A PIECEWISE LINEAR RECURRENT NEURAL NETWORK STRUCTURE AND ITS DYNAMICS

Xiao Liu

GlobeSpan Technologies Inc.
Red Bank, NJ 07701

Tülay Adalı, Levent Demirekler

Department of CSEE
University of Maryland Baltimore County
Baltimore, MD 21250

ABSTRACT

We present a piecewise linear recurrent neural network (PL-RNN) structure by combining the canonical piecewise linear function with the autoregressive moving average (ARMA) model such that an augmented input space is partitioned into regions where an ARMA model is used in each. The piecewise linear structure allows for easy implementation, and in training, allows for use of standard linear adaptive filtering techniques based on gradient optimization and description of convergence regions for the step-size. We study the dynamics of PL-RNN and show that it defines a contractive mapping and is bounded input bounded output stable. We introduce application of PL-RNN to channel equalization and show that it closely approximates the performance of the traditional RNN that uses sigmoidal activation functions.

1. INTRODUCTION

Recently, neural networks have been applied to a wide range of signal processing applications because of the growing need for alternatives to the linear structure that is typically assumed. Nonlinear signal processing with neural networks has provided significant performance improvement in a variety of applications (for a recent collection of these applications see e.g. [9]), when the underlying process involves nonlinearities and/or the signal to noise ratio is poor.

The attractiveness of piecewise linear (PL) models on the other hand, stems from the fact that while they allow use of a variety of analysis and development tools that are linear. they are also good approximators of functions that are highly nonlinear. They have been effectively used in control engineering, nonlinear circuit analysis [4], and in channel equalization [2]. While, the PL models are very easy to implement, they usually require large set of parameters to describe the linear relationship in each partitioned region of a domain space. A special class of piecewise linear structures, canonical piecewise linear (CPL) models, employ a global linear model in the partitioned domain space rather than using individual linear models in each. Hence they greatly reduce the parameter storage requirement of the piecewise linear model.

In this paper, we first define RCPL function by combining the CPL function with the ARMA model [5]. RCPL mapping is the process of finding partition boundaries of a sample space in an augmented domain space where an ARMA model is used for approximation in each partitioned region. We then present a framework for study of the dynamics of RCPL function, and show that it is a contractive mapping and is bounded input bounded output stable under a given regularity condition. We then propose a particular PL-RNN structure based on the RCPL function, and show that, for a channel equalization example, the proposed structure yields robust performance which closely approximates that of a traditional RNN using sigmoidal nonlinearity.

Research supported in part by NSF grant NCR-9614236.

2. RECURRENT CANONICAL PIECEWISE LINEAR FUNCTION

We present the definition of the PL-RNN based on canonical piecewise linear function. CPL network is initially introduced for nonlinear circuit analysis [3]. CPL structures provide a desirable compromise between the approximation ability of nonlinear models and the efficiency and theoretical accessibility of the linear domain, and reduce the parameter storage requirement of piecewise linear models considerably by employing a global linear representation. In [7], CPL structure is used for adaptive equalization and it is shown that, even with very simple CPL structures, important performance gains can be achieved with respect to a linear equalizer in equalization of linear channels. The CPL function is defined as [3]:

Definition 1 (*Canonical Piecewise Linear Function*): A piecewise linear function $f\colon D \rightarrow Q$, with a compact subset $D \subset R^N$ and compact subset $Q \subset R^M$, is called a canonical piecewise linear (CPL) function, if it can be expressed by a *global* representation:

$$f(\mathbf{x}) = \mathbf{a} + \mathbf{B}\mathbf{x} + \sum_{i=1}^{\tau} \mathbf{c}_i \left|\langle \alpha_i, \mathbf{x}\rangle + \beta_i\right| \qquad (1)$$

where $\mathbf{B} \in R^{M\times N}, \mathbf{a}, \mathbf{c}_i \in R^M, \alpha_i, \mathbf{x} \in R^N$ and $\beta_i \in R$.

