Power Prediction in Mobile Communication Systems Using an Optimal Neural-Network Structure

Xiao Ming Gao, Xiao Zhi Gao, Jarno M. A. Tanskanen, and Seppo J. Ovaska, *Senior Member, IEEE*

***Abstract*— This paper presents a novel neural-network-based predictor for received power level prediction in direct sequence code division multiple access (DS/CDMA) systems. The predictor consists of an adaptive linear element (Adaline) followed by a multilayer perceptron (MLP). An important but difficult problem in designing such a cascade predictor is to determine the complexity of the networks. We solve this problem by using the predictive minimum description length (PMDL) principle to select the optimal numbers of input and hidden nodes. This approach results in a predictor with both good noise attenuation and excellent generalization capability. The optimized neural networks are used for predictive filtering of very noisy Rayleigh fading signals with 1.8-GHz carrier frequency. Our results show that the optimal neural predictor can provide smoothed in-phase and quadrature signals with signal-to-noise ratio (SNR) gains of about 12 and 7 dB at the urban mobile speeds of 5 and 50 km/h, respectively. The corresponding power signal SNR gains are about 11 and 5 dB. Therefore, the neural predictor is well suitable for power control applications where "delayless" noise attenuation and efficient reduction of fast fading are required.**

***Index Terms*— Mobile communication systems, neural networks, neural networks structure optimization, power prediction, predictive minimum description length (PMDL) principle, Rayleigh fading signal.**

I. Introduction

As THE user capacity of a direct sequence code division multiple access (DS/CDMA) system is inherently interference limited, it is of paramount importance to keep the transmission power of each individual mobile user as low as possible while also receiving the signals of all users at an equal and constant power level at the base station [39]. This is crucial in the transmission from mobiles to a single base station, where all the mobile units need to be controlled by the base station to overcome the near–far effect. The feedback power control procedures allow the base station to send power control commands to either lower or raise independently each user's transmitting power level to maintain the received powers approximately constant and equal. A power control loop with predictive power level estimation is illustrated in Fig. 1.

Manuscript received August 8, 1996; revised July 1, 1997. This work was funded by Nokia Corporation and Telecom Finland.

X. M. Gao is with the Laboratory of Telecommunications Technology, Helsinki University of Technology, Otakaari 5A, FIN-02150 Espoo, Finland.

Z. Gao and S. J. Ovaska are with the Institute of Intelligent Power Electronics, Helsinki University of Technology, Otakaari 5A, FIN-02150 Espoo, Finland.

J. M. A. Tanskanen is with the Laboratory of Signal Processing and Computer Technology, Helsinki University of Technology, Otakaari 5A, FIN-02150 Espoo, Finland.

Publisher Item Identifier S 1045-9227(97)07967-8.

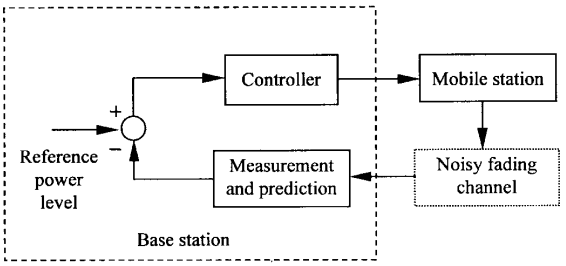

Fig. 1. Power control loop in a CDMA system.

In a recent paper [38], a new power prediction scheme was introduced for compensating the harmful delays in the closed power control loop. In addition to delay compensation, the linear power predictor also bandlimits the noisy power signal and, therefore, both reduces its noise content and smoothens out fast fading. However, the noise attenuation capabilities of the computationally efficient linear schemes are very limited under low signal-to-noise ratio (SNR). A neural-network-based nonlinear predictor was proposed in [25] for improved noise suppression capabilities. On the other hand, the structure of the proposed neural predictor is not optimal and its learning algorithm is not adaptive.

In this paper, we present an optimal neural-network-based predictor for efficient noise reduction. The *hybrid* neural predictor consists of an adaptive linear element (Adaline) and a multilayer perceptron (MLP) [34]. However, when applied directly to the signal heavily corrupted by additive noise, a neural network with excessive numbers of weights and nodes may easily learn the noise component rather than solely the primary signal. This is typically the case in practice when the numbers of these parameters are often chosen by *trial and error*, based on vague subjective optimization. To tackle this problem, we apply an information criterion-based model selection principle, the predictive minimum description length (PMDL) method [28], to select optimal neural-network structures [20], [11], [12]. The *hybrid* neural predictor structure is first optimized *off-line*. The optimized predictor is then used with *on-line adaptation* for predictive filtering of the noisy power signal, the statistical characteristics of which may change drastically with time.

The simulation results demonstrate that the neural predictor offers higher noise attenuation than the earlier linear predictor. Besides, this adaptive neural predictor has considerably wider prediction bandwidth. The linear polynomial predictors

with wide prediction bandwidth evidently have poor noise attenuation [38].

This paper is organized as follows. In Section II, we first review the applied mobile radio channel model, the noise type, and the linear prediction schemes. The neural-network-based *hybrid* predictor is then introduced. In Section III, we give an overview of different criterions for neural-networks selection. The stochastic complexity and its approximation, PMDL principle, are introduced in Section IV. In Section V, the application of PMDL is formalized and used for our neural predictor optimization. The optimized neural predictor is then applied to predict noisy power signals in a Rayleigh fading channel, and the simulation results are given in Section VI. Finally, we conclude this paper with a few remarks in Section VII.

II. Noisy Fading Power Signal and Power Prediction Schemes

A. Channel Model and Noise

A detailed description of the modeling of a Rayleigh fading radio channel and noise was given by Jakes in [15]. In this paper, our signal simulator assumes the superposition of plane waves whose arrival angles are uniformly distributed. Different plane waves are associated with different Doppler shifts ranging from the minimum to the maximum specified by the mobile speed. The simulator consists of low-frequency oscillators at these Doppler shift frequencies, and the frequency distribution results in a satisfactory approximation of the Rayleigh fading. The in-phase and quadrature components, x_c and x_s, respectively, are formed by summing the appropriately weighted oscillator outputs. After multiplication with the corresponding carrier components, the signal is centered at the carrier frequency. Our carrier frequency was 1.8 GHz, the sampling rate of the baseband equivalent in-phase and quardrature components was 1 kHz, and the applied vehicle speeds were 5 and 50 km/h (a "high speed" channel in an urban environment), respectively. The Rayleigh fading simulator is illustrated in Fig. 2.

