Transactions Briefs

L_p Norm Back Propagation Algorithm for Adaptive Equalization

Sammy Siu, Ching-Haur Chang, and Che-Ho Wei

Abstract— **The l_p norm back propagation algorithm for perceptron based adaptive equalization is analyzed taking account of possible numerical problem encountered when $p < 1$. Two methods are proposed to overcome the numerical problem. Computer simulations indicate that simultaneous improvement in convergence rate and bit-error-rate (BER) performance can be achieved by using $p < 2$.**

I. Introduction

In recent years, neural network has emerged as a powerful tool for nonlinear adaptive filtering [1]–[3]. The algorithm based on stochastic gradient with l_2 norm error criterion is often used in training the neural network. It is known that the l_2 norm gives equal weights for all errors during the weight updating process and may result in a slow convergence. The use of l_p norm with $p < 2$ gives small weights to large errors and hence reduces the influence of aberrant noise [4]–[9], while it gives large weights to small errors. This property can be used to improve the tracking capability of the network. Taking account of possible numerical problem for $p < 1$, the l_p norm back propagation algorithm with application to adaptive equalization is analyzed in this paper. Two methods are proposed to overcome the numerical problem.

II. Analysis of the L_p Norm Back Propagation Algorithm

Consider a back propagation model of an m-layer perceptron network with $m \in \{1, 2, \ldots, M\}$, $t_i(n)$ being the ith desired signal, and $v_i^{(M)}(n)$ being the ith estimated signal at the output layer [10], [11]. Then the l_p norm error function [9] is given as $\bar{E}(n) = p^{-1} \sum_j |t_j(n) - v_j^{(M)}(n)|^p$ where the factor p^{-1} is padded for mathematical convenience. Hereafter, $|e_i(n)|$ is used to stand for $|t_i(n) - v_i^{(M)}(n)|$ for simplicity in our later expressions. The ith error signal for the output layer is then given as

$$\delta_i^{(M)}(n) = \operatorname{sgn}(e_i(n))|e_i(n)|^{p-1} v_i'^{(M)}(n) \tag{1}$$

where $v_i'^{(M)}(n)$ denotes the derivative of $v_i^{(M)}(n)$. If the sigmoid function used is $f(x) = (1 - e^{-x})/(1 + e^{-x})$, then $v_i'^{(M)}(n) = (1 - v_i^{2(M)}(n))/2$. Eq. (1) indicates that changing the power metric p rescales $|e_i(n)|$ in $\delta_i^{(M)}(n)$. Fig. 1 depicts $|e_i(n)|^{p-1}$ versus $|e_i(n)|$ for different p. It indicates that $|e_i(n)|^{p-1}$ gives small weights for large $|e_i(n)|$ and large weights for small $|e_i(n)|$ for $p < 2$, and vice versa for $p > 2$. However, for $p = 2$, $|e_i(n)|^{p-1} = |e_i(n)|$. The algorithm then corresponds to the standard back propagation algorithm. For $p \geq 1$, $|e_i(n)|^{p-1}$ is bounded for all $|e_i(n)|$. However, for $p < 1$, $|e_i(n)|^{p-1}$ becomes very large as $|e_i(n)|$ is approaching zero. This may result in numerical problem during the weight updating process. The following two methods can be used to solve the problem. In Method I, $|e_i(n)|$ is replaced by θ whenever $|e_i(n)| < \theta$, where θ is a small positive value. In Method II, p is switched from $p_1 < 1$ to $1 \leq p_2 \leq 2$ whenever $|e_i(n)| < \theta$. Defining $\bar{\delta}_i^{(M)}(n) = \delta_i^{(M)}(n)|_{p=2}$, then $\delta_i^{(M)}(n) = |e_i(n)|^{p-2} \bar{\delta}_i^{(M)}(n)$. The increments of the weights for the output and hidden layers are then written as

$$\Delta w_{ij}^{(M)}(n+1) = \frac{\bar{\eta}}{|e_j(n)|^{2-p}} \bar{\delta}_j^{(M)}(n) v_i^{(M-1)}(n), \tag{2}$$

