FIR Equalizer using Genetic Programming

Ching-Ya Huang, Shih-Yen Tsai and Te-Jen Su

Abstract—**The main duty of communication systems is to assure to provide adequate message interchange, through a certain channel, between a transmitter and a receiver. The distortion takes place in the process of transmitting message, and it usually leads to severe degradation. Consequently we need a device named equalizer filters to recover the desired information from the received signal. In this paper, a FIR equalizer based on the GP approach to recover the transmitted signal is proposed. In addition, the equalizer coefficient will be estimated by the GP algorithm.**

***Index Terms*—Genetic Programming (GP), Finite Impulse Response(FIR),equalizer**

I. Introduction

The GP can evolve programs automatically by simulating the evolutionary mechanism on computers [1]. We can through the essences of GP, like robustness, domain independence and ability to search for satisfying solutions in solving complicated nonlinear problems, this study hoped that the evolved GP models could have a better applicability and accuracy of evaluations, and easily obtain the optimal coefficients of transfer function of the equalizer. The learning cure demonstrates our method is efficient and reliable.

II. Genetic programming

A. Expression Tree

Each program or individual on the population is generally represented as a tree composed of functions and terminals appropriate to the problem domain. The function set may contain standard arithmetic operators, mathematical functions, logical operators, and domain-specific functions. The terminal set usually consists of variables and constants. The simple expression is represented as shown in Figure 1 where the function set is {+, -, sin, exp}, and the terminal set is {x, 0.2} [3].

B. Tree-Simplification

In order to increase the speed of calculating, we must reduce the quantity of the component to improve the complexity. In the following, we will discuss how to simplify the tree.

Linear-in-parameters models can be formulated as:

$$y(t) = \theta_0 + \sum_{k=1}^{M} \theta_k p_k \left(x_1(t), \ldots, x_n(t) \right) \tag{1}$$

where $p_1 \ldots p_M$ are nonlinear functions, and $\theta_0, \ldots, \theta_M$ are model parameters.

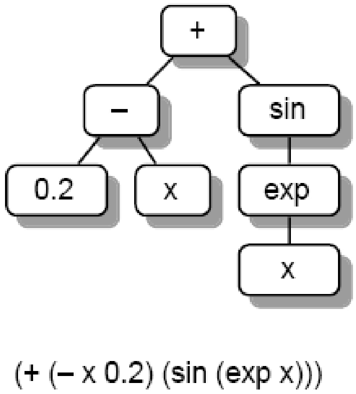

Figure 1: An example of GP Tree

The function terms were determined by decomposing the tree starting from the root as far as reaching non-linear nodes (nodes which are not '+' or '-') [4] [5].

In Fig 2: The root node is a '+' operator, so it is possible to decompose the tree into two subtrees: 'p' and 'p_1' trees. The root node of the 'p' tree is again a linear operator, so it can be decomposed into 'p_2' and 'p_3' trees. The root node of the 'p_1' tree is a nonlinear node ('*') so it cannot be decomposed, so finally the decomposition procedure results in three subtrees: 'p_1', 'p_2' and 'p_3'.

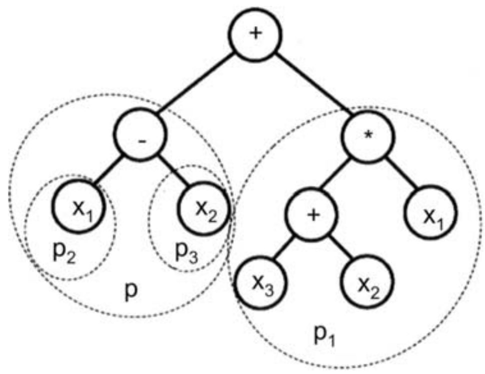

Figure 2: Decomposition of a tree to function terms

The parameters are assigned to the model after 'extracting' the p_i function terms from the tree. Least Squares Method can be used for the parameters identification.