The noise used was zero mean white Gaussian noise that was independently added to the in-phase and quadrature components. In this paper, we study the performance of the neural predictor for the prediction of noisy Rayleigh fading signals in a "bad" channel, where the component input SNR is 0 dB.

B. Linear Prediction Schemes for Noisy Fading Signal Prediction

The linear predictors often employed for predictive filtering of power signals are the *Heinonen–Neuvo* (H-N) FIR predictor [13] and the *recursive linear smoothed Newton* (RLSN) predictor [23]. Due to their recursive nature, the RLSN predictors can offer much better noise attenuation than the H-N predictors with equal computational burden. Both of these predictors are based on a low-degree polynomial signal model.

There exist two power prediction schemes: 1) direct prediction of the noisy power signal which has been calculated from the noisy in-phase and quadrature components and 2) computing the predictive estimates of the components separately and obtaining the power estimate by summing the squared values of these components. In many cases, however, the noise attenuation capability of the fixed linear methods is not satisfactory [38].

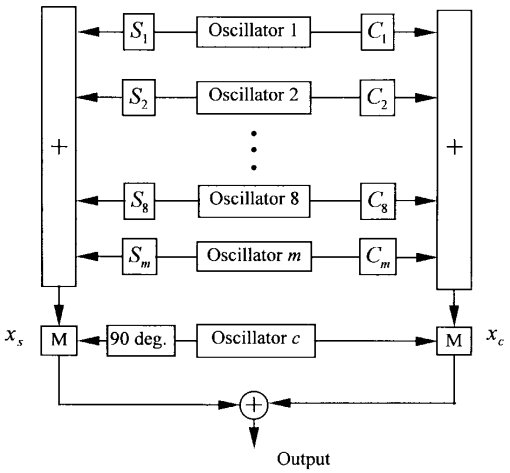

Fig. 2. Rayleigh fading channel simulator. Oscillator m is the maximum Doppler shift frequency oscillator. Oscillator 1, $\cdots$, Oscillator 8 are the Doppler shift frequency oscillators with appropriate frequency distribution, and Oscillator c is the carrier oscillator. Appropriate oscillator phase shifts are obtained by the choice of coefficients $\{S_1, \cdots, S_8\}$, and $\{C_1, \cdots, C_8\}$. M is the carrier modular.

C. Neural-Network-Based Nonlinear Power Predictor

Our neural-network-based predictor is shown in Fig. 3. Because of its remarkable nonlinearity, the MLP will harmfully learn the *noise component* when applied directly to the input signal under low SNR. Therefore, our predictor consists of two modules. An Adaline prefilter is used in Module-1. The output of the Adaline is then fed to the input of Module-2, where an MLP with one hidden layer is used. The hyperbolic tangent sigmoid functions are used as the nonlinear transfer functions of the hidden nodes, and the transfer function of the output node is linear. The single node in the output layer represents one-step-ahead prediction. A tapped delay line type input stage is employed to make it possible to filter out additive noise.

There are many ways to maximize the predictor's generalization and noise attenuation capabilities. From the network structure's point of view, we may select the optimal number of input and hidden nodes, or assume partial connections between different nodes and apply some pruning methods to eliminate very small weights in order to simplify the network structure [8], [9]. Another approach could be the use of a special training method such as early stopping [40], target smoothing [24], or training with jitter [14]. In this paper, we select explicitly the optimal number of input and hidden nodes of the neural predictor.

III. Neural-Network Selection

Although the multilayer perceptron neural network is a widely used network paradigm for solving nonlinear mapping

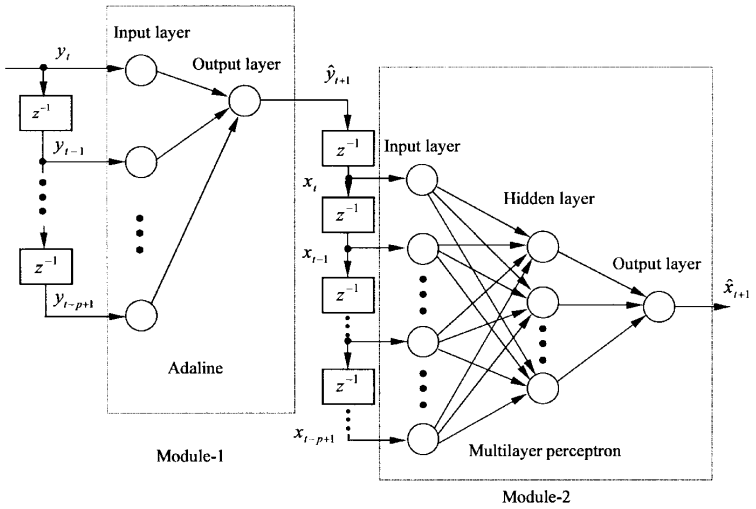

Fig. 3. The structure of the *hybrid* neural-network-based predictor.

problems, there is no general criterion for selecting the optimal structure for a specific problem. It is already known that a neural network with one hidden layer of sigmoidal units can approximate any continuous mapping arbitrarily well, provided that there are enough neurons in the hidden layer [7]. However, the performance for generating accurate outputs for the training inputs competes against predicting appropriate outputs for unknown inputs. For instance, in the case of a multilayered network, when we add some nodes in the hidden layers, the network can produce more precise outputs for the training data, but it may also give worse outputs for unseen data. Further, networks with excessive number of parameters or weights have a higher probability of reaching local minima during the training procedure, and this makes the reproduction even harder. Hence, it is important to find the the simplest possible network structure, i.e., use the minimum number of weights and nodes, without any degradation of performance.