and

$$\Delta w_{ij}^{(m-1)}(n+1) = \frac{(1 - v_j^{2(m-1)}(n))}{2} \sum_l \frac{\bar{\eta}}{|e_l(n)|^{2-p}} \times \bar{\delta}_l^{(m)}(n) w_{lj}^{(m)}(n) v_i^{(m-2)}(n), \tag{3}$$

where $\bar{\eta}$ is the learning gain at $p = 2$. Similar results can be applied to update the threshold levels. It is seen from (2) or (3) that the effective learning gain can be expressed as $\eta_i(n) = \bar{\eta}/|e_i(n)|^{2-p}$. Assuming that $v_i^{(M)}(n)$ is uniformly distributed in [-1, 1] [12] and $t_i(n)$ belongs to the signal set {-1, 1}, the effect of p on the learning gain for both training mode and decision-directed mode is given below.

Training Mode: Correct decisions make $|e_i(n)|$ to be uniformly distributed in [0, 1]. In this case, $E[|e_i(n)|^{2-p}] = (3 - p)^{-1}$ for $p \leq 2$ and $E[|e_i(n)|^{p-2}] = (p - 1)$ for $p \geq 2$. Thus the expected learning gain is given as

$$E[\eta_i(n)] = \eta_{cd} = \begin{cases} (3-p)\bar{\eta}, & p \leq 2 \\ (p-1)^{-1}\bar{\eta}, & p \geq 2 \end{cases} \tag{4}$$

where the subscript cd denotes "correct decision." It is seen that the expected learning gain increases by a factor of $(3-p)$ for $p \leq 2$, while decreases by $(p-1)^{-1}$ for $p \geq 2$. Incorrect decisions make $|e_i(n)|$ to be uniformly distributed in [1, 2]. Therefore, $E[|e_i(n)|^{2-p}] = (3-p)^{-1}(2^{3-p} - 1)$ for $p \leq 2$, $E[|e_i(n)|^{p-2}] = (p-1)^{-1}(2^{p-1} - 1)$

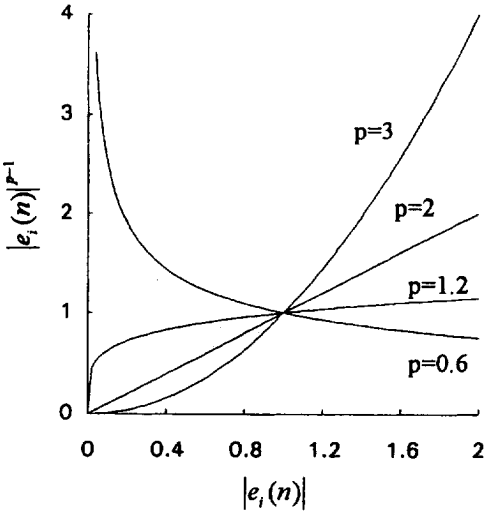


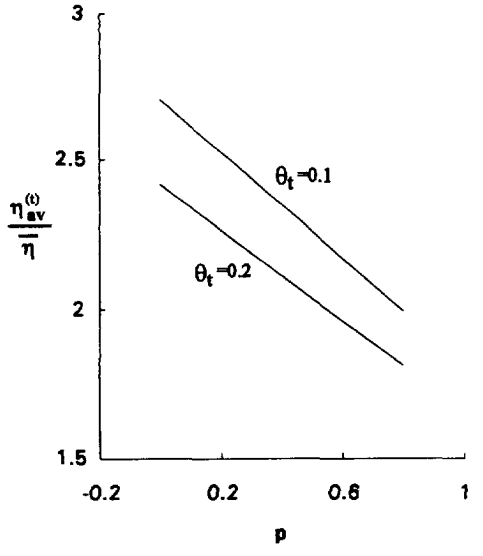
Fig. 1. $|e_i(n)|^{p-1}$ versus $|e_i(n)|$ for different p.

Manuscript received January 3, 1994; revised August 15, 1994 and February 20, 1995. This paper was recommended by Associate Editor J. Zurada.