Based on the above definition, in [1] we study the representation and approximation ability of CPL function and show that we can always construct a partition for which a CPL representation exists. and that we can approximate any given continuous nonlinear mapping with a CPL function. We can easily use the CPL function to describe the input/output behavior of a system, e.g.. for the univariate case:

$$y(n) = f(\mathbf{x}(n)) + v(n) \tag{2}$$

where f: $R^N \rightarrow R$ is a CPL function, $M = 1$, $\mathbf{x}(n) = [x(n), x(n-1), \cdots x(n-N+1)]^T$, $x(n)$ and $y(n)$ are the system input and output sequences respectively, and $v(n)$ is the additive noise component. The model in (2) is employed as an adaptive filter in [4] based on the mean square error criterion. It is shown that this kind of filter can have better adaptive performance, especially for modeling strong nonlinearities, while providing savings in computation and implementation.

In [5], we propose a general recurrent canonical piecewise linear (RCPL) network as an extension of piecewise linear function by incorporating a nonlinear auto-regressive moving average (NARMA) model where the elements in the vector $\mathbf{x}(n)$ are predicted by the NARMA model:

$$x_i(n) = h_i(\mathbf{x}(n-1), \cdots, \mathbf{x}(n-p_1), f(\mathbf{x}(n-1)), \cdots, f(\mathbf{x}(n-p_2))$$

where $i = 1, 2, \cdots, N$. In [8], a threshold auto-regressive (TAR) model is introduced and it is successfully applied to time series analysis. Since the coefficients of the TAR model depend on the thresholds of TAR process, TAR model can be considered as a special case of CPL auto-regressive moving average model.

A modified definition of RCPL function that is more useful for its application as a nonlinear filter can be obtained as follows: Let $p_1 = p_2 = 1$, $\mathbf{x}(n)$ is the state vector of the dynamic system and $\mathbf{u}(n)$ is the input vector of the system. The RCPL function is then defined as:

Definition 3: A function f: $D_1 \times D_2 \rightarrow Q$ with sample space $D_1 \subset R^N$, $D_2 \subset R^r$, and compact subset $Q \subset R^M$ is said to be a RCPL function if it can be expressed by the global representation:

$$\begin{aligned} f(\mathbf{x}(n), \mathbf{u}(n)) &= \mathbf{a} + \mathbf{B}_0\mathbf{x}(n) + \mathbf{B}_1 f(\mathbf{x}(n-1), \mathbf{u}(n-1)) \\ &\quad + \mathbf{B}_2\mathbf{u}(n) \end{aligned} \tag{3}$$

$$\begin{aligned} x_k(n) &= a_k + \langle \mathbf{h}_{0_k}, \mathbf{x}(n-1)\rangle + \langle \mathbf{h}_{1_k}, f(\mathbf{x}(n-1), \mathbf{u}(n-1))\rangle \\ &\quad + \langle \mathbf{h}_{2_k}, \mathbf{u}(n)\rangle + \sum_{i=1}^{\tau} c_{k_i} | \langle \alpha_{0_i}, \mathbf{x}(n-1)\rangle \\ &\quad + \langle \alpha_{1_i}, f(\mathbf{x}(n-1), \mathbf{u}(n-1))\rangle + \langle \alpha_{2_i}, \mathbf{u}(n)\rangle + \beta_i | \end{aligned} \tag{4}$$

where $\mathbf{x}, \mathbf{h}_{0_k}, \alpha_{0_i} \in R^N$, $f, \mathbf{a}, \mathbf{h}_{1_k}, \alpha_{1_i} \in R^M$, $\mathbf{u}, \mathbf{h}_{2_k}, \alpha_{2_i} \in R^r$, $\mathbf{B}_0 \in R^{M\times N}$, $\mathbf{B}_1 \in R^{M\times M}$, $\mathbf{B}_2 \in R^{M\times r}$, $a_k, c_{k_i}, \beta_{k_i}, \tau \in R$, $k = 1, 2, \cdots, N$ and x_k is the kth element in $\mathbf{x}$.