Many model selection schemes have been proposed for determining the network structure for a particular application [10], [19], [21], [26]. For example, a pruning-based approach, called optimal brain damage, was introduced in [6]. The aim of this method is to delete some weights that have the smallest values. However, because there are many complicated connections between the nodes, and those weights with smaller values may be very sensitive to the final solution, this method is not always feasible and some additional judgments should be made. Another method often used to optimize the network structure is by means of local connections and weight sharing [22]. In this scheme, the individual nodes in the hidden layer possess only a local region of inputs so that the number of weights can be reduced. Besides, there exist some other approaches based on the regularization technique, e.g., weight decay and weight-elimination [40]. The idea of these methods is to begin with a network that has an excessive number of parameters for the given problem. Each parameter in the network is associated with a cost function of the form

$$C = \sum_{k \in S} (\text{target}_k - \text{prediction}_k)^2 + \lambda \sum_{i,j} \frac{\left(\frac{w_{ij}}{w_0}\right)^2}{1 + \left(\frac{w_{ij}}{w_0}\right)^2}. \tag{1}$$

The first term in (1) is the sum of the squared errors over the set of observations S. The second term is a term penalizing model complexity, where w_{ij} is a weight, λ and w_0 are freely selected parameters. For a large absolute weight value, $|w_{ij}|$, the cost is approximately equal to λ. If a given performance on the training set can be obtained with fewer weights, this cost function will encourage the reduction and eventually eliminate as many weights as possible. The advantage of this method is that different structures and number of parameters need not be explicitly explored. However, the main difficulty with general regulation techniques is that when forcing smoothness by just adding some penalty terms, we may lose valuable information, e.g., in abruptly changing parts of the signal.

The above mentioned criteria are specific for neural networks. Besides, there are some general methods that can be used for model selection with various different model types. For example, there are many ways in which model selection for time series analysis can be done, which have been discussed by Shibata [36]. A variety of statistical tests have been developed for testing different models. However, hypothesis testing may not be a practical approach, because it involves a large number of tests and significance levels. Another approach is to find a criterion which balances the *over-fitting* and *under-fitting* characteristics of the model. The general form of such a criterion is a cost function

$$C = n \log \hat{\sigma}_k^2 + L(k). \tag{2}$$

Here n is the number of observations and σ_k^2 is the estimation of the error variance. The second term $L(k)$ is

some monotonous function dependent on the number of free parameters k. It describes a cost for the model complexity and can be considered as a penalty term. The choices of $L(k)$ determine the characteristics of the criterion. Several different forms have been suggested, e.g., by Akaike [2], Schwartz [35], and Rissanen [27]. The Akaike information criterion (AIC) has been widely used in the literature for model selection, and there exist some attempts to apply it to select the neural-network structure [10]. However, the AIC was shown to be inconsistent [16], and it has a tendency to overfit models. One of the most interesting model selection methods is the minimum description length (MDL) principle [27], which has been proven consistent and used successfully in the analysis of autoregressive (AR) and autoregressive moving average (ARMA) models. Recently, there have also been some attempts to apply this criterion to neural-network size selection, and the results show for some simple problems that this method can succeed in finding the optimal network structure [17]. The predictive MDL, where the coding is done in a predictive manner, is presented in this paper, and applied to neural-networks complexity selection.

IV. Stochastic Complexity and PMDL Principle

A. Probabilistic Model, Code Length, and Stochastic Complexity

Inspired by the algorithmic notion of complexity [18], [37], [4] as well as Akaike's work, Rissanen proposed the shortest code length for the observed data as a criterion for model selection [27]. In his subsequent papers [29]–[31], this gradually evolved into stochastic complexity. The word "stochastic" means that the models, relative to which the coding ought to be done, are probabilistic rather than being defined by programs in a universal computer as in the algorithmic theory.

The objective of any modeling is to learn the behavior of the machinery generating the observations. Modeling brings together two important ideas: coding and learning, where coding is synonymous with description, and learning is virtually synonymous with extraction of properties from data. The word description presumes the existence of a language, in which the properties can be expressed. If by a language we mean a set of appropriately formed strings of recognizable symbols, we may talk about the length of a description. The short description, in particular, the shortest one, plays a special role. Therefore, it seems worthwhile to examine models and modeling problems in terms of coding systems.

Rissanen has constructed a family of probability distributions from the code lengths in a system so that they define a random process or an information source. This allows the definition of a general notion of complexity of a data string relative to such a random process. Based on this, we can construct a coding system out of a family of parametrically defined probability distributions. Such a family also results if we begin with a parametric predictor together with a prediction error measure, so that prediction becomes a form of coding, and it starts looking like "all models are probabilistic." It is precisely these distribution families that the basic idea of the shortest code length has led to concrete applications.

An important coding system with many codes of each string is defined by a family of parametric distributions, where the prior distribution may be replaced by a prefix code. Then, let B^* be the set of all finite binary strings over the binary alphabet. Now, B^* can be partially ordered by the prefix property: $a < b$, if a is a prefix of b. A coding system can be defined to be a function D: $S \rightarrow X^*$, where the domain S is a subset of B^*, and X^* is the set of all finite strings over the set X, which is the alphabet of the symbols x_i, and $x = x^n = x_1 \cdots, x_n$. Any member c_i of S such that $D(c_i) = x$ is said to be a code word of the string x. The length of c_i is the number of binary symbols in it, and written as $|c_i|$. Next, let S_n denote the inverse image of X^n under the decoding map D, i.e., the set of all code words of all data strings of length n. Let $\overline{S}_n$ denote the set of the minimal elements of S_n under the partial order, and let $\overline{S}_n(x)$ denote the subset whose elements get decoded as x. It can be seen that regardless of the number of elements in the set $\overline{S}_n(x)$, the Kraft-inequality holds [5]

$$P'(x) = \sum_{c_i \in \overline{S}_n(x)} 2^{-|c_i|} \leq 1. \tag{3}$$

Define recursively in the length of the string, $P(x^0) = 1$, with the following property:

$$P(x^{n+1}) = \frac{P(x^n)P'(x^{n+1})}{\sum_{z \in X} P'(x^n, z)} \tag{4}$$

where (x^n, z) denotes the string of length $n+1$ formed by concatenating x^n with the symbol z. It is clear that the probability function P satisfies the marginality conditions for a random process

$$\sum_{z \in X} P(z) = 1, \quad P(x^n) = \sum_{z \in x} P(x^n, z). \tag{5}$$

Then, the complexity of x is defined, relative to the coding system D, to be

$$I(x|D) = -\log_2 P(x). \tag{6}$$

For applications, the most important coding system is obtained from a class of probabilistic models

$$M = \{f(x|\theta), \pi(\theta) | \theta \in \Omega^k, \qquad k = 1, 2, \cdots\} \tag{7}$$

