Design of 1-D Piece-wise Maps for Multi-user CDMA Communications Using a Novel GP-DNA Algorithm

SURENDRAN K. SHANMUGAM, VINAY VARADAN and HENRY LEUNG
Department of Electrical and Computer Engineering
University of Calgary
2500 University Drive N.W. Calgary
CANADA T2N 1N4

Abstract: - This paper addresses the design of one dimensional piece-wise maps to generate optimal codes for short and long code multi-user CDMA systems. A novel approach, GP-DNA, that uses genetic programming to search in the functional space and DNA computation to obtain the optimal initial conditions has been proposed as the design methodology. This method automatically designs orthogonal codes for short-code CDMA and codes with good aperiodic correlation properties for long code CDMA. Monte-Carlo simulations illustrate the improved performance of GP-DNA compared to conventional code designs.

Key-Words: - Genetic programming, DNA computation, chaos, CDMA, orthogonal codes

1 Introduction

The goal for third generation wireless systems is to offer high bit rate multimedia services like high-speed data, video and multimedia traffic. Code Division Multiple Access (CDMA) systems equipped with reliable multi-user interference (MUI) cancellations, have gained worldwide acceptance over competing TDMA and FDMA alternatives whose capacity is limited [1]. The multipath propagation characteristics of the channel and MUI degrade the system performance thereby necessitating improved design of spreading codes [2]. Hence the design of spreading sequences for CDMA communications is still an active field of research [3].

Over the past few years the application of chaotic sequences generated by nonlinear dynamical systems to Direct Sequence Spread Spectrum (DS/SS) systems has gained prominence. Among the various nonlinear dynamical systems that can be used to generate spreading sequences, polynomial maps such as the logistic map and chaotic markov maps are preferred due to their good pseudo-random properties [4,5]. The traditional analytical techniques for chaotic maps have used the Ergodic properties of chaotic maps to estimate their correlation performance [6]. Since exploitable sequences generated from sample trajectories are of much shorter length than what is required for the correlation performance of the map to approach the theoretical limit, there is a need to search for maps to generate short sequences with good performance.

The DNA computing paradigm introduced by Leonard M. Adleman [7] inspired an efficient implementation using chaotic systems [8]. In order to model the combinatorial processing engine and the associated DNA operations, a nonlinear chaotic dynamical system was proposed as the core engine of the search process.

In this paper we propose an algorithm based on DNA computation to quickly identify the initial conditions that generate optimal codes from any given nonlinear map. Further, using Genetic Programming, piece-wise nonlinear maps are designed that maximize the number of obtainable optimal codes from a single map. The combination of Genetic Programming (GP) and DNA computation leads to an efficient algorithm (GP-DNA) that can design optimal codes by specifying the map and the optimal initial conditions for the map. Initial results show that the design procedure is able to generate short orthogonal codes as well as long aperiodic codes with good correlation properties.

The paper is organized as follows. In Section 2 we introduce a general CDMA model signifying the importance of code design. Section 3 introduces the idea of code generation using nonlinear maps and the GP-DNA approach to optimal map design. Simulation results are presented in Section 4 with concluding remarks in Section 5.

2 CDMA System Model

The schematic diagram of a CDMA system is depicted in Fig 1 [9].

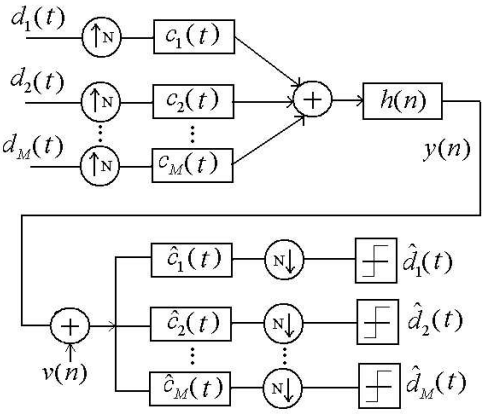

Fig 1: CDMA system model

The continuous time waveform at the transmitter can be expressed as,

$$x(t)=\sum_{j=1}^{M}\sum_{i=-\infty}^{\infty} A_j d_j(i) s_j(t-iT_B) \quad (1)$$