By comparing definitions 1 and 3, we can see that RCPL filter is a special case of the CPL filter. The RCPL filter partitions the input signal space into finite disjoint regions and in each region, it can be represented by a FIR filter with infinite length. Therefore, the result presented in [1] on the approximation ability of the CPL function also holds for the RCPL function.

3. DYNAMICS OF RCPL

To study the dynamics of the RCPL function described by (3) and (4), we first rewrite the function in the following form:

$$\begin{aligned} \bar{\mathbf{x}}(n) &= \bar{\mathbf{a}} + \bar{\mathbf{B}}_1\bar{\mathbf{x}}(n-1) + \bar{\mathbf{B}}_2\mathbf{u}(n) \\ &+ \sum_{i=1}^{\tau} \mathbf{c}_i \left|\langle \alpha_{1_i}, \mathbf{x}(n-1)\rangle + \langle \bar{\alpha}_{2_i}, \mathbf{u}(n)\rangle + \bar{\beta}_i\right| \end{aligned} \tag{5}$$

where

$$\bar{\mathbf{x}}(n) = \begin{bmatrix} \mathbf{x}(n) \\ f(\mathbf{x}(n), \mathbf{u}(n)) \end{bmatrix} \bar{\mathbf{a}} = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{a}} + \mathbf{B}_0\mathbf{a} \end{bmatrix} \bar{\mathbf{c}}_i = \begin{bmatrix} \mathbf{c}_i \\ \mathbf{B}_0\mathbf{c}_i \end{bmatrix}$$

$$\bar{\mathbf{B}}_1 = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{B}_0\mathbf{H}_0 & \mathbf{B}_1 + \mathbf{B}_0\mathbf{H}_1 \end{bmatrix} \bar{\mathbf{B}}_2 = \begin{bmatrix} \mathbf{H}_2 \\ \mathbf{B}_2 + \mathbf{B}_0\mathbf{H}_2 \end{bmatrix}$$

$\bar{\alpha}_{1_i} = (\alpha_{0_i}\ \alpha_{1_i})^T$, $\bar{\alpha}_{2_i} = \alpha_{2_i}$, $\bar{\beta}_i = \beta_i$ $\tilde{\mathbf{a}} = (a_1, a_2, \cdots, a_N)^T$, $\mathbf{c}_i = (c_{1_i}, c_{2_i}, \cdots, c_{N_i})^T$, $\mathbf{H}_j = (\mathbf{h}_{j_1}, \mathbf{h}_{j_1}, \cdots, \mathbf{h}_{j_N})^T$, for $i = 1, 2 \cdots, \tau$, $j = 0, 1, 2$.

We can then use the definition given above. to show that the RCPL function is bounded for bounded inputs.

Theorem 1: For the RCPL function defined by (3) and (4), assume that the input vector $\mathbf{u}(n)$ is bounded and the parameters satisfy the following condition: If there exists an $\varepsilon_0 \in (0, 1)$ such that

$$||\bar{\mathbf{B}}_1|| + \sum_{i=1}^{\tau} ||\bar{\mathbf{c}}_i||\ ||\bar{\alpha}_{1_i}|| \leq 1 - \varepsilon_0 \tag{6}$$

then, there is a real number d, such that for all $K \geq d$, the ball $D(K) = \{\mathbf{x} : ||\mathbf{x}|| \leq K\}$ is invariant under (3) and (4).

Proof of the theorem is given in [6]. Note that, the notation introduced here, in equation (5), is not the same as the one used in [6]. The proof of the theorem, however, follows the same procedure with this new definition. The definition given in (5) provides a convenient framework to study dynamics of RCPL function which we also use to prove the following:

Theorem 2: The map that defines the RCPL function (3) and (4) is a contractive mapping if the condition given in (6) is satisfied.

