



FAU Institutional Repository

<http://purl.fcla.edu/fau/fauir>

This paper was submitted by the faculty of [FAU's Harbor Branch Oceanographic Institute](#).

Notice: ©1999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This manuscript is available at <http://ieeexplore.ieee.org/> and may be cited as: Caimi, F. M., & Wang, D. (1999). Joint channel and data estimation: genetic algorithm based blind equalization. *Oceans'99 MTS/IEEE: Riding the crest into the 21st century*. (Vol. 2, pp. 931-937). Washington, DC: Oceans'99 MTS/IEEE Conference Committee. doi:10.1109/OCEANS.1999.804998

Joint Channel and Data Estimation: Genetic Algorithm Based Blind Equalization

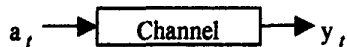
Frank M. Caimi*, Dali Wang
Department of Electrical Engineering/Engineering Division
Harbor Branch Oceanographic Institution, Inc.
5600 US 1 North
Ft. Pierce, Florida 34946

*current affiliation: Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL 34946
caimi@ee.fit.edu

Abstract--A genetic algorithm (GA) based blind equalization method is presented for joint channel and data estimation. The GA is an optimization technique using natural selection and evolutionary processes that searches for solutions of the problem through phases of evaluation, reproduction, crossover, and mutation repeatedly. The most important advantages of these algorithms are parallel search capability, convergence to global optimum, and reduced problem-dependence. Different schemes for each of above phases have been considered in achieving better results. Computer simulations have shown that for all the assumed situations, the presented algorithms establish the channel model and decode the transmitted data with satisfactory precision. Additional improvements for the GA approach are also presented, including GA coupled with a gradient based algorithm, neural network based calculation for the improvement of computational efficiency, and the genetic programming based modeling for nonlinear channels.

I. INTRODUCTION

The problem of blind equalization is illustrated in the following figure. We observe the



output signal sequence $\{y_t\}$ of an unknown channel, from which we want to determine (recover) the transmitted input signal sequence $\{a_t\}$. The most common approach for blind deconvolution is to filter the output by an estimate of the inverse channel followed by some decision device. If the input contains a known training sequence, it is straightforward to estimate a finite impulse response (FIR) model of the channel. The input signal can then be recovered by applying the Viterbi algorithm to the estimated model. However in many applications this approach cannot be used (e.g., when the channel is sufficiently time varying).

If we assume that the input signal belongs to a finite alphabet, there are only a finite number of possible input sequences. By considering each of these as a training sequence the finite FIR models are estimated, each associated with one input sequence. By associating a cost function to the estimates, namely the a posteriori probabilities of the input sequences, we can determine which one is the most likely and thus use this as an estimate. But here we face several obstacles. First, the number of possible input sequences increases exponentially with time. That means an unacceptable computational load for the real-time applications. Another obstacle is that in this approach the cost function adopted is usually multimodal having a local minima. Because of this, the most popular searching methods (e.g. using gradient-based algorithms) often converge to the local minima, rather than obtaining a proper solution. Furthermore, since the search algorithm is serialized, the search efficiency is usually very low and results in poor real-time performance.

Here we present a new algorithm for blind equalization, using a genetic algorithm (GA) [1]-[3] for equalization. The approach realizes joint channel and data estimation with parallel search ability and global optimum convergence. The GA uses the principles of evolution, natural selection, and genetics from natural biological systems in a computer algorithm to simulate evolution. Essentially, the GA is an optimization technique that performs a parallel, stochastic, but direct search to evolve the most fit population. In fact, it evaluates more than one area of the search space and can discover more than one solution to a problem. It provides a stochastic optimization method where if it "gets stuck" at a local optimum, it tries to simultaneously find other parts of the search space and "jump out" of the local optimum to a global one.