Where

$$s_j(t)=\sum_{k=0}^{N-1} s_{j,k} P_{T_C}(t-kT_C), \qquad P_{T_C}=\begin{cases} 1/\sqrt{T_C} & t\in[0,T_b] \\ 0 & otherwise \end{cases},$$

T_b = N T_c where N is the processing gain and T_b is the bit period and T_c is the chip period, P_{Tc} is the pulse shaping filter. And the up samplers and down samplers serve the purpose of spreading and dispreading by the factor of N (processing gain).

The complex baseband model of the multipath channel can be written as:

$$h(n)=\sum_{l=0}^{L-1}\alpha_l e^{j\beta_l}\delta(n-lT_C) \quad (2)$$

Where L is the number of resolvable paths in the channel and α_l is the Rayleigh fading amplitude of the l^{th} path and β_l is the channel phase which is uniformly distributed between [0, 2π).

The received signal is then given by:

$$y(n)=\sum_{j=1}^{M}\sum_{i=-\infty}^{\infty} d_m(i)\sum_{l=0}^{L-1} h(n)c_j(n-l-iT_b)+v(n) \quad (3)$$

At the receiver, the received signal is correlated with the locally generated sequence $\hat{c}(t)$ and the resulting output is despreaded (down sampling) and the decision is made on the estimate.

$$\hat{d}_p(i)=\sum_{k=0}^{N-1}\hat{c}_p^T(k)c_p(k)h_p d_p(i) + \sum_{j=1,j\neq p}^{M}\sum_{k=0}^{N-1}\hat{c}_j^T(k)c_j(k)h_j d_j(i)+\sum_{k=0}^{N-1}\hat{c}_j^T(n)v(n+k) \quad (4)$$

Where, $h_j=[h_j(0),h_j(1),.....,h_j(L-1)]^T$. The first term is the signal part and the second term is the MUI part and the last is AWGN part with variance σ^2.

Let the MUI in the equation (4) be modeled as gaussian noise. Then different receivers can be used at the receiver. One can simply use a matched filter or zero forcing detector or MMSE receiver, by choosing [10]:

$$\hat{c}_j(k)\equiv h_j^{\chi}C_j^{\chi}(k) \quad .....MF$$
$$\hat{c}_j(k)\equiv h_j^{\chi}R^{\dagger}C_j^{\chi}(k) \quad .....ZF \quad (5)$$
$$\hat{c}_j(k)\equiv h_j^{\chi}(R+(\sigma A_j^{-1})^2)^{\dagger}C_j^{\chi}(k) \quad .....MMSE$$

Where R is the correlation matrix whose $(j,p)^{th}$ element is $\rho_{jp}=\int_{-\infty}^{\infty} c_j^*(t)c_p(t)=c_j^{\chi}c_p$, χ stands for Hermitian and $\dagger$ stands for pseudo-inverse.

The above equations are equivalent to the RAKE receiver if all the delayed waves at the input of the receiver are combined coherently. However, the MF does not take advantage of the known interfering users sequences. Hence the design of $\hat{c}(k)$ becomes important in order to minimize MUI.

In the case of long codes, the design would involve the minimization of the auto-correlation and cross-correlation factors:

$$P_A=\max_{\tau\neq 0}\frac{1}{P_{jj}(0)}|P_{jj}(\tau)|$$
$$P_C=\max_{i\neq j}\max_{\tau}\frac{1}{P_{jj}(0)}|P_{ij}(\tau)| \quad (6)$$

where, $P_{ij}(\tau)=\sum_{k=0}^{N-1}c_i(k+\tau)c_j(k)$ is the aperiodic correlation function.

Short codes, however, should satisfy the orthogonality condition:

$$c_j^{\chi}(k)c_p(k)=\delta(j-p) \quad (7)$$

3 Map design using GP-DNA

The design of maps for generation of orthogonal codes is not a trivial problem. Both the choice of the map as well as the initial conditions used to generate the codes needs to be optimized in order to achieve good performance. There exist no easy analytical methods to solve the dual optimization problem of searching in for maps in the functional space while searching for good initial conditions in the space of real numbers.