Proof: Let $k(\mathbf{x}) = \bar{\mathbf{a}} + \bar{\mathbf{B}}_1\bar{\mathbf{x}} + \bar{\mathbf{B}}_2\mathbf{u}(n) + \sum_{i=1}^{\tau} \bar{\mathbf{c}}_i | \langle \bar{\alpha}_{1_i}, \mathbf{x}\rangle + \langle \bar{\alpha}_{2_i}, \mathbf{u}(n)\rangle + \bar{\beta}_i|$ then,

$$\begin{aligned} k(\mathbf{x}_1) - k(\mathbf{x}_2) = \bar{\mathbf{B}}_1(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) + \sum_{i=1}^{\tau} \bar{\mathbf{c}}_i (| \langle \bar{\alpha}_{1_i}, \mathbf{x}_1\rangle + \langle \alpha_{2_i}, \mathbf{u}(n)\rangle \\ + \bar{\beta}_i | - \left|\langle \bar{\alpha}_{1_i}, \mathbf{x}_2\rangle + \langle \bar{\alpha}_{2_i}, \mathbf{u}(n)\rangle + \bar{\beta}_i\right|) \end{aligned}$$

and by using (6), we get

$$\begin{aligned} ||k(\mathbf{x}_1) - k(\mathbf{x}_2)|| &\leq (\, ||\bar{\mathbf{B}}_1|| + \sum_{i=1}^{\tau} ||\bar{\mathbf{c}}_i||\ ||\bar{\alpha}_{1_i}||\,)\, ||\mathbf{x}_1 - \mathbf{x}_2|| \\ &\leq (1 - \varepsilon_0)||\mathbf{x}_1 - \mathbf{x}_2|| \end{aligned}$$

where $\varepsilon_0 \in (0, 1)$.

Theorem 2 shows that $k(\cdot)$ is a contractive mapping whenever (6) is satisfied. Thus, after receiving input vector $\mathbf{u}(n)$, which is assumed to be bounded, the function will always reach a unique equilibrium regardless of its initial state $\mathbf{x}_0$.

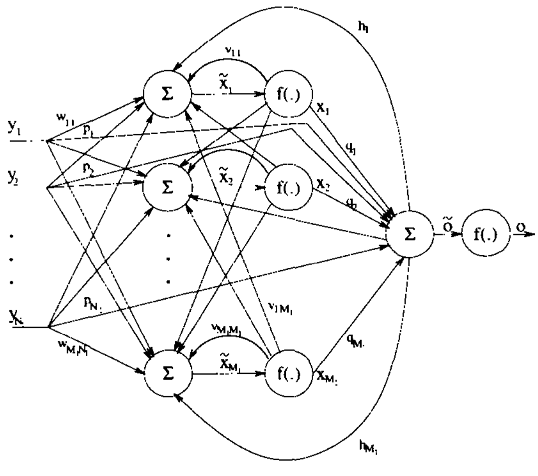

Figure 1: Recurrent Neural Network Structure

4. A PIECEWISE LINEAR RNN STRUCTURE

Based on the definition of the RCPL function given by (3) and (4), we define the piecewise linear RNN as follows.

Let $M = 1$, $N = M_1$, $r = N_1$, $\tau = 2M_1$, $\mathbf{u}(n) \equiv \mathbf{y}(n) = [y_1(n), \cdots, y_{N_1}(n)]^T$, $\mathbf{x}(n) = [x_1(n), \cdots, x_{M_1}(n)]^T$, $\tilde{o}(n) = f(\mathbf{x}(n), \mathbf{y}(n))$, where $\mathbf{y}(n)$ is the input vector and $\tilde{o}(n)$ is the output of the network and $x_k(n)$, $k = 1, 2, \cdots M_1$ is the output of hidden node k.