Now, assume the output of the channel is described as follows:

$$y_t = b_1 a_t + b_2 a_{t-1} + \dots + b_n a_{t-n+1}, \quad (1)$$

where n is the estimate of the system order. Let n -dimensional vector b contain the unknown FIR coefficients $b = (b_1, b_2, \dots, b_n)^T$, and regression vector $\varphi_t = (a_t, a_{t-1}, \dots, a_{t-n+1})^T$, we have

$$y_t = \varphi_t^T b. \quad (2)$$

Furthermore, collect input sequence into vector $A_t = (a_1, a_2, \dots, a_t)^T$, outputs into vector $Y_t = (y_1, \dots, y_t)^T$, from equation (2) we get the matrix equation:

$$Y_t = \Phi_{t,n} b, \quad (3)$$

where $\Phi_{t,n}$ is the Toeplitz matrix containing the input sequence A_t :

$$\Phi_{t,n} = \begin{bmatrix} a_n & a_{n-1} & \dots & a_1 \\ a_{n+1} & a_n & \dots & a_2 \\ \dots & \dots & \dots & \dots \\ a_t & a_{t-1} & \dots & a_{t-n+1} \end{bmatrix}. \quad (4)$$

Thus, the blind equalization problem is just to solve the constrained nonlinear (bilinear) system (3) with respect to b and $\Phi_{t,n}$, where input signal $\{a_i\}$ belongs to its limit alphabet.

It can be proven [4] that in order to solve the problem, the sequence $\{a_i\}$ must be persistent exciting (PE) of order $2n-1$ at time t , i.e. $(\Phi_{t,2n-1})$ has full column rank. In this case, vector b can be solved from (3) such that:

$$b = \Phi_{t,n}^+ Y_t, \quad (5)$$

where $\Phi_{t,n}^+ = (\Phi_{t,n}^T \Phi_{t,n})^{-1} \Phi_{t,n}^T$ denotes the pseudo-inverse of $\Phi_{t,n}$. And so:

$$Y_t = \Phi_{t,n} \Phi_{t,n}^+ Y_t. \quad (6)$$

For a more realistic case, we should consider the time varying channel disturbed by noise, and the equation (3) should be replaced by:

$$Y_t = \Phi_{t,n} b_t + E_t, \quad (7)$$

where E_t is the additive noise vector. In such a case equation (6) would no longer hold, and should be replaced by:

$$(\Phi_{t,n}, b_t) = \arg \min (E_t^T E_t) = \arg \min (Y_t - \Phi_{t,n} b_t)^T (Y_t - \Phi_{t,n} b_t), \quad (8)$$

where Y_t is the known measured values. If $\Phi_{t,n}$ is also known, we can get b_t via Least Square method:

$$b_t = \Phi_{t,n}^+ Y_t. \quad (9)$$

So the problem is how to find $\Phi_{t,n}$ that minimizes the cost function

$$J(\Phi_{t,n}) = (Y_t - \Phi_{t,n} b_t)^T (Y_t - \Phi_{t,n} b_t) \\ = (Y_t - \Phi_{t,n} \Phi_{t,n}^+ Y_t)^T (Y_t - \Phi_{t,n} \Phi_{t,n}^+ Y_t) \quad (10)$$

in an efficient and recursive way.

II. GENETIC ALGORITHM BASED BLIND EQUALIZATION

Genetic algorithms (GAs) [1]-[3] provide a basis for one of the most powerful search methods used for optimization. The method itself is based on the natural selection process which was first outlined in the Darwinian principle of survival of the fittest which stated that species evolve through their fittest genus until the species reaches its evolutionary optimum.

Simulating this process, GA searches the problem solution space by using strings to represent the parameters to be optimized. These strings are called chromosomes and their individual integer components are called genes. A number of these chromosomes can be initially generated at random. This is called the population and the number of chromosomes is the population size. This initial population is the first in the generation and may be evaluated by the following steps.

Firstly, the individual chromosomes are evaluated to obtain a measure of how close the simulated responses are to the desired responses. In general, the evaluation is achieved by using a cost function that is minimized in order to obtain optimum results. The cost function is defined by the difference between the desired and actual responses obtained from the evaluations, and the GA looks for an optimum with a low cost.

At this stage in the GA new chromosome solutions are created by the operations of Reproduction, Crossover, and Mutation which change the parameter values and allow the GA to search in different areas of the search space. This is achieved in the following ways.

1. Reproduction involves a process in which the fittest chromosome of the present population is kept for the next

population. The remaining are replaced by new chromosomes which are formed through the crossover and mutation of the present population.

2. Crossover takes any two chromosomes from the present generation (these are called the parents), selects a number of the parents genes from one and swaps them with the same number and positioned genes in the other. This process forms two new chromosomes, called the children. This is repeated until there are enough children to replace the remainder of the present population that were not selected for reproduction.

3. Mutation is simply the random selection of a percentage of the new population's genes and the random change of their values.