where Ω^k is a subset of the k-dimensional Euclidean space with nonempty interior, in which the number of parameters k may range over all natural numbers [29]. $\pi(\theta)$ is the so-called "prior" distribution. Hence, there are k "free" parameters. It must also be required that each distribution $f(x|\theta)$ satisfy the marginality conditions for a random process. With a small abstraction, $f(x|\theta)$ and $\pi(\theta)$ can be considered as densities which, if the observations consist of truncated numbers, assign probabilities to them. Similarly, for a strict coding interpretation, the parameters which are numerical data, may be truncated to some precision δ as well. Then a prefix code C can be constructed, which assigns to each truncated parameter

a vector, say $\theta(i)$, and a code word with length $L[\theta(i)]$ with the least integral upper bound of

$$-\log_2 \pi[\theta(i)] - k\log_2 \delta. \tag{8}$$

Similarly, another prefix code can be constructed, which assigns to the data the code word $C[x|\theta(i)]$ of length $L[x|\theta(i)]$, given by the least integral upper bound of

$$-\log_2 f[x|\theta(i)] - n\log_2 \epsilon \tag{9}$$

where ϵ is the precision of each data item. The function D, which decodes x out of the concatenated code words $C[\theta(i)]C[x|\theta(i)]$, defines a coding system. If the generally small excess of the integer valued code length over the negative logarithms of the probabilities is ignored, we obtain by letting the precision δ tend to zero

$$\begin{aligned} P'(x) &\approx \epsilon^n \sum_i f[x|\theta(i)]\pi[\theta(i)]\delta^k \\ &\to \epsilon^n \int_{\theta\in\Omega^k} f(x|\theta)\, d\pi(\theta) \end{aligned} \tag{10}$$

which by a further limiting process defines a marginal density $f(x|M)$ for the data. Hence, with the abstraction in terms of densities, (6) can be converted into the following form:

$$\begin{aligned} I(x|M) &= -\log_2 f(x|M) \quad \text{with} \\ f(x|M) &= \int_{\theta\in\Omega^k} f(x|\theta)\, d\pi(\theta) \end{aligned} \tag{11}$$

which is called the stochastic complexity of x, relative to the model class M.

Although the model class M includes the so-called "prior" distribution π, its role in expressing prior knowledge is not different from that of $f(x|\theta)$. In fact, the former need not be selected at all, because it is possible to construct it from the model class as a generalization of Jeffrey's prior [32]. Also, particularly important pairs of distributions of $f(x|\theta)$ and $\pi(\theta)$ are the so-called conjugate distributions, because for them the integral (11) can be evaluated in a closed form [31].

B. The Predictive Minimum Description Length Principle

The stochastic complexity (11) is an abstract quantity involving no specific model. In practice, however, it is the model that we want. There are various coding schemes, each of which gives a code length as an approximation of stochastic complexity, and thus provides us a model selection criterion. One of those approximations is the predictive minimum description length principle, where the coding is done in a predictive manner. By predictive coding we mean that the conditional density of the possible values of the "next" observation x_{t+1}can be modeled as

$$f_{k,\hat{\theta}(t)}(x_{t+1}|x^t) \tag{12}$$

where $\hat{\theta}(t) = \hat{\theta}(x^t)$ is obtained by an estimation algorithm for the parameter θ with k components. Such a density allows us to encode the observation $x_{t+1}$with the code length $-\log_2 f_{k,\hat{\theta}(t)}(x_{t+1}|x^t)$.

Generally speaking, for example, in curve fitting and related problems, the models are not primarily represented in terms of a distribution. Rather, we can use a parametric predictor $\hat{x}_{t+1} = F(x|\theta)$ as in the case of neural networks. Let $\hat{x}_{t+1}$ be the prediction output of the network, $x = (x_t, x_{t-1}, \cdots, x_{t-p+1})$ the input, and θ the array of all the weights as parameters. In addition, there is a distance function $\delta(\epsilon_{t+1})$ to measure the prediction error $\epsilon_{t+1} = x_{t+1} - \hat{x}_{t+1}$, where x_{t+1} is the target output. Such a prediction model can be immediately reduced to a probabilistic model in which the optimization of ϵ_{t+1} causes the optimization of θ. In this case, we define the conditional Gaussian distribution

$$f(x_{t+1}|x^t, \theta, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\epsilon_{t+1}^2/2\sigma^2} \tag{13}$$

where $x^t = (x_1, x_2, \cdots, x_t)$. The density (13) is then extended to a sequence by multiplication, and the total code length of n observations can be expressed as

$$-\ln f(x^n, \theta, \sigma) = \frac{1}{2\sigma^2} \sum_{t=0}^{n-1} \epsilon_{t+1}^2 + \frac{n}{2}\ln(2\pi\sigma^2). \tag{14}$$

After having fixed the model class, we have the problem of estimating the shortest code length obtainable with this class of models. Let $\hat{\theta}(x^t)$ and $\hat{\sigma}^2(x^t)$ be written briefly as $\hat{\theta}_t$ and $\hat{\sigma}_t^2$. They are the *maximum likelihood* estimates, i.e., the parameter values which minimize the code length $-\ln f(x_{t+1}|x^t, \theta, \sigma)$ for the past data. In particular

$$\hat{\sigma}_t^2 = \frac{1}{t}\sum_{i=1}^{t} \epsilon_i^2. \tag{15}$$

Therefore, the predictive code length for the data is given by

$$-\ln f(x^n|k) = \frac{1}{2}\sum_{t=0}^{n-1}\left[\frac{\epsilon_{t+1}^2}{\hat{\sigma}_t^2} + 2\ln\hat{\sigma}_t\right] + \frac{\ln}{2}\, n(2\pi) \tag{16}$$

in which $\hat{\sigma}_0$ is a suitable initial value. The choice of $\hat{\sigma}_0$ is unimportant, because it affects only the first value in the sum. In this form, the *model cost*, i.e., the code length needed to encode the model appears only implicit. That, however, cannot be avoided and is indeed included in this criterion. This can be demonstrated by rewriting (16) in another form

$$-\ln f(x^n|k) = -\ln f(x^n, \hat{\theta}_n, \hat{\sigma}_n) + \sum_{t=0}^{n-1} \ln\frac{f(x^t, \hat{\theta}_t, \hat{\sigma}_t)}{f(x^{t+1}, \hat{\theta}_t, \hat{\sigma}_t)} \tag{17}$$

where the first term is the minimized code length of (14), obtained when the parameters are replaced by the maximum likelihood estimates. But the data could not be decoded if the decoder did not know the estimated parameters with which the code was designed. Hence, the induced conditional density for each observation x_{t+1} given the past,

$$f(x_{t+1}|x^t, \hat{\theta}_t, \hat{\sigma}_t) = \frac{f(x^{t+1}, \hat{\theta}_t, \hat{\sigma}_t)}{f(x^t, \hat{\theta}_t, \hat{\sigma}_t)} \tag{18}$$

contributes the *model cost*, which is represented by the sum in (17). $\hat{\Theta}_t$ represents the neural networks' parameters, i.e.,

weights and biases. That cost term may be regarded as the accumulated errors when the model parameters are estimated from the *past* data.