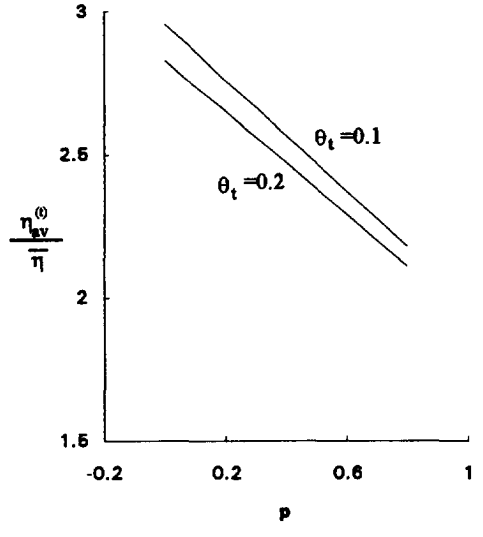
S. Siu is with the Telecommunication Laboratories, MOTC, at Chung-Li 32099, Taiwan, R.O.C.

C. H. Chang and C. H. Wei are with the Institute of Electronics at National Chiao-Tung University, Hsin-Chu 30050, Taiwan, R.O.C.

IEEE Log Number 9412390.

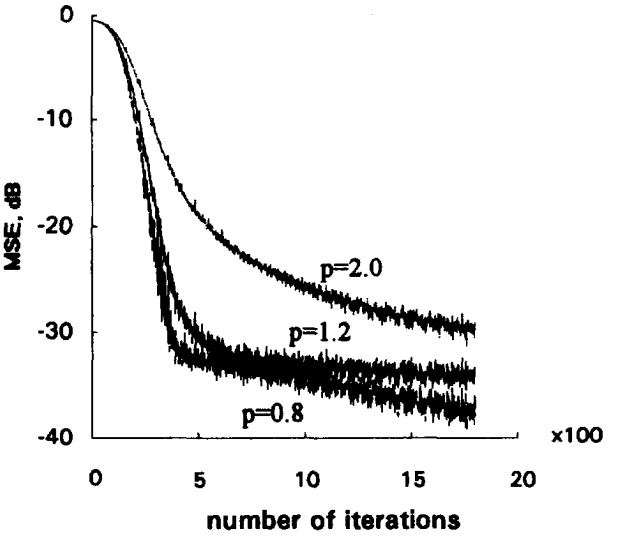


(a)

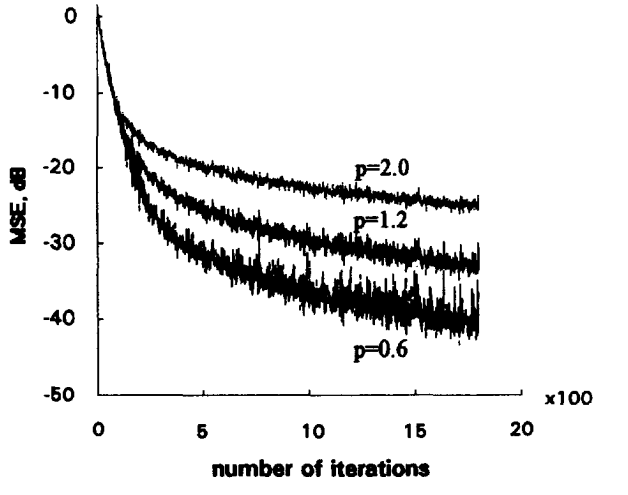


(b)

Fig. 2. Learning gain enhancement $(\eta_{\text{av}}^{(t)}/\bar{\eta})$ as a function of p for (a) Method I and (b) Method II under $P(e) \approx 0$. For Method II, the value of p is switched to $p = 1.0$ whenever $|e_i(n)| \leq \theta_t$.



(a)



(b)

Fig. 3. Learning curves for the two DFE's using different p and SNR= 20 dB. (a) MLP: $\bar{\eta} = 0.1$, $\beta = 0.05$, and $\theta_t = 0.2$ for $p = 0.8$ (b) PPS: $\bar{\eta} = 0.1$, $\beta = 0.05$, and $\theta_t = 0.2$ for $p = 0.6$.