After the chromosomes are altered to form the new population, they are incorporated into the optimization process and evaluated in the same way as the initial population. The whole GA process is then repeated a set number of times so that more points within the problem search space can be evaluated. This number of iteration is called the generation size and when it is reached, the GA has reached the optimum. Past experience has shown that this optimum is usually very close to the global optimum of the problem.

The problem of blind equalization by the basic GA is solved as follows.

Suppose M is the size of the alphabet of a_i . At time t there exists an upper bound M' on the number of sequences that may be considered as the input sequences. Let S_t is the collection of these sequences.

The initial population is created randomly (based on uniform distribution), which is composed of k ($\ll M'$) chromosomes (possible input sequences) $\{A_t^1, A_t^2, \dots, A_t^k\} \subset S_t$, each with t symbols.

Step 1. Evaluate the fitness for strings $A_t^i, i=1, \dots, k$, respectively.

The performance index (cost function) of A_t^i is:

$$J_t^i = (Y_t - \Phi_{t,n}^i \Phi_{t,n}^{i'} + Y_t)^T (Y_t - \Phi_{t,n}^i \Phi_{t,n}^{i'} + Y_t), \quad i=1,2,\dots,k. \quad (11)$$

Calculate and sort $J_t^i, i=1,\dots,k$, in ascending order. The order number of J_t^i is $r_t^i, 1 \leq r_t^i \leq k$. Then the fitness value of string A_t^i is

$$f_t^i = d(k-r_t^i)/k, \quad i=1,\dots,k, \quad (12)$$

$d > 0$ is a constant.

Step 2. Reproduction

Based on the survival-of-the-fittest mechanism, string A_t^i is reproduced (cloned) with probability

$$p_t^i = f_t^i / \sum_{i=1}^k f_t^i, \quad i=1,\dots,k. \quad (13)$$

We repeat the reproduction k times, every time only one string from $\{A_t^1, A_t^2, \dots, A_t^k\}$ is reproduced. For simplicity, the collection of reproduced k strings is still represented by $\{A_t^1, A_t^2, \dots, A_t^k\}$.

Step 3. Crossover.

Mate $\{A_t^1, A_t^2, \dots, A_t^k\}$ by selecting string pairs from it randomly (assume k even), to get $k/2$ string pairs. For every pair, crossing site c_t^i is chosen from $\{1,2,\dots,t-1\}$ randomly (based on uniform distribution), then the symbols from the two strings are swapped after the c_t^i th symbol with a crossover probability p_c . The resulting strings are still represented by $\{A_t^1, A_t^2, \dots, A_t^k\}$.

Step 4. Mutation.

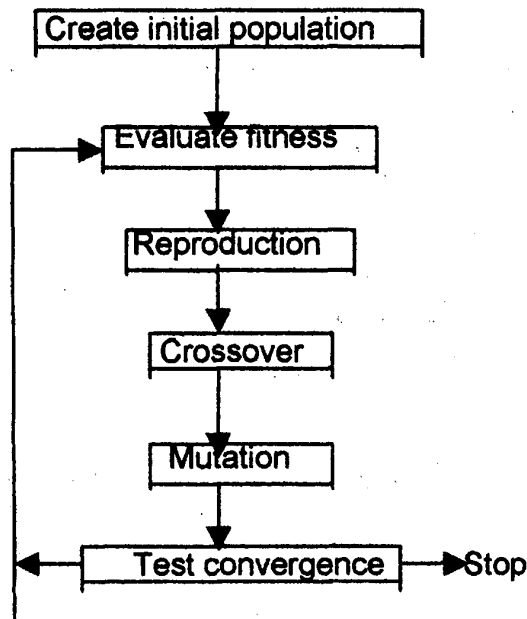
The mutation operator is a stochastic symbol-wise complementation applied with uniform probability $p_m \ll 1$. That is, for each single symbol in the population the value of the symbol is flipped from 0 to 1 or from 1 to 0 (assume $M=2$) with probability p_m .

Repeat the steps 1-4 several times until we get the population $\{A_t^1, A_t^2, \dots, A_t^k\}$ for which J_t^i , in formula (11) are such that

$$|J_t^i - J_t^{i+j}| < \epsilon, \quad i=1,\dots,k, \quad (14)$$

where j is the repeat index, $\epsilon > 0$ is the threshold value chosen a priori, and the desired estimate of A_t is the A_t^i , from the last population corresponding to the minimum J_t^i .