Unlike the two-pass *nonpredictive* MDL principle, in which an explicitly calculated code length for the parameters is added to the negative logarithm of the likelihood function, the PMDL algorithm has the great advantage that the network parameters need not be encoded, and they can be calculated from the past string by an algorithm. The model costs are added to the prediction errors, and the *over-fitting* and *under-fitting* are penalized automatically. Applications of the PMDL principle in selecting optimal neural network for time series prediction can also be found in [12] and [20].

V. Neural-Network Predictor Optimization Using the PMDL Principle

In this section, we investigate the applications of the PMDL principle to the optimization of a neural-network predictor in a noisy power signal prediction.

The structure of our *hybrid* predictor, shown in Fig. 3, is determined by optimizing separately both the Adaline and MLP using the PMDL principle. The optimal number of input nodes of Module-1 is first determined by using a simulated power signal. The selection of Module-2 then uses the output of the optimized Module-1. The optimizations of these two modules are similar, and therefore only the latter is introduced in details.

Suppose that the MLP has p input nodes $x(t), x(t-1), \cdots, x(t-p+1)$ and q hidden nodes $z_1(t), z_2(t), \cdots, z_q(t)$. Our problem is now to optimize p and q. Here $x(t), t = 0, 1, \cdots, N-1$, are the sample values of the time series to be predicted. Therefore, we have

$$\begin{aligned} u_i(t) &= \sum_{j=1}^{p} w_{ji} x(t-j+1) + w_{0i}, \qquad i = 1, \cdots, q \\ z_i(t) &= \tanh[u_i(t)] \\ \hat{x}(t+1) &= \sum_{i=1}^{q} v_i z_i(t) + v_0. \end{aligned} \tag{19}$$

In order to apply the PMDL principle, we divided $\{x(t), t = 0, 1, \cdots, N-1\}$ into $k_{\max} = (N/d)$ consecutive segments of length d as a parameter. In case d does not divide N, the last segment is shorter. For each network candidate with p inputs and q hidden nodes, we first train the network using a *hybrid* optimization technique to minimize the quadratic function [3]

$$S_{kd} = \sum_{t=(k-1)d}^{kd-1} [x(t+1) - \hat{x}(t+1)]^2. \tag{20}$$

With the so obtained optimal weights and biases from (20), we use (19) to predict the points $x(t+1), t = kd, kd+1, \cdots, (k+1)d-1$ in the subsequent $(k+1)$th segment to obtain the squared "honest" prediction error

$$R_{(k+1)} = [x_{k+1}(t+1) - \hat{x}_{k+1}(t+1)]^2. \tag{21}$$

Here, by "honest" we mean that the parameters of the $x_o(t+1)$ predictor are only determined by the past data. The predictions of the data points in the very first segment are taken as zero. Hence the predictive code length $C_{(k+1)d}$ of the $(k+1)$th segment can be calculated by (16), where the lower and upper limits in the sum are replaced by kd and $(k+1)d-1$, respectively, and n in the last term is replaced by the segment length d. This procedure is repeated until the code lengths of all the segments are found. Adding the total predictive code lengths together, we get the accumulated code length

$$C_{\text{total}} = \sum_{i=1}^{k_{\max}} C_i \tag{22}$$

where C_i is the code length of the ith segment. In case d does not divide N, the code length of the last segment should be added to (22). Then, we calculate the per symbol code length as

$$C_{\text{per}} = C_{\text{per}}(p, q, d) = \frac{1}{N} C_{\text{total}}. \tag{23}$$

The network with the minimum C_{per} indicates the optimal predictor structure.

Due to the local minima problem with neural networks, minimization of (20) must be handled carefully. If a deterministic optimization algorithm, such as the conventional backpropagation is used, the above procedures should be repeated many times, each of which has random initialization values. The final code length of each model is the averaged code length of all the experiments. If a stochastic search algorithm, such as the simulated annealing algorithm [1], is used, the *temperature* must be decreased as slowly as possible so that a global minimum, or at least a relatively good local minimum could be found.

VI. Simulation Results

A. Off-Line Optimization of Neural-Network Predictor Structures

Due to the time-varying and mobile speed-dependent characteristics of the power response of the Rayleigh fading channel, it is not practical to optimize the predictor structure for a power signal covering the whole speed range. Therefore, we only consider the optimization of the network structures under two extreme conditions when the vehicle speed is 5 and 50 km/h with component input SNR of 0 dB. The additive noise used is zero mean white Gaussian noise.

The optimization of our neural-network predictor consists of two steps. First, Module-1 is optimized with the PMDL principle using input samples $y = (y_t, y_{t-1}, \cdots, y_{t-p+1})$ from a segment of a received power signal, as shown in Fig. 4. This time series has 1500 samples and the segment length d selected here is 125. The predictive code lengths of all model candidates in Module-1 are given in Fig. 5, with component input SNR of 0 dB at the speed of 5 km/h. It is easy to find that the optimal Adaline has seven input nodes. After the optimal structure of the Adaline is determined, the optimization of the MLP in Module-2 is then made in the same way using the output of Module-1. Under the same component SNR and speed, the code lengths of different models of MLP are

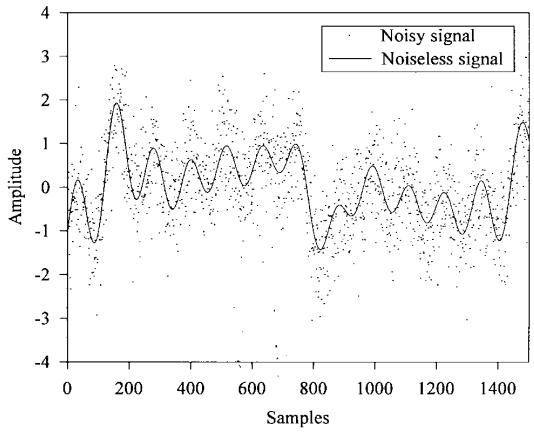

Fig. 4. A segment of in-phase component under SNR of 0 dB along with the noiseless signal at the speed of 5 km/h.