3.1 Obtaining Codes from Maps

Different codes are obtained from individual maps by choosing different initial conditions and mapping the obtained time series to Bernoulli symbols. Thus, for a processing gain of N, and the number of users equal to M, the codes can be obtained from a given map as follows:

$$x_j(k) = f^k(x_j(0)), \quad k = 0, 2, ..., N-1,\ j = 1, 2, ..., M \tag{8}$$

The sequences generated by the map are then converted to spreading codes by:

$$c_j = g\{x_j(n) - E(x_j(n))\} \tag{9}$$

where $g(x) = \begin{cases} 1 & x > 0 \\ -1 & x \le 0 \end{cases}$. The initial conditions $x_j(0)$ need to be chosen efficiently in order to obtain optimized codes.

3.2 DNA Computation for Initial Condition Search

In order to search for the optimal initial conditions, we propose a novel scheme inspired by DNA computation.

DNA computation involves the manipulation of sequences of nucleotides. The DNA consists of four nucleotides A, T, G and C that are used to form strings to encode information [7]. A multiset of such strings can be used to represent candidate solutions of an optimization problem. The actual or optimal solution (or one of the solutions) has to be selected among the many candidates. Adleman's DNA computation paradigm uses a set of operations that form the *restricted DNA model* to process a multiset of DNA strings stored in a tube.

Instead of using actual DNA in tubes, it is also possible to formulate the DNA computation paradigm using symbolic dynamics of nonlinear dynamical systems [8]. The solutions can be represented as strings of binary numbers constrained to a certain length. In the case of nonlinear maps, the symbolic dynamics of any given initial condition can be obtained by using a simple threshold. In other words, given a nonlinear map *f* the sequences can be obtained as explained in (8) and (9). Thus, for a given map *f* we can obtain several different symbolic sequences associated with different initial conditions.

Since chaotic dynamical systems are sensitive to initial conditions, small changes in the initial condition can lead to different symbolic sequences that can each be associated with a possible solution to the given optimization problem.

Thus, it is possible to search amongst different candidate solutions to the problem by just perturbing the initial conditions of the nonlinear map. Thus, given a set of multiple criteria that need to be satisfied for a symbolic sequence to be acceptable, each of the criteria can be associated with some feedback, all of which need to be satisfied for no perturbation.

Since in our case each of the codes need to be optimized with respect to all the other sequences, the individual feedbacks output a 1 whenever the new code is not orthogonal to the previous codes and a 0 when they are orthogonal. An OR operation is performed on the output from each of the feedback blocks. The resulting 1 or 0 is then multiplied by a perturbation value δ generated from another chaotic system, to obtain the actual perturbation ξ that gets added to the initial condition of the nonlinear map. The resulting DNA computation based algorithm is depicted in Figure 2.

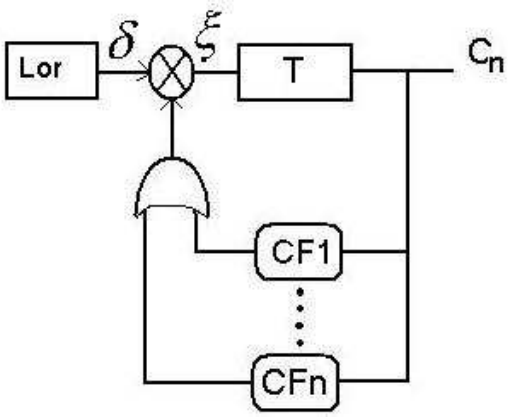

Fig 2: Schematic of DNA Computation

The LOR system is a chaotic system that generates the perturbations used to steer the chaotic map T towards optimal solutions. Thus the system T generates optimized codes.

3.3 Genetic Programming for Map Design

Now that we can search for optimized codes generated by any given map, we need to optimize the map in order to maximize the number of codes obtainable from it. This is achieved by using GP.