The optimal $\Theta = [\theta_1, \ldots, \theta_M]^T$ parameter vector can be analytically calculated:

C. Y. H. is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan 807, R.O.C. (e-mail: duckdh@yahoo.com.tw).

S. Y. T. is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan 807, R.O.C. (e-mail: 1095320108@cc.kuas.edu.tw).

T. J. S. is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan 807, R.O.C. (e-mail: sutj@cc.kuas.edu.tw).

$$\Theta = (P^T P)^{-1} P^T y \quad (2)$$

Where $y = [y(1), \ldots, y(N)]^T$ is the measured output vector, and the P regression matrix is:

$$P = \begin{pmatrix} p_1(x(1)) & \cdots & p_M(x(1)) \\ \vdots & \ddots & \vdots \\ p_1(x(N)) & \cdots & p_M(x(N)) \end{pmatrix} \quad (3)$$

The compact matrix form corresponding to the linear-in-parameters model is

$$y = P\Theta + e \quad (4)$$

where P is the regression matrix, Θ is the parameter vector, e is the error vector [6].

In order to inspect the individual contributions of each term, we use OLS method. Assumes that the regression matrix P can be orthogonally decomposed as $P=WA$, where A is an M*M upper triangular matrix and W is an N*M matrix with orthogonal columns in the sense that $W^TW=D$ is a diagonal matrix. N is the length of y vector and M is the number of repressors.

After this decomposition one can calculate the OLS auxiliary parameter vector g as

$$g = D^{-1} W^T y \quad (5)$$

where g_i is the corresponding element of the OLS solution vector. The output variance $y^T y/N$ can be explained as:

$$y^T y = \sum_{i=1}^{M} g_i^2 \omega_i^T \omega_i + e^T e \quad (6)$$

Thus the error reduction ratio of P_i term can be expressed as:

$$ERR = \frac{g_i^2 \omega_i^T \omega_i}{y^T y} \quad (7)$$

We calculate the error reduction ratio values of the terms. Based on these values, we eliminate the less significant terms. We can transform the trees to simpler trees, but their accuracy is close to the original trees [6].

III. Fitness Function

The most important concept of GP is the fitness function. In this paper, we define the fitness function is based on the mean square error (MSE) between the calculated and the measured output values, Fitness is as:

$$\frac{1}{N} \sum_{k=1}^{N} \left(y(t) - \sum_{k=1}^{M} \theta_k p_k \left(x_1(t), \ldots, x_n(t) \right) \right)^2 \quad (8)$$

where N is the number of the data-points used for the identification of the model.

IV. FIR Equalizer with GP

A system called Finite Impulse Response (FIR) means its output will gradually decay to zero in a finite duration as long as its input duration is finite. Because there is no feedback in this structure, so it's always stable and quantization errors are usually less critical for FIR filter than for IIR filter.

The relationship between input and output of an FIR filter can be formulated as

$$y[n] = \sum_{l=0}^{R-1} k_l \cdot x[n-l] \quad (9)$$

where k[n] is impulse response of the filter, x[n] is input sample, and R is number of taps.

A typical transmission system is represented as shown in Figure 3 [6].

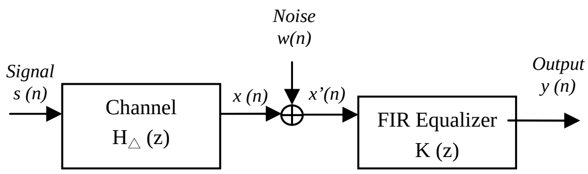

Figure 3: typical transmission system.

where s (n) is the transmitted symbol stream, w (n) is the additive noise, supposed zero-mean and Gaussian distributed, H△ (z) is the discrete equivalent of the linear time-invariant communication channel subjected to parameter uncertainty belonging to a convex bounded polyhedral domain. [8]

To realize the FIR Equalizer, we compute z-transform of to obtain. [7]

$$Y(z) = \sum_{l=0}^{R-1} k_l \cdot z^{-l} X[z] \quad (10)$$

where $k_0, k_1, \ldots, k_{R-1}$ are equalizer coefficients, and z is a delay element.

And the state-space model for the FIR equalizer:

$$K(z) = \frac{Y(z)}{X(z)} = \sum_{l=0}^{R-1} k_l \cdot z^{-l} \ \ldots \text{with order R-1} \quad (11)$$

The discrete data sequence, s (n) passes through the linear time-invariant uncertain channel, H△ (z), and the observation sequence, $x'(n)$ is then formed by the addition of an unknown measurement disturbance, $w(n)$, with the output of the communication uncertain channel, H△ (z). [8]

In modern interference-limited cellular telephony systems, the main error source is the noise. We can through GP algorithms to improve the performance of the FIR equalizer. The Model of the AWGN channel with GP is represented as shown in Figure 4. We will easily obtain the optimal coefficients of transfer function of the equalizer by GP.