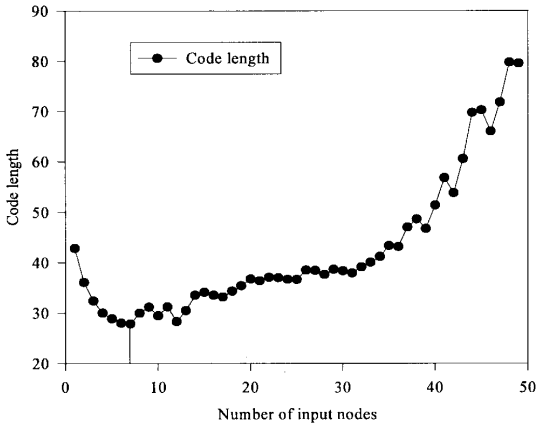

Fig. 5. Code lengths of different models of Adaline at the speed of 5 km/h with component input SNR of 0 dB.

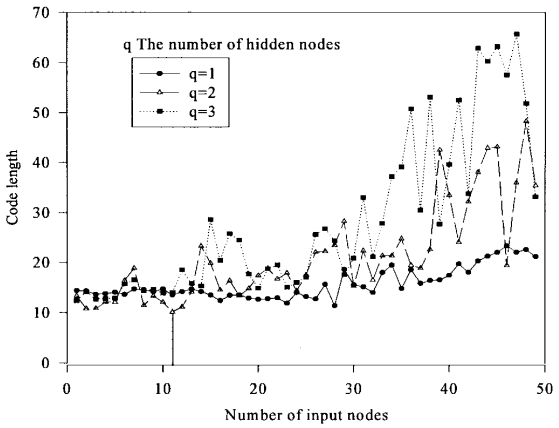

Fig. 6. Code lengths of different models of MLP at the speed of 5 km/h with component input SNR of 0 dB.

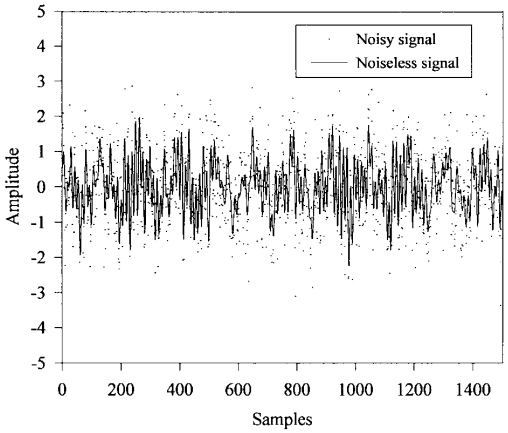

Fig. 7. A segment of in-phase component under SNR of 0 dB along with the noiseless signal at the speed of 50 km/h.

given in Fig. 6. In general, a large number of hidden nodes is rarely used because the computational complexity will increase drastically. Hence, we limit the search to a small range, i.e., $q = 1, 2, 3$. The MLP with 11 input nodes and two hidden nodes turns out to be the best structure.

Similarly, at the speed of 50 km/h, the power signal shown in Fig. 7 is used for optimization. The code lengths of different models of Adaline and MLP are given in Figs. 8 and 9, respectively. It is easy to find that the optimal Adaline has 22 input nodes and the MLP has 18 input nodes and only one hidden node. The optimized Adaline and MLP make up our optimal neural predictor.

B. Real-Time Prediction with On-Line Adaptation

As the fading signals are highly nonstationary, which is the case in mobile communication applications, the learning must be adaptive. In our predictor, the adaptation of the Adaline in Module-1 uses the computationally efficient Widrow–Hoff (LMS) algorithm [41], and the MLP in Module-2 uses an *on-line* backpropagation algorithm with a moving window. This allows the predictor to adapt new data quickly while adequately forgetting the old data [33].

The structure of our *hybrid* predictor is first optimized *off-line* using the procedures described above. The obtained optimal predictor is then used for prediction of in-phase and quadrature components of the fresh demodulated fading signal separately. At the speed of 5 km/h with in-phase component under SNR of 0 dB, the output of the predictor together with the noiseless in-phase component is shown in Fig. 10. At the speed of 50 km/h with in-phase component under SNR of 0 dB, the corresponding results are given in Fig. 11. The results with the quadrature component are similar.

We use the measure

$$\mathrm{SNR}_{\mathrm{gain}}(dB) = \mathrm{SNR}_{\mathrm{out}}(dB) - \mathrm{SNR}_{\mathrm{in}}(dB) \qquad (24)$$

as the quantitative measure of the prediction performance. The optimal neural predictor can produce about 12-dB SNR gain under the component SNR of 0 dB at the speed of 5 km/h.

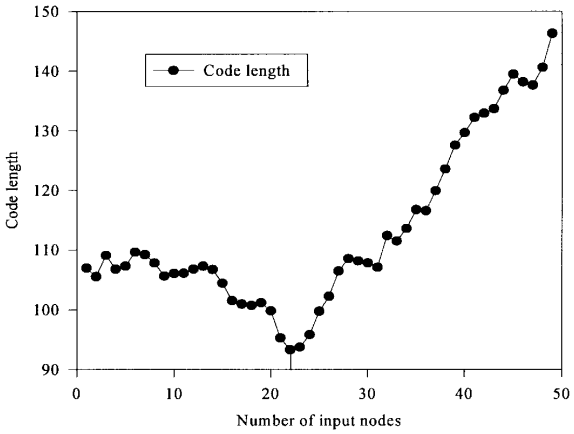

Fig. 8. Code lengths of different models of Adaline at the speed of 50 km/h with component input SNR of 0 dB.

