Proceedings
ECCTD'97
Budapest, September 1997

CDMA COMMUNICATIONS USING MAPS WITH STORED INFORMATION

Yu. V. Andreyev, A. S. Dmitriev, D. A. Kuminov, and S. O. Starkov

Institute of Radioengineering and Electronics, Russian Academy of Sciences, Moscow. Russia
e-mail: dmitr@chaos.msk.su

Abstract — **In this report, we consider the maps with stored information as sources of spreading pseudo-random sequences for communications systems. We show that application of such sources allows us to (i) have very rich set of code sequences; (ii) realize self-synchronization of the transmitter and receiver; (iii) guarantee security of the communications.**

I. INTRODUCTION

Application of generators of chaos as sources of noise-like signals for spread-spectrum communications systems, in particular, to Code Division Multiple Access (CDMA) communications systems, seems promising due to simplicity of their implementation and opportunity of having self-synchronized receivers based on the principle of the chaotic synchronous response [1]. According to this principle, we can form a "master–slave" pair, where the master is a chaotic system, and the slave is a system playing the role of a nonlinear matched filter. The effect of the first system on the second one is the chaotic synchronous response. Thus, the master and the slave systems can be synchronized without external synchronization signal. Recognition of this fact has caused the great interest to the application of the properties of chaos to spread-spectrum communications systems [2]–[10].

However, in order to use chaos and complex dynamics in practice, a number of problems have to be solved. In particular, the coherent receivers, proposed by now are sensitive to external noise and interference.

In this report, we propose to employ special chaotic systems — 1-D and N-D maps with stored information [11]–[17] — as sources of pseudo-random code sequences for spread-spectrum communications systems.

From the viewpoint of nonlinear dynamics, the systems with stored information are a special case of chaotic systems. They demonstrate transient chaos when starting from arbitrary initial conditions in the case of storing information as stable cycles. In the case of storing information as unstable cycles, stationary dynamic chaos is realized in such systems. The dynamic systems with stored information have a number of properties inherent of chaotic systems of general kind. In particular, they can be synchronized and can play the role of nonlinear matched filters for signals of a special kind (for example, for the code sequences stored in these maps). On the other hand, their periodic trajectories go through a finite number of points, and in this sense, they are close to systems with pseudo-random sequences.

It seems to us that the set of above properties can be useful in CDMA communications systems.

II. MAP WITH STORED INFORMATION AS A SOURCE OF SPREADING CODE

The procedure for storing and retrieval of information as limit cycles of one-dimensional (and multidimensional) dynamic systems was proposed in [11] and further developed in [12]–[17].

Let a sequence of symbols be given (information block)

$$\{a_i\} \equiv a_1 a_2 \ldots a_n, \qquad (1)$$

whose elements a_i belong to an alphabet with N symbols.

For this sequence, we design a special piecewise-linear 1-D map of a unit interval into itself. This map possesses a stable limit cycle of the period n whose points are unambiguously related to the elements of sequence (1). In the simplest case, each element of the alphabet is related to its own value and its own subinterval (of the length $1/N$) of the mapping variable. The linear map segments containing the points of the given cycle are called information regions of the map.

To retrieve the information block, we set the initial condition within one of the subintervals corresponding to the symbols of the information block, iterate the map, and transform the sequence of numbers occurring by iteration into the sequence of symbols.

If the initial condition is arbitrary, then the occurrence of the phase trajectory on the stable limit cycle can be preceded by a chaotic transient process.

Similarly, we can design a map with a few stable limit cycles whose points are unambiguously related to the elements of the stored sequences $\{a_i^j\}$, where j is the sequence index.

However, the capabilities of this method prove to be limited. For example, we cannot store sequences containing equal symbols. In order to overcome this limitation, we introduce the notion of the level of storage q and design the maps whose points correspond not to one symbol of sequence (1), but to q consecutive symbols. This generalization of the map design procedure allows us to store any symbol sequences having no equal substrings with the length equal to q. If there are such equal substrings, the storage at this level is impossible.

The information is retrieved now by means of setting the initial condition within one of the subintervals of the unit interval $I = [0, 1]$ corresponding to a substring of q symbols of an information block, iteration of the map, and subsequent transformation of the sequence of numbers into the sequence of symbols.