The flow chart of the genetic algorithm is as follows:



III. COMPUTER SIMULATION AND CONCLUSION

Computer simulations have been done for the proposed algorithm in order to prove its effectiveness. Besides the basic GA, several different schemes for the GA operation have been considered in order to get better results.

In practice, since it is expected that better parents generate better offspring, parent solutions that have higher fitness values have a higher probability to be selected for reproduction. To this end, several selection methods were used. Roulette wheel selection, which gives the selection probability to individuals in inverse proportion to their cost function values, has been used first. In this method, however, the scale of cost function values are not always suitable for the scale of selection probability. Suppose there are a few individuals whose cost function values are very low, and others whose are very high. The few parents are almost always selected and the variety of their offspring becomes small, which results in convergence to a local minimum. So, another selection method, i.e. rank-based selection, which uses an order scale of the cost functions instead of a distance scale as stated above in the Step 1, has been used next. We also used elitist strategy in the selection. The elitist strategy directly copies the best parents into the next generation in order to prevent the best individual in the offspring generation from becoming worse.

Crossover is the most essential operation in GA. In fact, there are several ways to combine parent chromosomes. Although the one-point crossover method described above was inspired by biological process, it has one major drawback

in which certain combinations of schema cannot be combined in some situations. To overcome this problem, a multi-point crossover method, which uses more than one cut point, and a uniform crossover method, which is based on a randomly generated crossover mask, are also assumed.

Mutation is a basic operation to spread the searching area of GA in order to get the global optimal solution. When parent chromosomes have similar bit patterns, the distance between the parents and offspring created by crossover is close in a genotype space. This means that the crossover cannot escape from the local minimum if individuals are concentrated near the local minimum. Mutation is just the operation to avoid this trapping at a local area by exchange bits of chromosome. On the other hand, if the mutation rate is too high, the GA searching becomes a random search, and it becomes difficult to quickly converge to the global optimum. So, in our simulations, an adaptive mutation method has been adopted. Before the crossover operation, the distance between the parents in the genotype space is measured by Hamming Distance. Then the mutation probability p_m of the offspring is determined based on the obtained value of the distance such that the smaller the distance is the larger the p_m is. In this way, we can keep the variety of the population (i.e. keep the global searching ability of the algorithm); meanwhile, we can prevent the "excellent individuals" from disruption.

In the simulations, the linear channel model (1) has been assumed. Different model order has also been assumed; for example, the model order n could be 4, 5, 6, ..., 10. The data length t could be 50, 100, 500 and so on. The population size k is assumed to be proportional to the data length t (e.g. $k=1.2*t$). The crossover probability was selected according to: $0.7 \leq p_c \leq 0.9$, and the mutation probability according to: $0.005 \leq p_m \leq 0.05$. For the adaptive mutation, we have $p_m = \min \{a/(H+c), 0.05\}$, where H is the Hamming distance of the parents, and a and c are positive constants (chosen properly).

In the simulations, the proposed algorithm determined the assumed channel model and the transmitted data for each situation with acceptable efforts. In general, the relative errors of the model coefficients were smaller than 0.01, and the data error rate was smaller than 0.01. The convergence of the algorithm varied for different model order and different data sizes. For example, if the order was $n=4, 5, 6$, and the data length $t=50 \sim 100$, the algorithm converged in about 30~60 generations; but if the order was $n=8, 9, 10$, and/or the data length $t=200 \sim 500$, the algorithm needed more than 150 generations to converge to the solutions correctly.

In the selection phase, rank-based selection together with the elitist strategy provided better convergence. In the mutation

phase of our cases, the mutation probability should not be larger than 0.05 in order to avoid the reduction of the algorithm's convergence. A properly designed adaptive mutation often improves and speeds up the algorithm's global searching ability. Through the simulations, we found that for our application, in general, two-point crossover provided an optimal number for multi-point crossover to improve the offspring production and increase the convergence rate.

In conclusion, computer simulations have shown that the GA is effective to complete the blind equalization problem. Based only on the received channel output signal sequence, it can recover the original transmitted signal and determine the channel model. Compared with other approaches for blind equalization, it needs less prerequisite information (i.e. the problem-dependence is smaller). Perhaps, the main advantage is its use where gradient algorithms falter due to multiple local optimization minima in the error function. Since the GA works over a knowledge base (i.e. the population of structures) that is distributed over the search space, it is more essential and easy to reach the global optimum. The GA then has superior global searching capability in the space with complex searching surface. Based on its intrinsic parallel search feature, in general, the algorithm provides the possibility of fast convergence to the global optimum.