For the parameters in (5), we choose $\mathbf{a}=\mathbf{0}$, $\mathbf{B}_0=(q_1, q_2, \cdots, q_{M_1})$, $\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{B}_2=(p_1, p_2, \cdots, p_{N_1})$ and for the ones in (6), $a_k = 0$ (which is the kth element of $\mathbf{a}$), $\mathbf{h}_{0_k} = \mathbf{0}$, $\mathbf{h}_{1_k}=h_k$, $\mathbf{h}_{2_k} = \mathbf{0}$, $c_{k_i}=1/8$ and $\beta_i=4$, $i=1, \cdots, M_1$ and $c_{k_i}=-1/8$ and $\beta_i=-4$, $i=M_1+1, \cdots, 2M_1$. $\alpha_{0_{i_k}}=(v_{k1}, v_{k2}, \cdots, v_{kM_1})^T$, $\alpha_{1_{i_k}}=\mathbf{0}$, $\alpha_{2_{i_k}}=(\omega_{k1}, \omega_{k2}, \cdots, \omega_{kN_1})^T$, $i=1, \cdots, 2M_1$ and $k=1, \cdots, M_1$. The structure is expressed by the following equations:

$$o(n) = g(\tilde{o}(n)), \quad \tilde{o}(n) = \sum_{i=1}^{M_1} q_i x_i(n) + \sum_{i=1}^{N_1} p_i y_i(n) \qquad (7)$$

$$x_k(n) = g(\tilde{x}_k(n)) \qquad (8)$$

$$\tilde{x}_k(n) = h_k \tilde{o}(n-1) + \sum_{i=1}^{M_1} v_{ki} x_i(n-1) + \sum_{i=1}^{N_1} \omega_{ki} y_i(n) \qquad (9)$$

for $k = 1, 2, \cdots, M_1$.

The network defined by (7)-(9) is shown in Figure 1. Note that the PL-RNN structure we have defined above involves the introduction of a nonlinearity at the output which restricts the output of the PL-RNN to the interval [-1,1]. It is observed that this modification to the general RCPL structure given in (3) and (4) has improved the performance and robustness of the network by pushing the network parameters into the region that will satisfy the condition given by (6) early on during training.

The activation function for PL-RNN is chosen as:

$$g(s) = \frac{|s+4| - |s-4|}{8}$$

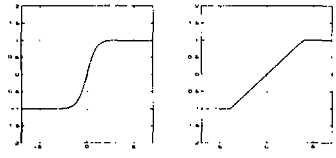

Figure 2: Activation Function $g(\cdot)$ for RNN and PL-RNN

and as $g(s) = \tanh(s)$ for the traditional RNN by using the same structure, shown in Figure 1. These two activation functions are plotted in Figure 2.

If we assume that $d(n)$ is the desired response and select the mean square error $J(n) = E\{e^2(n)\} = E\{(d(n) - o(n))^2\}$ as the cost function, we can obtain the learning algorithm by steepest descent minimization of $J(n)$. The filter coefficient updates are given by:

$$\begin{aligned} q_k(n+1) &= q_k(n) + c(n)x_k(n) && (10)\\ p_j(n+1) &= p_j(n) + c(n)y_j(n) &&\\ h_k(n+1) &= h_k(n) + c(n)q_k(n)g'(\tilde{x}_k(n))\tilde{o}(n-1) &&\\ v_{kk'}(n+1) &= v_{kk'}(n) + c(n)q_k(n)g'(\tilde{x}_k(n))x_{k'}(n-1) &&\\ w_{kj}(n+1) &= w_{kj}(n) + c(n)q_k(n)g'(\tilde{x}_k(n))y_j(n) && (11) \end{aligned}$$

for k, $k'=1, \cdots, M_1$ and $j=1, \cdots, N_1$, where $c(n)=\mu e(n) g'(\tilde{o}(n))$, μ is the step-size and $g'(s)$ is the derivative of the nonlinear function evaluated at s, given by

$$g'(s) = \begin{cases} 0.25 & \text{if } -4 < s < 4 \\ 0 & \text{otherwise} \end{cases}$$

for PL-RNN and $g'(s) = 1 - \tanh^2(s), \forall$ s, for the RNN.