for $p \geq 2$, and

$$E[\eta_i(n)] = \eta_{id} = \begin{cases} (3-p)(2^{3-p}-1)^{-1}\bar{\eta}, & p \leq 2 \\ (p-1)^{-1}(2^{p-1}-1)\bar{\eta}, & p \geq 2 \end{cases} \quad (5)$$

where the subscript id denotes "incorrect decision." Let $P(e)$ be the error probability. Then the average value of $E[|e_i(n)|^{2-p}]$ or $E[|e_i(n)|^{p-2}]$ can be found by assigning a weight of $P(e)$ to that when incorrect decisions are made and a weight of $(1 - P(e))$ to that when correct decisions are made. Thus $E[|e_i(n)|^{2-p}] = (3-p)^{-1}[1 + P(e)(2^{3-p} - 2)]$ for $p \leq 2$ and $E[|e_i(n)|^{p-2}] = (p-1)[1+P(e)(2^{p-1}-2)]^{-1}$ for $p \geq 2$. The average learning gain is then obtained as

$$\eta_{\text{av}}^{(t)} = \begin{cases} (3-p)[(1+P(e)2^{3-p}-2)]^{-1}\bar{\eta}, & p \leq 2 \\ (p-1)^{-1}[1+P(e)(2^{p-1}-2)]\bar{\eta}, & p \geq 2 \end{cases} \quad (6)$$

where the superscript t denotes "training mode." For small $P(e)$, $\eta_{\text{av}}^{(t)}$ approximates $(3-p)\bar{\eta}$ for $p \leq 2$ and approximates $(p-1)^{-1}\bar{\eta}$ for $p \geq 2$.

Decision-Directed Mode: $|e_i(n)|$ is uniformly distributed in [0, 1] and hence the average learning gain is obtained as

$$\eta_{\text{av}}^{(d)} = \begin{cases} (3-p)\bar{\eta}, & p \leq 2 \\ (p-1)^{-1}\bar{\eta}, & p \geq 2 \end{cases} \quad (7)$$

where the superscript d denotes "decision-directed mode."

The effect of θ on the learning gain for the two methods used to overcome the numerical problem is analyzed below.

For Method I, $|e_i(n)|$ is uniformly distributed in $[\theta, 1]$ when correct decisions are made in training mode and when the equalizer is operated in decision-directed mode. The average learning gains for training mode and decision-directed mode are, respectively, found as

$$\eta_{\text{av}}^{(t)} = (3-p)(1-\theta_t)\{(1-\theta_t^{3-p}) - P(e) \times [(1-\theta_t^{3-p}) - (1-\theta_t)(2^{3-p}-1)]\}^{-1}\bar{\eta}, \quad (8)$$

[1]Hereafter p_1/p_2 is used to denote that p is switched from p_1 to p_2 whenever $|e_i(n)| \leq \theta$.