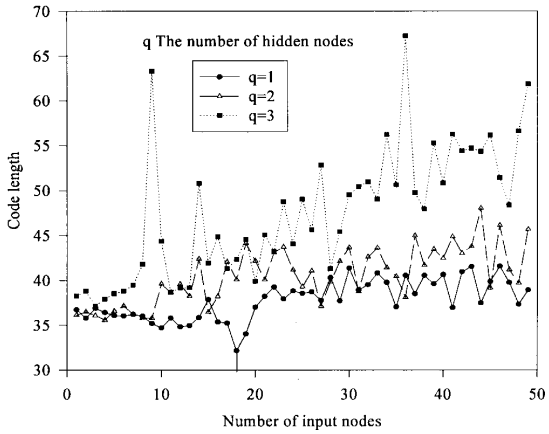

Fig. 9. Code lengths of different models of MLP at the speed of 50 km/h with component input SNR of 0 dB.

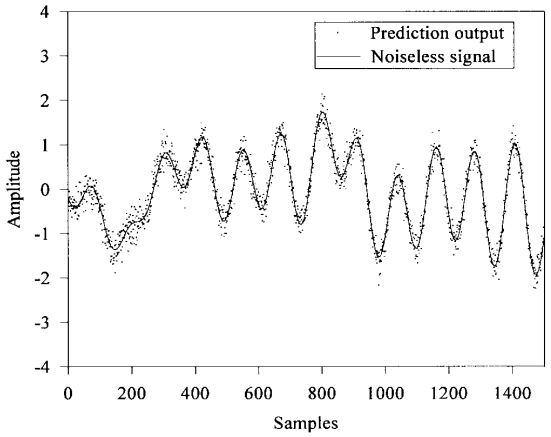

Fig. 10. The prediction output of noisy in-phase component under input SNR of 0 dB at the speed of 5 km/h along with the noiseless signal.

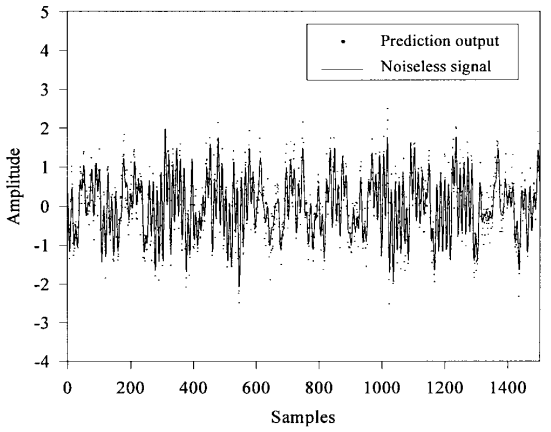

Fig. 11. The prediction output of noisy in-phase component under input SNR of 0 dB at the speed of 50 km/h along with the noiseless signal.

Similarly, about 7-dB SNR gain can be obtained at the speed of 50 km/h.

The prediction of received power signal can be obtained by summing the squared predictions of in-phase and quadrature components. The SNR gain of power signal can also be computed using (24). Thus, the obtained SNR gains are 11 and 5 dB at the low and high speed, respectively.

VII. Conclusions

In this paper, we proposed a *hybrid* neural predictor for received signal power prediction needed in high performance DS/CDMA systems. In order to get good noise attenuation and generalization capability, we used the PMDL principle to select the complexity of our neural predictor. The simulations demonstrated that the PMDL method indeed provided us valuable guidance in selecting the optimal structure of the predictor. The results show that the optimized neural predictors can give us a significant SNR gain at low speed while the improvement at high speed is clearly smaller. Notice that, in this paper, the neural predictor was optimized *off-line* using the PMDL method together with a *hybrid global* searching algorithm. However, in practice, when the optimized predictor is used for *on-line* prediction, one must make a tradeoff between the sampling rate and adaptation speed. Although the neural predictor has higher computational complexity than the conventional linear approaches, it is feasible from the application point of view, because the required sampling rate is only 1 kHz. Therefore, custom VLSI and DSP processors are the potential implementation platforms of our adaptive predictor. The presented neural predictor is a natural preprocessing stage for advanced fuzzy and neural power controllers.

Acknowledgment

The authors would like to thank the reviewers for their insightful comments and constructive suggestions. They also

thank Prof. J. Saarinen and Dr. M. Lehtokangas for helpful conversations on using the PMDL method.

REFERENCES

[1] E. Aarts and P. Van Laarhoven, *Simulated Annealing: Theory and Practice*. New York: Wiley, 1987.

[2] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.

[3] N. Baba, "A new approach for finding the global minimum of error function of neural networks," *Neural Networks*, vol. 2, pp. 367–373, 1989.

[4] C. J. Chaitin, "On the length of program for computing finite binary sequences," *J. Assoc. Comp. Mach.*, vol. 13, pp. 547–569, 1966.

[5] T. M. Cover and J. A Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[6] Y. le Cun, B. Boser, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 598–605.

[7] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Contr. Signals. Syst.*, vol. 2, pp. 303–314, 1989.

[8] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, and C. Muller, "Neural modeling for time series: A statistical stepwise method for weight elimination," *IEEE Trans. Neural Networks*, vol. 6, pp. 1355–1364, Nov. 1995.

[9] W. Finnoff, F. Hergert, and H. G. Zimmermann, "Improve generalization performance by nonconvergent model selection methods," *Neural Networks*, vol. 6, pp. 771–783, June 1991.

[10] D. B. Fogel, "An information criterion for optimal neural-network selection," *IEEE Trans. Neural Networks*, vol. 2, pp. 490–497, Sept. 1991.

[11] X. M. Gao, S. J. Ovaska, and I. O. Hartimo, "Restoration of noisy speech signal using an optimized neural-network structure," in *Proc. IEEE Int. Conf. Neural Networks*, Washington, D.C., June 1996, pp. 1841–1846.

[12] X. M. Gao, S. J. Ovaska, M. Lehtokangas, and J. Saarinen, "Modeling of speech signals using an optimal neural-network structure based on the PMDL principle," to appear in *IEEE Trans. Speech Audio Processing*.

[13] P. Heinonen and Y. Neuvo, "FIR-median hybrid filters with predictive FIR substructures," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 829–899, June 1988.

