad \gamma = \frac{1-\beta d}{1+d}. \tag{77}$$

Returning to the integral term in (75), we recombine the iterated expectation and note that

$$\begin{aligned} &\left| \int_{t_n} \sum_{i=1}^{n} v_{t_n}(\boldsymbol{z}(i))\, dP(t_n) - \int_{t_n} \sum_{i=1}^{n} v_{t_n}(\boldsymbol{z}(i)) \cdot P_{\boldsymbol{Z}|T_n}(D_{n,i}(\epsilon)|T_n = t_n)\, dP(t_n) \right| \\ &\leq \sup_{i=1,2,\cdots,n} |v_{t_n}(\boldsymbol{z}(i))| \cdot \left| \sum_{i=1}^{n} \left(\frac{1}{n} - \int_{\boldsymbol{z},t_n \in D_{n,i}(\epsilon)} dP(\boldsymbol{z},t_n) \right) \right| \\ &\leq L_v \left| 1 - \sum_{i=1}^{n} P_{\boldsymbol{Z},T_n}(D_{n,i}(\epsilon)) \right| \qquad (78) \\ &\leq L_v |1 - P_{\boldsymbol{Z},T_n}(B_n(\epsilon))| = L_v \prod_{j=1}^{n} q_j^{1/2}(\epsilon) \\ &\leq \delta \qquad (79) \end{aligned}$$

where we have again invoked Lemma 2 in the last line for δ, as defined in (72). Combining the inequalities (71), (75), and (79) yields

$$\begin{aligned} &|R_2(f,\tilde{f}_n) - \tilde{R}_2(f,\tilde{f}_n)| \\ &\quad \leq r(n) + 2\delta(n) \\ &\quad = \mathcal{O}(n^{-\beta}) + \mathcal{O}(n^{-\alpha}), \quad 0 < \beta < 1/d \\ &\qquad \alpha + \beta < 1 \end{aligned} \tag{80}$$

where the condition $\alpha + \beta < 1$ is required for (72) to hold. This result implies that the asymptotic rate of convergence of $\tilde{R}_2(f,\tilde{f}_n)$ to $R_2(f,\tilde{f}_n)$ can be made arbitrarily close to (but strictly less than) $\mathcal{O}(n^{-1/d})$, from which the desired conclusion follows. □

We note that (67) is satisfied when, e.g., the kernel function K is Lipschitz since we have assumed in both condition A.2) and the theorem preamble that the underlying map f is Lipschitz and chosen conditions so that the NWRE and, hence, the RBFN converge to f in a compatible mode. Furthermore, for the smooth functions and absolutely continuous measures assumed throughout, m.s. convergence over a compact set D implies pointwise a.e. convergence; hence, the estimates $\tilde{f}_n$ must converge to a Lipschitz function.

ACKNOWLEDGMENT

The authors would like to acknowledge the assistance of D. McArthur at the Communications Research Laboratory for providing the speech data listed in Tables V and VI. We also thank the anonymous reviewers who helped shape this paper with their constructive comments and suggestions.

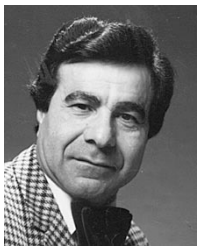
REFERENCES

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
[2] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Comput.*, vol. 3, pp. 246–257, 1991.
[3] S. Haykin, S. Puthusserypady, and P. Yee, "Dynamic reconstruction of a chaotic process using regularized RBF networks," Commun. Res. Lab. Rep. 353, McMaster Univ., Hamilton, Ont., Canada, Sept. 1997.
[4] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D*, vol. 35, pp. 335–356, 1989.
[5] D. Lowe and A. R. Webb, "Time series prediction by adaptive networks: A dynamical systems perspective," *Proc. Inst. Elect. Eng. F*, vol. 138, pp. 17–34, Feb. 1990.
[6] T. Terano, K. Asai, and M. Sugeno, *Fuzzy Systems Theory and Its Applications*. Boston, MA: Academic, 1992.
[7] V. Kadirkamanathan and M. Kadirkamanathan, "Recursive estimation of dynamic modular RBF networks," in *Advances in Neural Information Processing Systems* San Mateo, CA: Morgan Kaufman, 1996, vol. 8, pp. 239–245.
[8] T. Leen and G. Orr, "Weight-space probability densities and convergence for stochastic learning," in *Proc IJCNN*, Baltimore, MD, 1992, vol. 4, pp. 158–164.
[9] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, pp. 954–975, 1993.
[10] D. Lowe and A. McLachlan, "Modeling of nonstationary process using radial basis function networks," in *Proc. IEE ANN*, Cambridge, U.K., 1995, no. 409, pp. 300-305.
[11] M. von Golitschek and L. L. Schumaker, "Data fitting by penalized least squares," in *Algorithms for Approximation II*, J. C. Mason and M. G. Cox, Eds. London, U.K.: Chapman & Hall, pp. 210–227.
[12] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA; SIAM, 1990, vol. 59.
[13] E. A. Nadaraya, "On estimating regression," *Theor. Probab. Appl.*, vol. 9, pp. 141–142, 1964.
[14] E. A. Nadaraya, "On nonparametric estimation of density functions and regression curves," *Theor. Probab. Appl.*, vol. 10, pp. 186–190, 1965.
[15] G. S. Watson, "Smooth regression analysis," *Sankhyā A*, vol. 26, pp. 359–372, 1964.
[16] L. Györfi, W. Härdle, P. Sarda, and P. Vieu, *Nonparametric Curve Estimation from Time Series*. Heidelberg, Germany: Springer-Verlag, 1989, vol. 60, Lecture Notes in Statistics.
[17] D. Bosq, *Nonparametric Statistics for Stochastic Processes*. New York: Springer-Verlag, 1996, vol. 110, Lecture Notes in Statistics.
[18] M. Jones, F. Girosi, and T. Poggio, "Priors, stabilizers and basis functions: From regularization to radial, tensor, and additive splines,"