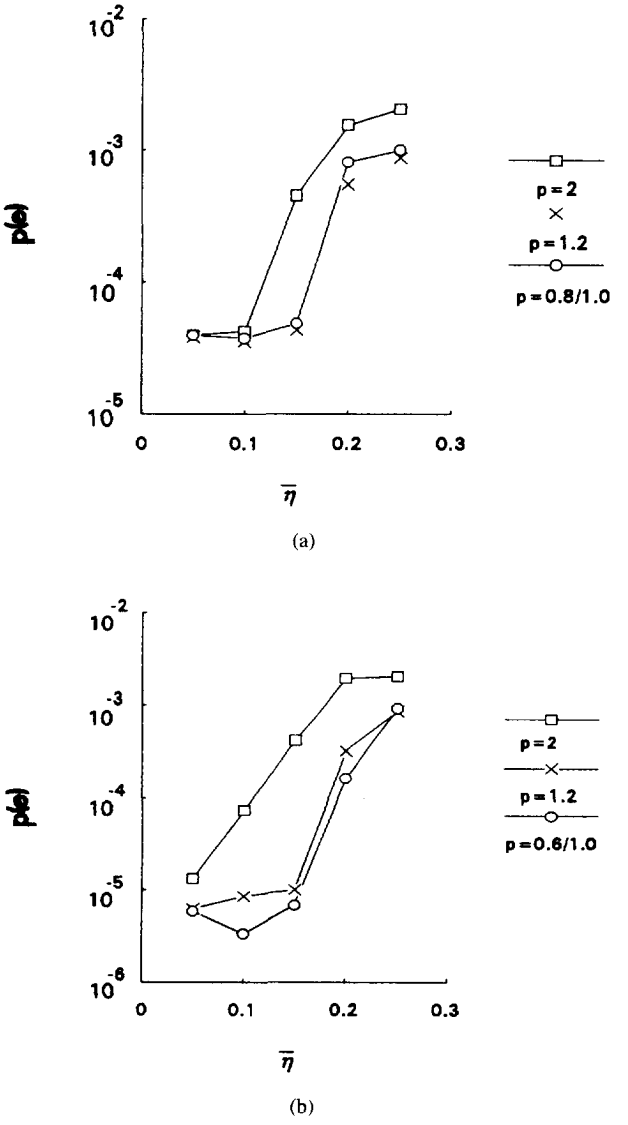


Fig. 4. BER performances for the two DFE's using different $\bar{\eta}$. (a)MLP: $\theta_t = 0.3$, $\theta_d = 1.0$ for $p = 0.8/1.0^1$, SNR = 16 dB (b) PPS: $\theta_t = 0.2$, $\theta_d = 1.0$ for $p = 0.6/1.0$, SNR = 18 dB.

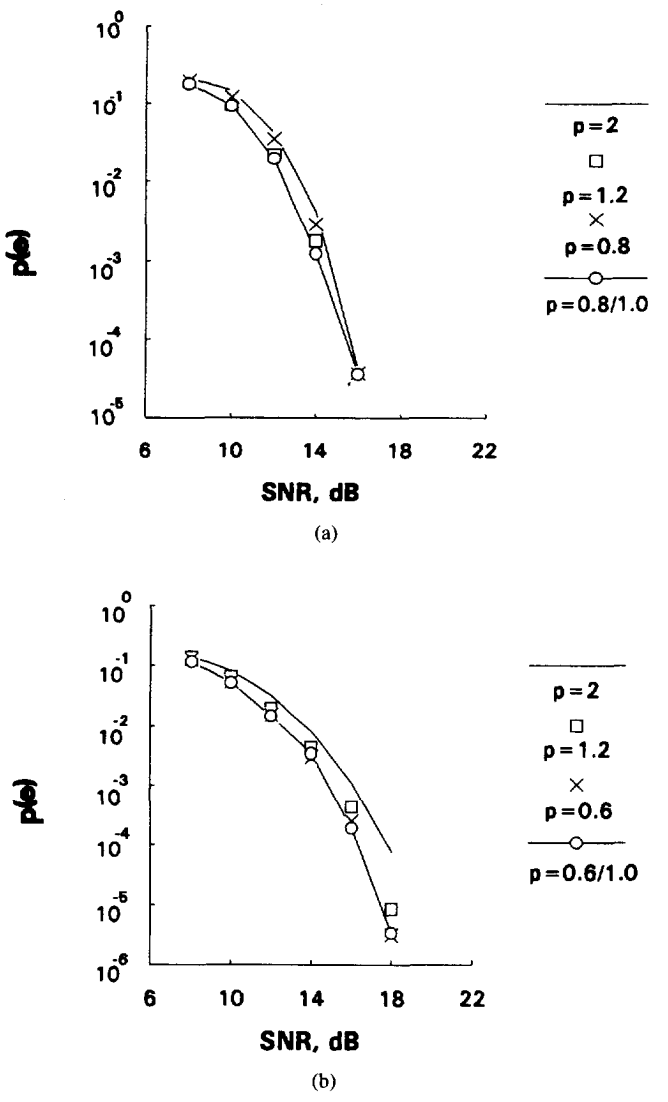


Fig. 5. BER performances for the two DFE's as a function of SNR. (a) MLP: $\bar{\eta} = 0.1$, $\beta = 0.05$, $p = 0.8/1.0$, and $\theta_t = 0.3$, $\theta_d = 1.0$ (b) PPS: $\bar{\eta} = 0.1$, $\beta = 0.05$, $p = 0.6/1.0$, and $\theta_t = 0.2$, $\theta_d = 1.0$.

and

$$\eta_{\text{av}}^{(d)} = (3-p)(1-\theta_d)(1-\theta_d^{3-p})^{-1}\bar{\eta} \tag{9}$$

where the subscripts t and d denote "training mode" and "decision-directed mode." For small θ_t or θ_d, both $\eta_{\text{av}}^{(t)}$ (when $P(e) \approx 0$) and $\eta_{\text{av}}^{(d)}$ approximate $(3-p)\bar{\eta}$.