IV. CURRENT AND FUTURE WORK

A. Improvement of the Computational Efficiency of the GAs

Through the simulations we know that the GAs are effective in the area of blind equalization. GAs are usually much more robust than gradient based algorithms and can tolerate even an approximate or noisy design objective evaluation. Through the simulations, we also have found that the computational efficiency of GAs is sometimes not high enough, especially for the high-dimension applications where the computations become very time-consuming. This drawback might reduce the possibility of using the GA approach for real-time applications. Therefore, any effort in the direction of computational cost reduction for the optimization process should be seen as an opportunity to face more challenging design problems. In planned follow on work, we will follow two major paths: (1) improve the efficiency of the GAs optimizer by introducing the gradient based hill-climbing operator; (2) reduce the time needed for the single fitness calculation by introducing a neural network mapping approach. Each of these are discussed in the following sections.

(1) GA coupled with gradient based algorithm and its application in blind equalization in every generation, the population is created based on the stochastic mechanism in GA. The advantage of this mechanism lies in keeping a variety

of individuals in the population, and so keeping the global searching ability of the algorithm. The algorithms often converge very slowly, especially for the high-dimensionality case. An effective way to make evolutionary procedures competitive with the specialized approaches to optimization, while maintaining their favorable features, is through hybridization (integration). This consists of selecting the proper combination of the evolutionary procedure and an optimization technique of different nature, possibly suited for the domain of the chosen application or characterized by a complementary behavior. In many cases, hybridization may lead to better solutions than those obtainable using the two methods individually.

One scheme of hybridization is coupling the genetic algorithm to the gradient based optimization technique. The coupling has been obtained by introducing an operator that performs a gradient based optimization, working at the same level as the other classical genetic operators. Gradient based methods have found wide application in engineering because of their capacity to effectively refine and improve existing solutions (especially in the initial stage of the adaptation process). But when elements already coincide or fall very close to local minima, the new individuals introduced by the method provide very little information to the procedure. In general, for the multi-modal case, the gradient-based optimization does not guarantee that the best results are obtained (global optima). The hybridization would favorably exploit the best features of both methods and provide a robust design procedure.

In order to realize the hybridization, the optimization process begins with a GA and subsequently refines the best solution found by switching to the gradient based method. A hill-climbing operator (HCO) is introduced. Through the selection of crossover and mutation operators, an intermediate generation is created. Some of its elements can then be selected and fed into the HCO to obtain the new generation. The choice of the elements can be done according to three alternative schemes: (1) the HCO is applied only to the best fit individual; (2) a number of individuals determined by an assigned probability is chosen using the selection operator adopted; (3) a number of individuals determined by an assigned probability is chosen in a purely random fashion. The HCO is not allowed to carry out a converged optimization, but operates for a fixed number of iterations (usually 2 at maximum). In practice, the HCO acts with an assigned probability and is applied every other generation.

Some initial simulations have shown that as the GA and the auxiliary HCO characteristics tend to be complementary--the drawbacks of both techniques can be attenuated. The effectiveness of GA in searching for global optima in large parameter spaces is coupled to the high efficiency of the HCO in improving the initial solution. So, the typical drawback

of GA not being able to efficiently refine a solution is avoided by the hybrid scheme. On the other hand, the use of gradient based technique within the framework of GS prevents pitfalls of convergence to local minima and of dealing with convex design spaces only.

(2) Neural network based calculation in the GA and its application in blind equalization

Genetic algorithms are known to be robust optimization techniques capable of effectively exploring very large parameter space, and performing well over a wide range of optimization problems. However, they generally require a high number of time-consuming evaluations for the objective function, thus offering poor performance in terms of computational efficiency, especially when compared to optimization techniques tailored to a particular application domain. For this reason, the solution of complex optimization problems with an evolutionary procedure requires careful consideration about the computational efficiency of the algorithm. If the evaluation of the objective function is computationally fast, genetic algorithms are very good candidates for a successful solution to the problem, particularly if the search space is very large or/and non-convex. The peculiar nature of evolutionary computing often allows the solution of optimization tasks that other techniques are not able to cope with. Insertion of a neural network based calculation in the evolutionary process in order to improve the computational efficiency of the GA, however, is possible and may make feasible real-time application for blind equalization.