Genetic programming (GP) as a search and optimization technique is well known and has been applied to many problems [11,12]. Genetic programming is particularly useful in the specific application of inverse problems such as finding optimal mathematical expressions that fit certain criteria [11]. We have therefore chosen GP to search in the functional domain for optimal maps.

In GP individual maps are represented as trees, where the leaf nodes are input variables from the terminal set *T*, and internal nodes are operators from the functional set *F* [4]. We set $F=\{+, *, -, \mathrm{mod}\}$ and $T=\{x(t-1),\ \textit{random ephemeral}\ \text{constant}\}$.

Since objective of the GP design is to search for maps that can provide maximum number of codes of the given length that satisfy the given objective function, the fitness of the individuals in any population of GP is assigned depending on the number of optimal codes that can be obtained from them using the DNA computation algorithm. We can now summarize the GP-DNA map search algorithm as follows:

i. An initial population of *N* random functions (trees) is initialized with the "ramped half-and-half" method, which we describe now. An equal amount of trees are assigned to depths $d=\{2, 3, \ldots, D_{max}\}$. At each depth, half the trees are generated with the "full" method, and half with the "grow" method. In the "full" method, all the leaf nodes have the same depth d_i; internal nodes are randomly selected from *F*, and leaf nodes are randomly selected from *T*. The probability density among the choices for *F* and for *T* is uniform. The "grow" method starts by randomly choosing a function in *F* and an input in *T* for the root node (50% probability for each). Then a corresponding symbol is chosen from *F* or *T*. For each *F* that is chosen, the "grow" method is recursively applied to that node. Any nodes that attain depth d_i must choose from *T*. For our search, we chose *F* = {+,*,-,mod1} (where mod1(a) = a (mod 1)), and *T* ={x(t-1), random ephemeral constant in the range [-10,10]}.

ii. Rank each individual proportional to the number of orthogonal codes obtained using the DNA algorithm. Sort the population according to rank and assign fitness to the individuals by linearly interpolating from the best to the worst rank. Average the fitnesses of individuals with the same rank so that all of them are sampled at the same rate.

iii. Create a new population by reproducing or combining selected functions in the current population. Reproduction of a function is simply the copying of a function to the new population. The genetic operator of Crossover is used to combine functions. Crossover involves crossing over two random subtrees of two respective parents. The probability of an individual being selected for reproduction or crossover is proportional to that individual's fitness rank in the population. We set the probability of performing reproduction vs. crossover to 0.2 vs. 0.8. Also, we forced the population to be 100% diverse (i.e. no duplicates).

iv. Repeat steps ii and iii until convergence to an optimal solution or the maximum number of generations is exceeded. We set the maximum number of generations to be 51.

The design performance of GP-DNA is evaluated in the following section.

4 Performance Analysis

4.1 Design for Short Code CDMA

In this section we present performance analysis results for low processing gain DS-CDMA. GP-DNA was used to design maps that generate orthogonal codes with N = 16 and M = 16. The evolution of the best fitness across generations of the GP-DNA is shown in Fig 3.

The map designed by GP-DNA is as follows:
x(n) = (mod1(((mod1(x(n-1)))+((mod1((mod1((x(n-1))+(mod1(x(n-1)))))+(mod1(((8.3863).*(x(n-1))).*((2).*(x(n-1)))))))).*(8.5751))).*(mod1(-8.2996))))

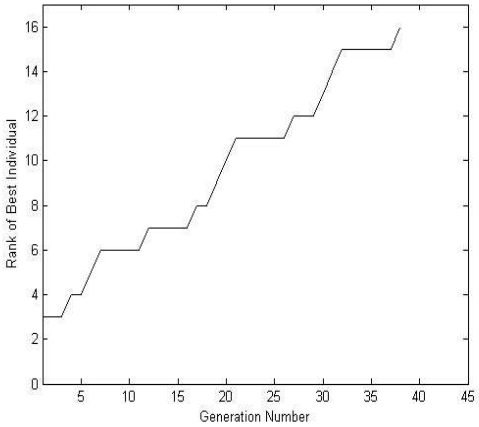

Fig 3: GP-DNA evolution

The state-space portrait of the above map is shown in Fig 4.