A natural way of overcoming the limitation of the storing procedure is a use of higher storage levels. However, here we meet principle restrictions. Beginning from a certain storage level, the length of the information regions becomes very small, and by computer simulation of the maps, we have to use double-precision arithmetic instead of single-precision calculations with all consequences (slower calculations, tougher memory requirements, etc.).

However, the problem of storing any number of arbitrary sequences is radically solved by means of the special encoding [13], [14].

As follows from the properties of the map with stored information, the map can be used as a source of encoding sequence for communications systems. We have to choose the code base n and a pseudorandom sequence (sequences) with this code base, and then to design the corresponding map. The storing of an information sequence on the map is preceded by the special encoding of the sequence, hence, the communications system must have two additional elements, the encoder and decoder.

The encoder of the map (*EM*) transcodes the information sequence from the initial alphabet into the alphabet used in this map. The decoder of the map (*DM*) performs the inverse operation: it transforms the sequences of the mapping variable values into the symbols of the initial alphabet.

Note again, that in contrast to the systems generating binary PN-sequences, any symbol sequences can be stored on the map, and the initial alphabet can have any finite number of elements. The maps have practically no restrictions on the kind of the sequences and on the volume. Any information that can be represented as a string of symbols belonging to a given alphabet can be stored on the map, and then used as an encoding spreading sequence (the carrier). It can be pseudorandom sequences with arbitrary number of possible states L, literature texts (then the length of the initial alphabet is equal to the number of the elements of the given language alphabet), pictures (bitmaps) composed of pixels with different colors (then the length of the initial alphabet is equal to the number of different colors in the picture), etc.

The possibility of storing several information strings in a single map allows organization of individual channels of private communications by means of storing the same string only for those users who want to have access to a given communications channel. Then, the information that is transmitted through this private channel cannot be retrieved by other users.

III. AN EXAMPLE OF A COMMUNICATIONS SYSTEM

We shall consider a communications system composed of a source of information, a transmitter, a communications channel, a receiver, and a user of the information. The information in the source is represented by a binary sequence $\{b_k\}$.

The transmitter involves a map $\mathbf{M_1}$ with spreading code sequences stored in it. One or several code sequences can be stored on the map. In the case of storing

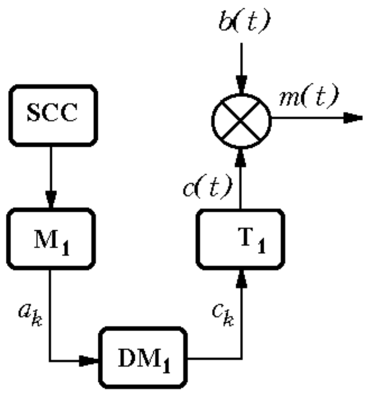

Figure 1: The transmitter.

several sequences, the user receives an opportunity of the spreading code choice (SCC) for the communication with another user. In particular, each pair of users can have their own spreading code. N sequences $\{a_i^j\}$, $j = 1, ..., N$, are stored in the map in a generalized alphabet. To translate them to the binary form, a decoder of a sequence $\{a_i\}$ from the generalized alphabet of the map into binary sequence $\{c_k\}$ is introduced in the transmitter.

Note, that the indices i and k of sequences $\{a_i\}$ and $\{c_k\}$ do not coincide, in general, *k*³*i*. Let the signals $b(t)$ and $c(t)$ represent the nonzero sequences $\{b_k\}$ and $\{c_k\}$ as signals assuming the values ± 1.

The product of an information-carrying signal $b(t)$ and signal $c(t)$ gives a signal, whose spectrum is equal to the convolution of the spectra of the two signals. This and a number of subsequent operations with a signal are completely similar to those used in the classical systems of the binary information transmission with noise-like signals (PN-sequences). Therefore, the properties of the signal at all stages of transformation are well-known. Namely, if the message signal $b(t)$ has a narrow spectrum, and the noise-like signal $c(t)$ the broad spectrum, then the signal product, the modulated signal $m(t)$, will have the spectrum close to the spectrum of the broadband signal $c(t)$.

The signal at the receiver input $r(t)$, passed through the communications channel, consists of the transmitted signal $m(t)$ and a signal $h(t)$ of additive noise. Hence,

$$R(t) = m(t) + \eta(t) = c(t) \cdot b(t) + \eta(t). \quad (2)$$

In order to extract the initial signal of the message $b(t)$, the received signal $r(t)$ is fed to the demodulator, which consists of a multiplier, an integrator, and a decision-making unit (Fig. 2). The encoding sequence is generated in the map M_2, decoded into the binary form in DM_2, represented as a two-level binary signal, and fed to the multiplier along with the signal $r(t)$. The signal $c'(t)$ is an exact copy of signal $c(t)$ used in the transmitter.