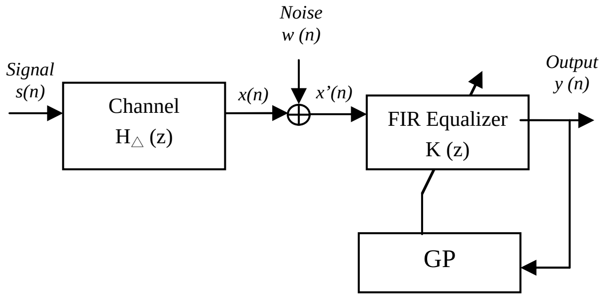

Figure 4 : Model of the AWGN channel with GP

V. Experiment Results

In this section, we use GP method to compare equalizer experiment, which use LMI as [8]. Suppose there is 1% relative normal distributed noise was added to the channel. We consider the uncertain linear discrete-time FIR channel chosen as

$$H_{\Delta}(z) = 0.04\text{-}5z^{-1} + 7z^{-2} - 0.21z^{-3}\text{-}0.5z^{-4} + 0.72z^{-5} + 0.36z^{-6} + 0.21z^{-7} + 20.03z^{-8} + 0.07z^{-9} \qquad (12)$$

and we set the FIR equalizer length is five and delay is chosen as d=6. First, we assume that the given system is perfectly known. By combining the above solving procedure with the Genetic Programming approach, the optimal γ value obtained is 0.461, with the associate equalizer coefficients
k_0=0.0357,k_1=0.0963,k_2=0.0465,k_3=0.075,k_4=0.0116

The state-space model is

$$K(z) = k_0 + k_1 z^{-1} + k_2 z^{-2} + k_3 z^{-3} + k_4 z^{-4} \qquad (13)$$

A flowchart of GP is in Figure 5. [9]

Step 1: Creation of initial population

We generate forty initial chromosomes of random compositions of the functions and terminals.

Step 2: Tree-Simplification

To avoid nonlinear in parameter models, the parameters must be removed from the set of terminals.

We use LS method to identification parameter…see Eqs. (2) (3) (4).

And use OLS method to eliminate the less significant terms…see Eqs. (5) (6) (7).

Step 3: Evaluate Population Fitness

The fitness function reflects the goodness of a potential solution which is proportional to the probability of the selection of the individual. According to the fitness value, we selects the parents of the next generation and determines which individuals survive from the current generation. In here, we use "Tournament selection".

Step 4: Use Genetic Operation

The genetic operations include crossover and mutation. We set mutation rate is 0.3, and crossover rate is 0.7.

Step 5: New generation

After the genetic operations are performed on the current population, the new generation replaces the now-old generation. This iterative process of measuring fitness and performing the genetic operations is repeated over many generations. The run of GP terminates when the termination criterion is satisfied.

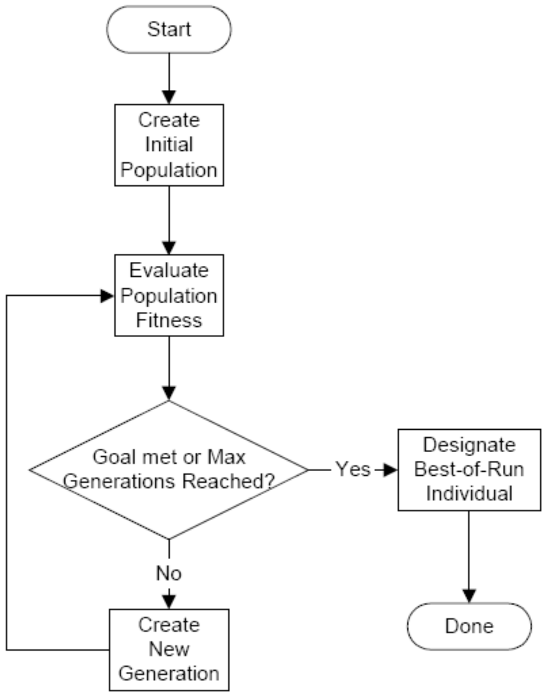

Figure 5: flowchart of GP

Table 1: Parameters of GP in the examples

Objective function	$H_{\Delta}(z) = 0.04\text{-}5z^{-1} + 7z^{-2} - 0.21z^{-3}\text{-}0.5z^{-4} + 0.72z^{-5} + 0.36z^{-6} + 0.21z^{-7} + 20.03z^{-8} + 0.07z^{-9}$
Function set	{+, -, *}
Terminal set	{ z^{-1} , z^{-2} , z^{-3} ,z^{-4}}
Population size	40
generation	50
Max tree depth	6
Type of selection	Tournament selection
Type of mutation	point-mutation
Type of crossover	one-point
Crossover rate	0.7
mutation rate	0.3

As these steps, we can obtain FIR equalizer coefficient as Table 2, which is to recover the transmitted signal.