For Method II, $1 \le p_2 \le 2$ is used when $|e_i(n)|$ is distributed in $[0, \theta]$, while $p_1 < 1$ is used when $|e_i(n)|$ distributed in $[\theta, 1]$ or [1,2]. Therefore, the average learning gains in training mode and decision-directed mode are respectively obtained as

$$\eta_{\text{av}}^{(t)} = \left\{[1-P(e)]\left(\frac{\theta_t^{3-p_2}}{3-p_2} + \frac{1-\theta_t^{3-p_1}}{3-p_1}\right) + P(e)\frac{2^{3-p_1}-1}{3-p_1}\right\}^{-1}\bar{\eta}, \tag{10}$$

and

$$\eta_{\text{av}}^{(d)} = \left\{(3-p_2)^{-1}\theta_d^{3-p_2} + (3-p_1)^{-1}(1-\theta_d^{3-p_1})\right\}^{-1}\bar{\eta}. \tag{11}$$

For $\theta_t^{3-p_1} \ll 1$ or $\theta_d^{3-p_1} \ll 1$, both $\eta_{\text{av}}^{(t)}$ (when $P(e) \approx 0$) and $\eta_{\text{av}}^{(d)}$ tend to $(3-p_1)\bar{\eta}$. For $\theta_d = 1$, $\eta_{\text{av}}^{(d)}$ becomes $(3-p_2)\bar{\eta}$. Defining $\eta_{\text{av}}^{(t)}/\bar{\eta}$ as the learning gain enhancement, Fig. 2 depicts $\eta_{\text{av}}^{(t)}/\bar{\eta}$ as a function of p for the two methods. Method II actually reflects small errors to $\delta_i^{(M)}(n)$, instead of limiting small errors by θ. It is therefore seen that $\eta_{\text{av}}^{(t)}/\bar{\eta}$ is larger for Method II than that for Method I. Also, both $\eta_{\text{av}}^{(t)}$ and $\eta_{\text{av}}^{(d)}$ increase with decreasing θ_t or θ_d for both methods. However, the value of θ can only be determined empirically.

III. Computer Simulations

Two perceptron based decision feedback equalizers (DFE) are employed in the simulations. One is based on a multilayer perceptron (MLP) with 9 neurons in hidden layer 1, 3 neurons in hidden layer 2, and 1 neuron in the output layer [2], and the other based on a third-order polynomial-perceptron structure (PPS) [13], [14]. The channel chosen is of the form $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The input to the channel is random in {-1, 1}. The channel noise is white Gaussian with zero-mean. The number of taps in the feedforward section of the DFE is 4 and in the feedback section is 1. Fig. 3 shows the learning curves for the two structures using different p and SNR = 20 dB. It indicates that both structures converge faster for smaller p. Fig. 4 shows the BER performances for the two structures using different $\bar{\eta}$. It is seen that substantial performance improvement

is obtained with increasing $\bar{\eta}$. This is because the noise caused by larger $\bar{\eta}$ can be suppressed by using $p < 2$. Fig. 5 shows the BER performances for the two structures as a function of SNR. It can be seen that better BER performance can be achieved by using smaller p and this is especially true for the PPS structure.

IV. Conclusion

The l_p norm back propagation algorithm for adaptive equalization, taking account of possible numerical problem encountered when $p < 1$, is analyzed. Two methods are proposed to overcome the numerical problem. Simulation results indicate that simultaneous improvement in convergence rate and BER performance can be obtained by using $p < 2$.

References

[1] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Application of multilayer perceptrons as adaptive channel equalizers," *Adaptive Systems in Control and Signal Processing 1989*, Selected Papers from the 3rd IFAC Symposium, Glasgow, U.K. pp. 573–578, Apr. 19–21, 1989.

[2] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalization using neural network structures and performance comparison with standard architecture," *IEE Proc. Pt. I*, vol. 137, no. 4, pp. 221–225, Aug. 1990.

[3] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Process.*, vol. 39, pp. 1874–1884, Aug. 1991.

[4] S. J. Hanson and D. J. Burr, "Minkowski-ξ back propagation: Learning in connectionist models with non-Euclidean error signals," in *AIP Conf. Proc. Neural Inform. Process. Syst.*, Denver, CO, 1987, pp. 348–357.