5. SIMULATION RESULTS

The performance of the two RNN structures introduced in section 4 are compared for an equalization example. They are used in the equalization of the following channel:

$$y(n) = y_l(n) - 0.9y_l^3(n) + \eta(n) \qquad (12)$$

where the multipath component is given by $y_l(n) = x(n) + 0.5x(n-1)$, $x(n)$ is the input signal, $y(n)$ is the channel output and $\eta(n)$ denotes the zero mean white Gaussian noise. The input signal $x(n)$ is assumed to be an independent sequence taking values from $\{-1, 1\}$ with equal probability. The dimension of the observation vector N_1 is chosen as 2 to be able to visualize the decision boundaries, and the number of nodes in the hidden layer M_1 is chosen as 5. The weights are initialized to uniformly distributed random values between -0.1 and 0.1 and the algorithm given by Eqs. (10)-(11) is used for training. Several learning parameters are tested at 15dB signal to noise ratio (SNR) (SNR is defined in terms of the *input* signal power to the noise variance). The results that indicate similar performance for the PL-RNN and the RNN equalizers are shown in Figure 3. For the results shown in Figures 4 and 5, the step-size μ is chosen as 0.7 for PL-RNN and 0.5 for the RNN, a value which yields the best average performance for each at 15 dB. To obtain the SNR curve shown in Figure 4, the two equalizers are trained with 10000 samples and then tested

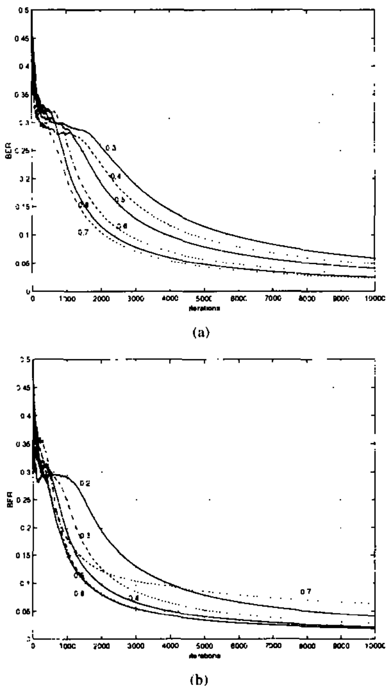

Figure 3: Convergence curves with varying step-sizes for (a) PL-RNN; (b) RNN

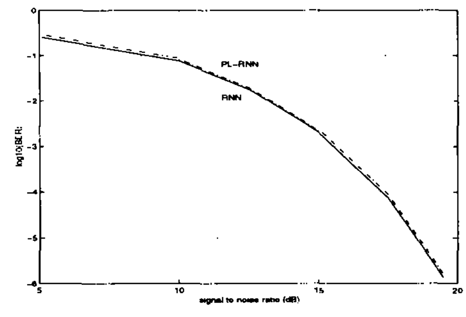

Figure 4: SNR curves for RNN and PL-RNN

for 20000 samples for 50 independent realizations. The two structures again yield very comparable performance.

The decision regions described by each equalizer also proved to be similar. In Figures 5a and 5b, we only show the decision regions formed by the PL-RNN equalizer. Since for a recurrent structure, the boundaries depend on the value of the previous network output, we assume that after convergence, the network output will be mostly $\{-1, 1\}$ and consider these two cases. Figures 5a and 5b show the decision boundaires obtained by using PL-RNN at 15 dB for $o(n-1) = -1$ and $o(n-1) = 1$ respectively. They also show the optimal Bayesian decision boundary for these given network output, $o(n-1)$, values. Finally, the optimal Bayesian decision boundary for the given channel is plotted in Figure 5c. As the channel has a memory of length 1, the observation at time n depends only on the current and previous value of the input, i.e., on $x(n)$ and $x(n-1)$. At time n, let $x(n-1) = -1$, then, assuming noise is zero, the observation $y(n-1)$ can either be 1.5375 (if $x(n-2) = -1$) or -0.3875 (if $x(n-2) = 1$) for the given channel model. Hence at time n, there are four possible observation pairs $(y(n), y(n-1))$: $(0.3875, 1.5375)$, $(0.3875, -0.3875)$, $(1.5375, 1.5375)$ and $(1.5375, -0.3875)$, instead of 8. Note that the first two pairs corresponds to $x(n)=1$ and the last two pairs corresponds to $x(n)=-1$. This results in a Bayesian based decision boundary plotted shown as the straight line in Figure 5a. Note that the boundaries obtained using PL-RNN are very close to the optimal ones. Similar discussion holds for Figure 5b.