In the basic GA based blind equalization, in every period of the iteration, the objective (loss) function (10) should be evaluated for every individual of the population. If we introduce the gradient based operator, the derivative of the objective function should also be evaluated for some individuals of the population in the iteration. Since both the objective function and its derivative function are highly nonlinear, highly-dimensional, and include matrix inversions, we face complex computational issues which are main time-consuming factors in the algorithm presented. The essential and key problems are how to speed up the real-time computations of the two nonlinear functions. To this end, the real-time neural networks (NNs) are assumed to replace the direct real-time evaluation of the two nonlinear functions in the applications.

NNs have displayed many desirable properties required for the complex nonlinear problems. Some of these include: 1) learning by experience (training); 2) ability to generalize (i.e. to map similar inputs to similar outputs); 3) the ability to map nonlinear functions to parallel distributed process for faster processing of large scale data; 4) to provide a degree of robustness through fault tolerance; 5) to be robust in the presence of noise; and 6) to allow multivariable

capabilities. All of these properties are very significant for our present application.

For our case, the multi-layer perceptron is taken into account as the NN structure. We design two kinds of NNs, NN1(t, n) and NN2(t, n), used for the evaluations of the objective functions (10) and the derivative of the objective functions, respectively. Here t represents the length of the data and n represents the system order. For every t and every n , in which we are interested, the NN1 and NN2 have $(2t-n+1)$ input nodes and one output node, respectively. The sigmoid function is assumed as the activation function for every neuron in the NNs. We should collect proper and sufficient input-output data of function (10) and its derivative in advance, which are used to train the NN1 and NN2, respectively. Though the popular back propagation algorithm could be used to train the NNs, the basic genetic algorithm could also be used to optimize the NN structure (topology) including the number of hidden layers, the number of neurons in each layer, and the parameters of the NNs. All of the numbers and parameters should be the optimization parameters of the GA. In our case, the computational efficiency for the training phase of the NNs is not important for the application because the training is performed off-line. Once the training is completed, the NN1 and NN2 will be used on-line in order to provide the desired fast mapping for the improved GA. In this way the main obstacles against the improvement of the computational efficiency would be overcome, and the presented improved GA would be feasible in a real-time application.

B. Genetic Programming Based Modeling for Nonlinear Channel

Some communication channels are nonlinear, such as the complex underwater channel over broad bandwidth. In these cases, using a linear equation (1) as the model of the channel will produce unsatisfactory results. Thus, building nonlinear channel models can be of essential importance. The identification of linear systems is a mature field and there are a large number of powerful algorithms for structure and parameter identification. Unfortunately, most of those algorithms cannot be extended for the structure identification of nonlinear models. This shortcoming is caused by the large amount of possible relations between the input and output variables. Thus, more powerful search and optimization algorithms are required to solve such complex tasks. Gas have proven to be very powerful search and optimization tools. Since the population consists of fixed length character strings, GAs have their limitation when it comes to identifying the structure of a system. Genetic programming (GP) offers a solution to this problem. Its paradigm solves the representation problem by significantly increasing the complexity of the structure undergoing adaptation. In particular, such a structure consists of tree like arranged nodes that may contain parameters, variables, mathematical

operators and other functions. Our future work will concentrate on how to use GM in the modeling of nonlinear channels.

V. CONCLUSIONS

We have described a general approach for the use of GA blind equalization methods in underwater communications channel estimation. Our initial work has resulted in the development of a simulation model that has provided performance commensurate with receivers using the decision feedback equalizer. The GA, although slower to adapt, has a number of features that may make it a preferable approach for some ACOMMS applications. This paper is Harbor Branch contribution number 1311.

ACKNOWLEDGMENTS

The authors wish to thank the Office of Naval Research under grant N00014-96-1-5022 and the Atlantic Foundation for partial support of this work. This is Harbor Branch contribution number 1247.

REFERENCES

- [1] Randy L. Haupt and Sue E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, Inc., 1998.
- [2] Phil Mars, et al, Learning Algorithms: Theory and Applications in Signal Processing, Control and Communications, CRC Press, Inc., 1996.
- [3] Thomas Back, Evolutionary Algorithm in Theory and Practice, Oxford University Press, 1996.
- [4] Fredrik Gustafsson and Bo Wahlberg, Blind Equalization by Direct Examination of the input Sequences, IEEE Trans. Commun., vol. 43, pp. 2213-222, 1995.