[5] P. J. Huber, *Robust statistics*. New York: Wiley, 1981.

[6] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley, 1986.

[7] R. Yarlagadda, J. B. Bednar, and T. L. Watt, "Fast algorithms for l_p deconvolution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, Nn. 1, pp. 174–182, 1985.

[8] S. Siu and C. F. N. Cowan, "Adaptive equalization using the l_p back propagation algorithm," *IEE Int. Conf. Artificial Neural Networks*, Bournemouth, U.K., Nov. 18–20, 1991.

[9] J. Schroeder, R. Yarlagadda, and J. Hersey, "L_p normed minimization with applications to linear predictive modeling for sinusoidal frequency estimation," *Signal Process.*, vol. 24, no. 2, pp. 193–216, 1991.

[10] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986. vol. 1, ch. 8.

[11] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[12] A. K. Rigler, J. M. Irvine, and T. P. Vogl, "Rescaling of variables in back propagation learning," *Neural Networks*, vol. 4, pp. 225–229, 1991.

[13] C. H. Chang, S. Siu, and C. H. Wei, "A decision feedback equalizer utilizing higher-order correlation," in *Proc. 1993 IEEE ISCAS*, Chicago, IL, May 1993, pp. 707–710.

[14] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalization using a polynomial-perceptron structure," *IEE Proc., Pt. I*, vol. 137, no. 5, pp. 257–264, Oct. 1990.

Closed-Form Impulse Responses of Discrete-Domain Multidimensional Filters

Dave Jin and L. T. Bruton

***Abstract*— It is known that useful two-dimensional (2-D) and three-dimensional (3-D) discrete-domain recursive transfer functions may be designed by applying the MD bilinear transformation to the continuous-domain transfer functions of prototype MD inductance-resistance networks. Closed-form expressions are derived for the impulse responses of these 2- and 3-D discrete-domain filters.**

I. Introduction

Prototype three-dimensional (3-D) inductance-resistance continuous-domain networks, having Laplace transform transfer functions of the form

$$T_1(s_1, s_2, s_3) = \frac{R}{R + s_1 L_1 + s_2 L_2 + s_3 L_3}, \tag{1}$$

have been shown [1] to be useful for the design of 3-D discrete-domain recursive filter transfer functions by applying the triple bilinear transformation to (1). In particular, such filters can be used to selectively enhance 3-D linear trajectory (LT) and 3-D plane wave (PW) space-time signals.

The demonstrated usefulness of such filters in image processing has motivated this work, in which closed-form expressions are derived for the impulse responses, $h(\mathbf{n})$ (where the boldface $\mathbf{n}$ represents the integer m-tuple $n_1, n_2, \ldots, n_m$), of both the 2-D and 3-D LT filters.

Closed-form expressions for $h(\mathbf{n})$ are not generally available for MD filters, primarily because of the lack of a Fundamental Theorem of Algebra for multivariate polynomials[1]. Therefore, MD transfer functions cannot be expanded by the method of partial fraction expansion as in the 1-D case. However, closed-form expressions are available [2] for *purely first-order* m-D z-transform transfer functions. This result has led to a deeper understanding of the 2-D stability of transfer functions obtained via the 2-D bilinear transformation [3].

The availability of algebraic expressions for $h(\mathbf{n})$ facilitates further research on the input-output properties of LT and PW 3-D recursive filters, including issues relating to MD convolution and MD stability. The transfer functions considered here are more general than in [2]. In this brief, we present the derivation for the impulse response of the 2-D LT filter using the method of residues. The extension to the 3-D case is straightforward but very lengthy, and is given in [4].

Manuscript received October 25, 1993; revised August 19, 1994 and October 5, 1994. This work was supported by Micronet, the Federal Network of Centres of Excellence, and the Natural Sciences and Engineering Research Council. This paper was recommended by Associate Editor W.-S. Lu.

The authors are with the Department of Electrical and Computer Engineering, The University of Calgary, Calgary, Alberta, Canada T2N 1N4.

IEEE Log Number 9414331.

[1] For single-variable polynomials, the Fundamental Theorem of Algebra allows any Nth degree single-variable polynomial to be factored into N first degree factors.