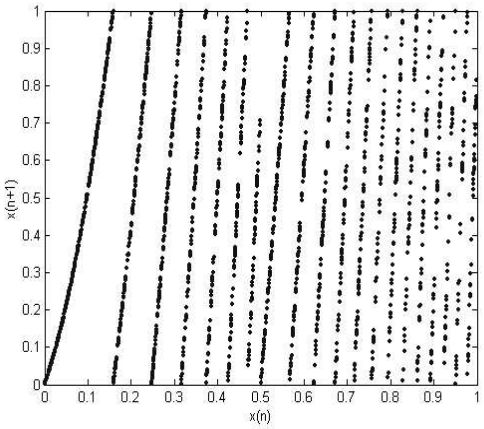

Fig 4: State space portrait of GP-DNA designed map for NxM = 16x16

The orthogonal property of the designed codes can easily be seen from the frequency domain plot in Fig 5.

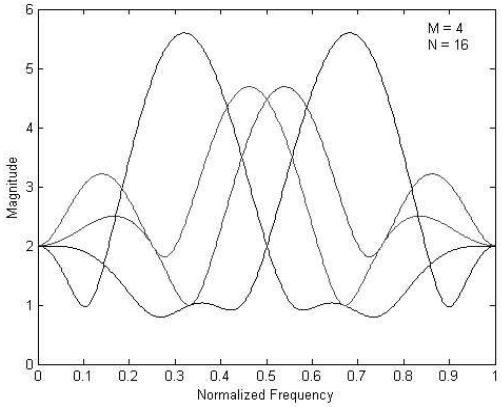

Fig 5: Frequency domain plot of GP-DNA codes

The performance of the designed codes were evaluated for different channels. We assume the receiver has perfect bit and chip synchronization and perfect knowledge of the channel status. We compared the performance of the GP-DNA codes with Walsh codes, Gold code and Markov map generated codes for N = 32 and M = 4. The result of the Monte-Carlo simulation for L = 5 multipath Rayleigh fading channel is shown in Fig 6.

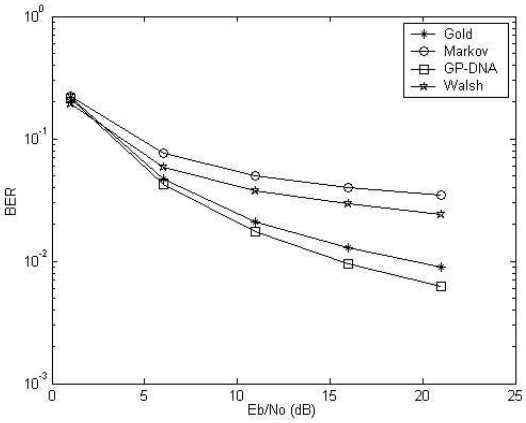

Fig 6: BER vs. Eb/No for multipath Rayleigh fading channel

We have also evaluated the performance of the designed codes in AWGN channel and MUI with M = 8 and N = 32. The performance was evaluated with both Zero-Forcing detector and MMSE receiver. The results of the simulations are shown in Fig 7. The performance of GP-DNA code is about the same as gold code in MMSE and better in decorrelator.

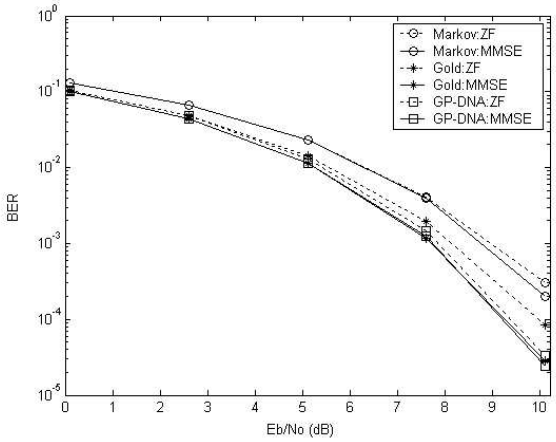

Fig 7: BER vs. Eb/No for AWGN noise MUI channel

A significant aspect of the GP-DNA design is that the algorithm generates different solutions with

orthogonal properties, which is not possible in the case of either the Walsh or gold sequences.