[14] L. Holmström and P. Koistinen, "Using additive noise in backpropagation training," *IEEE Trans. Neural Networks*, vol. 3, pp. 24–38, Jan. 1992.

[15] W. C. Jakes, Ed., *Microwave Mobile Communications*. New York: Wiley, 1974.

[16] R. L. Kashyap, "Inconsistency of AIC rule for estimating the order of autoregressive models," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 996–998, 1980.

[17] G. Kendall and T. Hall, "Optimal network construction by minimum description length," *Neural Computa.*, vol. 5, pp. 210–212, Feb. 1993.

[18] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems of Inform. Transmission*, vol. 1, pp. 4–7, 1965.

[19] C. Ledoux and J. Grandin, "Two original weight pruning methods based on statistical tests and rounding techniques," *Proc. Inst. Electr. Eng.: Vision, Image, and Signal Processing*, vol. 141, pp. 230–237, 1994.

[20] M. Lehtokangas, J. Saarinen, P. Huuhtanen, and K. Kaski, "Predictive minimum description length criterion for time series modeling with neural networks," *Neural Computa.*, vol. 8, pp. 583–593, Apr. 1996.

[21] N. Murata, S. Yoshizawa, and S. I. Amari, "Network information criterion for determining the number of hidden units for an artificial neural-network model," *IEEE Trans. Neural Networks*, vol. 5, pp. 865–872, Nov. 1994.

[22] S. Nowlan and G. Hinton, "Simplify neural networks by soft weight-sharing," *Neural Computa.*, vol. 4, pp. 473–493, 1992.

[23] S. J. Ovaska and O. Vainio, "Recursive linear smoothed Newton predictors for polynomial extrapolation," *IEEE Trans. Instrum. Meas.*, vol. 41, pp. 510–516, Aug. 1992.

[24] R. Reed, R. J. Marks II, and S. Oh, "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter," *IEEE Trans. Neural Networks*, vol. 6, pp. 529–538, May 1995.

[25] J. Renko, J. Heiskala, and P. Tuominen, "Neural-network control system for DS/CDMA power signal prediction," in *Proc. IEEE Nordic Signal Processing Symp.*, Espoo, Finland, Sept. 1996, pp. 139–142.

[26] B. Ripley, "Neural networks and related methods for classification," *J. Royal Statist. Soc.*, vol. 56, pp. 409–456, 1994.

[27] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[28] ——, "Universal coding, information, prediction, and estimation using predictive MDL principle," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, July 1984.

[29] ——, "Stochastic complexity," *J. Roy. Statist. Soc.*, Series B, vol. 49, pp. 223–239, 1987.

[30] ——, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, pp. 1080–1100, 1986.

[31] ——, *Stochastic Complexity in Statistical Inquiry*. Series in Computer Science, vol. 15. Singapore: World, 1989.

[32] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 40–47, Jan. 1996.

[33] V. Ruiz de Angulo and C. Torras, "On-line learning with minimal degradation in feedforward networks," *IEEE Trans. Neural Networks*, vol. 6, pp. 657–668, May 1995.

[34] D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vols. 1 and 2. Cambridge, MA: MIT Press, 1986.

[35] G. Schwartz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.

[36] R. Shibata, "Various model selection techniques in time series analysis," in *Handbook of Statistics*, E. J. Hannan, P. R. Krishnaiah, and M. M. Rao, Eds. Amsterdam, The Netherlands: North-Holland, 1985, pp. 179–187.

[37] R. J. Solomonoff, "A formal theory of inductive inference," Part I, *Inform. Contr.*, vol. 7, pp. 1–22, 1964, Part II, *Inform. Contr.*, vol. 7, pp. 224–254, 1964.

[38] J. M. A. Tanskanen, A. Huang, T. I. Laakso, and S. J. Ovaska, "Prediction of received signal power in CDMA cellular systems," in *Proc. 45th IEEE Veh. Technol. Conf.*, Chicago, IL, July 1995, pp. 922–926.

[39] O. K. Tonguz and M. M. Wang, "Cellular CDMA networks impaired by Rayleigh fading: System performance with power control," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 515–527, Aug. 1994.

[40] A. Weigend, B. Huberman, and D. Rumelhart, "Prediction the future: A connectionist approach," *Int. J. Neural Syst.*, vol. 1, pp. 193–209, 1990.

[41] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

Xiao Ming Gao received the B.S., M.S., and Ph.D. degrees in electrical engineering from Harbin Institute of Technology, Harbin, China.

From 1994 to 1995, he was a Researcher in the Department of Information Technology, Lappeenranta University of Technology, Finland. He was a Senior Researcher in Laboratory of Telecommunications Technology, Helsinki University of Technology from 1995 to 1997. His current research focuses on digital signal processing, neural networks and their applications in speech coding, adaptive filtering, and wireless communications systems.

Xiao Zhi Gao was born in Harbin, China, on February 13, 1972. He received the M.S. degree in electrical engineering from Harbin Institute of Technology, China, in 1996.

He has been working in the Institute of Intelligent Power Electronics, Helsinki University of Technology as a Researcher since the fall of 1996. His current research interests are neural networks, fuzzy logic, and reinforcement learning with their applications in intelligent control.

Jarno M. A. Tanskanen was born in Lahti, Finland, in 1968. He received the M.Sc. degree in technical physics from Helsinki University of Technology, Finland, in 1995.

Since 1994, he has been with the Laboratory of Signal Processing and Computer Technology, Helsinki University of Technology, where he is involved in research on predictive power level estimation in mobile communication systems.

Seppo J. Ovaska (M'90–SM'91) received the Diploma Engineer degree in electrical engineering from the Tampere University of Technology, Finland, in 1980, the Licentiate of Technology degree in information technology from the Helsinki University of Technology, Finland, in 1987, and the Doctor of Technology degree in electrical engineering from the Tampere University of Technology, in 1989.

He is presently a Professor of Industrial Electronics and Vice-Head of the Department of Electrical and Communications Engineering at the Helsinki University of Technology (HUT). Before joining the HUT, he was a Professor of Electronics at the Lappeenranta University of Technology, Finland. From 1979 to 1992, he held engineering, research, and R&D management positions with Kone Elevators, both in Finland and the United States. He holds nine patents in the area of elevator instrumentation. His research interests are in signal processing applications and industrial electronics.

Dr. Ovaska is an Associate Editor of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.