Tech. Rep. A. I. Memo 1430, Mass. Inst. Technol., Cambridge, Mar. 1994.
[19] L. Xu, A. Krzyżak, and A. Yuille, "On radial basis function nets and kernel regression: Statistical consistency, convergence rates, and receptive field size," *Neural Networks*, vol. 7, pp. 609–628, 1994.
[20] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.
[21] P. Douhkan, *Mixing: Properties and Examples*. New York: Springer-Verlag, 1994, vol. 85, Lecture Notes in Statistics.
[22] R. C. Bradley, "Basic properties of strong mixing conditions," in *Dependence in Probability and Statistics*, E. Eberlein and M. S. Taqqu, Eds. Boston, MA: Birkhauser, vol. 11, Progressive in Probability and Statistics, pp. 165–192.
[23] P. V. Yee, "Regularized radial basis function networks: theory and applications to probability estimation, classification, and time series prediction," Ph.D. dissertation, Dept. Elect. Comput. Eng., McMaster Univ., Hamilton, Ont., Canada, 1998.
[24] H. Tong, *Non-Linear Time Series: A Dynamical Systems Approach*. Oxford, U.K.: Oxford Sci., 1990.
[25] R. L. Eubank, *Spline Smoothing and Nonparametric Regression, Statistics: Textbooks and Monographs*. New York: Marcel Dekker, 1988, vol. 90.
[26] D. W. K. Andrews, "Asymptotic optimality of generalized C_L, cross-validation, and generalized cross-validation in regression with heteroskedastic errors," *J. Econometr.*, vol. 47, pp. 359–377, 1991.
[27] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996, 3rd ed.
[28] J. C. Platt, "A resource allocating network for function interpolation," *Neural Comput.*, vol. 3, pp. 213–225, 1991.
[29] S. Haykin, A. H. Sayed, J. Zeidler, P. Yee, and P. Wei, "Adaptive tracking of linear time-variant systems by extended RLS algorithms," *IEEE Trans. Signal Processing*, vol. 45, pp. 1118–1128, May 1997.
[30] M. R. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," in *Proc. ICASSP*, Tampa, FL, Apr. 1985, p. 937.
[31] S. Haykin and L. Li, "Non-linear adaptive prediction of nonstationary signals," *IEEE Trans. Signal Processing*, vol. 43, pp. 526–535, Feb. 1995.
[32] J. Baltersee and J. A. Chambers, "Nonlinear adaptive prediction of speech with a pipelined recurrent neural network," *IEEE Trans. Signal Processing*, vol. 46, pp. 2207–2216, Aug. 1998.
[33] J. C. Schouten and C. M. van den Bleek, *RRCHAOS Time Series Analysis Software*, Chemical Reactor Engineering Section, Delft Univ. Technology, Delft, The Netherlands, 1994.
[34] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: The method of surrogate data," *Physica D*, vol. 58, pp. 77–94, 1992.
[35] B. Auestad and D. Tjøstheim, "Identification of nonlinear time series: First order characterization and order determination," *Biometrika*, vol. 77, pp. 669–687, 1990.
[36] P. Yee and S. Haykin, "A dynamic, regularized Gaussian radial basis function network for nonlinear, nonstationary time series prediction," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 3419–3422.
[37] L. Devroye and L. Györfi, *Nonparametric Density Estimation, the L_1 View*. New York: Wiley, 1985.
[38] B. Townshend, "Nonlinear prediction of Speech," in *Proc. ICASSP*, Toronto, Ont., Canada, May 1991, vol. 1, pp. 425–428.
[39] F. D. de Maria and A. R. Figueiras-Vidal, "Nonlinear prediction for speech coding using radial basis functions," in *Proc. ICASSP*, Detroit, MI, May 1995, vol. 1, pp. 788–791.
[40] W. W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, pp. 221–239, June 1989.

Paul Yee (S'86–M'98) received the B.A.Sc. (Hons.) degree engineering from the University of British Columbia, Vancouver, Canada, in 1989 and the Ph.D. degree from McMaster University, Hamilton, Ont., Canada, in 1998, all in electrical engineering.

From 1989 to 1991, he worked as a Design Engineer at Newbridge Networks Corporation, Kanata, Ont.. From 1991 to 1998, he was affliated with the Communications Research Laboratory, McMaster University, as a Ph.D. candidate. In April 1998, he joined DATUM Telegraphic, Inc., Vancouver, where he is presently a Member of Technical Staff. His research interests center around the theory and application of nonparametric modeling to probability estimation, classification, and time-series prediction.



Simon Haykin (F'82) received the B.Sc. degree (with first-class honors) in 1953, the Ph.D. degree in 1956, and the D.Sc. degree in 1967, all in electrical engineering, from the University of Birmingham, Birmingham, U.K.

He is the Founding Director of the Communications Research Laboratory, McMaster University, Hamilton, Ont., Canada. His research interests include nonlinear dynamics, neural networks, adaptive filters, and their applications in radar and communication systems. In 1996, he was awarded the title "University Professor." He is the Editor of *Adaptive and Learning Systems for Signal Processing, Communications, and Control*, which is a new series of books for Wiley-Interscience.

Dr. Haykin was elected Fellow of the Royal Society of Canada in 1980. He was awarded the McNaughton Gold Medal, IEEE (Region 7), in 1986. He is a recipient of the Canadian Telecommunications Award from Queen's University.