4.2 Long code design for CDMA

We also applied our code design algorithm for long code CDMA systems. The performance of the codes were evaluated for (N = 32, M = 4 and length of code equal to 2^{15}-1) L=5 multipath Rayleigh fading channel. The results of the simulations are shown in Fig 8.

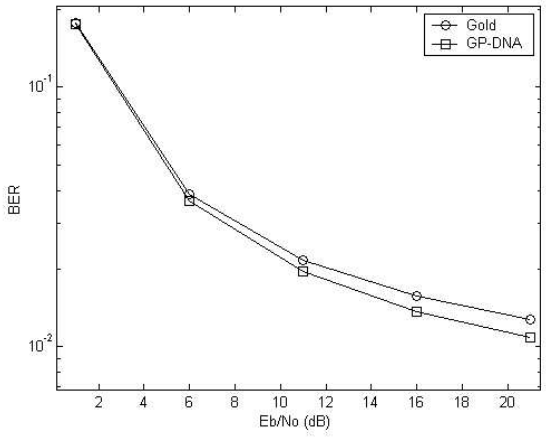

Fig 8: Performance of Long code CDMA system in multipath Rayleigh fading channel

As can be seen from Fig 8 the performance of the GP-DNA codes is slightly better than gold sequences as it has optimized auto- and cross-correlation properties.

5 Conclusion

In this paper we have proposed a novel technique combining genetic programming and DNA computation to design maps that produce maximum number of optimized codes for both short and long code CDMA. Monte-Carlo simulations results confirmed that the performance of the GP-DNA codes were better than the standard Walsh and gold sequences for different channel conditions. Further the ability of GP-DNA to design different maps that generate codes with similar performance allows flexibility in the design.

References:

[1] Savo G. Glisic and Pentti A. Leppanen, *Code Division Multiple Access Communications*, Kluwer Academic Publishers, 1995.

[2] Hui Liu, *Signal Processing in CDMA Applications*, Artech House Publishers, 2000.

[3] T. Sang, R. Wang and Y. Yan, "Constructing chaotic discrete sequences for digital communications based on correlation analysis," *IEEE Trans. on Signal Processing*, vol. 48, no. 9, 2000, pp. 2557-2565.

[4] K.S. Surendran, A. Pachai and K. Murali, "On the performance of chaotic spreader based multicode multicarrier system for broadband wireless communication system," *submitted to IEE Proceedings on Communications.*

[5] J. Tou, P. Yip and H. Leung, "Spread-spectrum signals and the chaotic logistic map," *Circuits Systems and Signal Processing*, vol. 18, no. 1, 1999, pp. 59-73.

[6] S.H. Isabelle and G.W. Wornell, "Statistical analysis and spectral estimation techniques for one-dimensional chaotic signals," *IEEE Trans. on Signal Processing*, vol. 45, no. 6, 1997, pp. 1495-1506.

[7] L.A. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, 1994, pp. 1021-1024.

[8] G Manganaro and J Pineda de Gyvez, "DNA Computing based on Chaos," *Proc. Of ICEC*, vol. 1, 1997, pp. 255-260.

[9] M.K. Tsatsanis, "Inverse filtering criteria for CDMA systems," *IEEE Trans. On Signal Processing*, vol. 45, 1997, pp. 102-112.

[10] R.Lupas and S.Verdu, "Linear Multiuser Detectors for Synchronous Code Division Multiple Access Channels," *IEEE Trans. Information Theory*, vol. 35, 1989, pp.123-136.

[11] V. Varadan and H. Leung, "Design of piece-wise maps for spread spectrum communication using genetic programming," *Proc. of ISCAS*, vol. 3, 2001, pp. 225-228.

[12] V. Varadan and H. Leung, "Reconstruction of polynomial systems from noisy time series measurements using genetic programming," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 4, 2001, pp. 742-748.