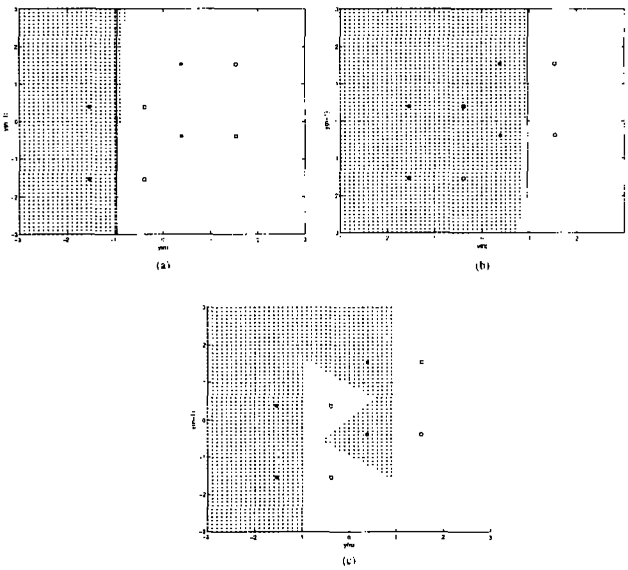

Figure 5: Decision boundary formed by PL-RCPL equalizer and the optimal Bayesian boundary when (a) $o(n-1) = -1$ and (b) $o(n-1) = 1$; (c) Optimal Bayesian decision boundary for the given channel. ("o" denotes -1 and "*" 1)

6. REFERENCES

[1] T. Adalı and X. Liu, "Canonical piecewise linear network for nonlinear adaptive filtering and its application to blind equalization," *Signal Processing*, vol. 61, no. 2, pp. 145-155, 1997.

[2] C. P. Callender, C. F. N. Cowan, and S. Theodoridis, "The piecewise linear adaptive filter for channel equalisation in digital communications," in *Signal Processing VI: Theories and Applications*, 1992, pp. 143-146.

[3] L. O. Chua and A. Deng, "Canonical piecewise-Linear networks," *IEEE Trans. Circuits Syst.*, vol. 35, No. 1, pp. 101–111, Jan. 1988.

[4] J. Lin and R. Unbehauen, "Adaptive nonlinear digital filter with canonical piecewise-linear structure," *IEEE Trans. Circuits Syst.*, vol. 37, no. 3, pp. 347–353 , March, 1990.

[5] X. Liu and T. Adalı, "Recurrent canonical piecewise linear network for blind equalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Munich, Germany, April 1997, pp. 3213-3216.

[6] X. Liu and T. Adalı, "Recurrent canonical piecewise linear network: theory and application in blind equalization," in *Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, Amelia Island, FL, Sep. 1997, pp. 446-455.

[7] C. J. Pantaleon-Prieto and A. R. Figueiras-Vidal, "A new adaptive equalizer based on the canonical piecewise linear model," in *Proc. 1994 European Conf. on Signal Processing (EUSIPCO)*, equalization," Edinburgh, Scotland, UK, vol. 3, pp. 736-739, Sept. 1994.

[8] H. Tong, *Non-linear Time Series: A Dynamical System Approach*, Clarendon Press Oxford, 1990.

[9] J. Principe, L. Giles, N. Morgan, E. J. Wilson, eds. *Neural Networks for Signal Processing VII, Proc. of the IEEE Workshop*, Amelia Island, FL, New York, NY: IEEE Press, 1997.