Table 2: Equalizer coefficients with GP

Equalizer coefficients	Equalizer with GP
k_0	0.04187
k_1	0.08544
k_2	0.04112
k_3	0.07781
k_4	0.00987

The equalizer performance is described by the probability of misclassification with respect to the signal-to-noise ratio (SNR). With the assumption of independent identically distributed (i.i.d.) sequence the SNR can be defined as

$$SNR = 10\log_{10}\frac{\sigma_s^2}{\sigma_e^2} \qquad (14)$$

Where σ_s^2 represents the signal power and σ_e^2 is the variance of the Gaussian noise. BER comparison at two algorithms shown in Fig. 6.

The BER comparison at the two algorithms of GP is shown in Fig. 5.

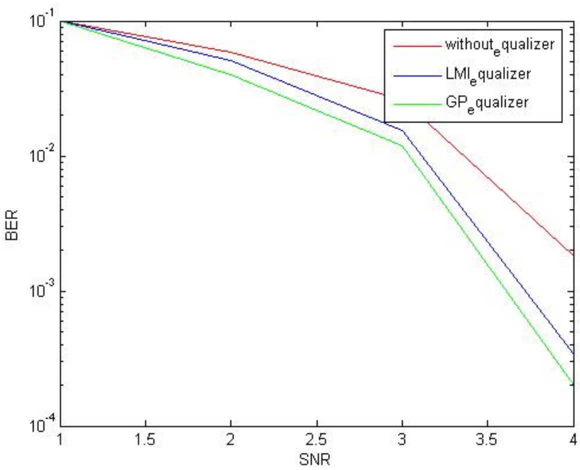

Figure 6: BER comparison at the two algorithms

These simulation results show that the BER of the solutions was lower for the GP method, and also show that the noise is suppressed effectively.

VI. Summary

Compared to LMI approach [8] which have some specific constraints , the proposed GP approach algorithm obtains an equalizer that work well whether the shape of the channel coefficient matrix is square or fat — therefore , a more generalized approach is developed. Simulations demonstrate the satisfied performance of this FIR equalizer.

VII. References

[1] Chen, Y. H. & Sun, K. T. “A Study of Genetic Programming on Cooperative Model for an Ant Colony”, 2000 International Computer Symposium—Artificial Intelligence Workshop, National Chung-Cheng University, Taiwan. (2000).

[2] Luke, S. (2000). “Issues in Scaling Genetic Programming”, Breeding Strategies, Tree Generation, and Code Bloat. PhD thesis, Department of Computer Science, University of Maryland, A. V. Williams Building, University of Maryland, College Park, MD 20742 USA.

[3] Ferreira, C. “Gene expression programming: a new adaptive algorithm for solving problems”, Complex Syst 2001, 13, pp 87–129.

[4] Sette, S.; Boullart, L. “Genetic Programming: principles and applicaionts”, Engineering Applications of Artificial Intelligence 2001, 14, pp 727–736.

[5] X Hong, C J Harris : "Nonlinear Model Structure Design and Construction using Orthogonal Least Squares and D-optimality Design", IEEE Transaction on Neural Networks, Vol. 13, No.5, pp 1245-1250 (2002)

[6] R. R. de F. Attux, L. N. de Castro, F. J. Von Zuben, J. M. T. Romano (2003b) “A Paradigm for Blind IIR Equalization Using the Constant Modulus Criterion and an Artificial Immune Network,” Proceedings of the IEEE NNSP 2003, Toulouse, França.

[7] D. R. Bull, and D. H. Horrocks, “Primitive operator digital filter,” IEE Proc. Circuits, Devices and Systems, vol.138, pp.401-412, Mar. 1991

[8] Su, T.J., Lin, S.Y., Jong, G. J. and Chen, Y. C., “Robust FIR Equalization of Uncertain Channels-LMI Approach,, ” Electrical Engineering (Springer),2004,SCI

[9] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, Cambridge, MA (1992).