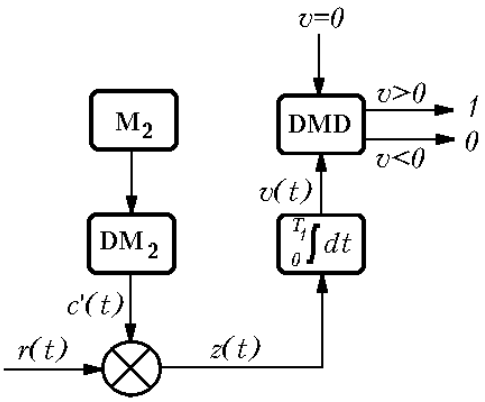

Figure 2: The receiver.

The receiver is supposed to work at ideal synchronization with the transmitter. In this case, the signal at the multiplier output in the receiver is given by expression

$$Z(t) = c'(t) \cdot r(t) = c(t) \cdot c'(t) \cdot b(t) + c'(t) \cdot \eta(t). \quad (3)$$

Equation (3) shows that the information signal $b(t)$ is multiplied twice by $c(t)$, while the undesirable signal $h(t)$ is multiplied by it only once.

The signal $c(t)$ assumes the values ± 1, and the difference vanishes, when is powered to square, hence,

$$Z(t) = b(t) + c'(t) \cdot \eta(t). \quad (4)$$

As is seen from equation (4), the information signal $b(t)$ and additional term $c'(t) \cdot \eta(t)$ are reproduced at the multiplier output. Multiplication of the noise $\eta(t)$ by $c'(t)$ means that the spreading code will act at the noise just as the initial signal in the transmitter. The component of the information signal $b(t)$ is narrowband, while the additional component $c'(t) \cdot \eta(t)$ is broadband. Hence, when the signal from the multiplier output is passed through a low-pass frequency filter, the most part of the false component $c'(t) \cdot \eta(t)$ will be filtered out. Thus, the effect of the interference $\eta(t)$ will be essentially decreased.

In the receiver, the integrator plays the role of the low-pass filter. The integration is carried out over a time interval, corresponding to the length of one bit of the information signal $0 < t < T$, and gives the value V. The decision is made with value V: if V is greater than the threshold zero value, we conclude that the binary one have been received in the interval $0 < t < T$, and if $V < 0$, the signal have been zero.

IV. SELF-SYNCHRONIZATION OF THE TRANSMITTER AND RECEIVER

In classical spread-spectrum systems, the synchronization of the transmitter and receiver is achieved, as a rule, with special circuits. The process of synchronization has two phases: establishment (entry) and maintenance.

In communications systems using chaotic oscillations, the questions of synchronization of the transmitter and receiver can be solved on the basis of the effect of self-synchronization of two systems with chaotic dynamics, or, in an important special case, on the basis of the effect of chaotic synchronous response.

Below we study the synchronization in the communications systems based on maps with stored information, and show that we can provide the synchronization between the transmitter and receiver without any special external circuits at any code sequence pre-stored in the maps, and with almost instant establishment of the synchronization.

In order to investigate the process of establishment of the synchronization, consider a communications system composed of a transmitter and a receiver, both containing identical 1-D maps with a random sequence stored in it. This 10,000-symbol random sequence assuming the values ± 1 plays the role a PN-sequence of classical spread-spectrum communications systems. In the transmitter, this sequence is modulated by a binary message symbol and is sent through the communications channel to the receiver. In the channel, the transmitted signal can be distorted by noise.

Note, that the proposed synchronization scheme has nothing in common with correlation-based synchronization schemes. Synchronization here is established with the help of the property of associative memory, inherent of these maps [11]–[17]. That is, the received sequence is fed to the map for recognition, and if it contains a sufficiently big undistorted fragment of the stored sequence, then without looking through and comparing with all the sequences stored in the map, an initial condition for the map is formed. Starting from this initial condition, we obtain the same sequence that was generated by the transmitter map, i.e., synchronization is established.

The synchronization scheme is shown in Fig. 3.

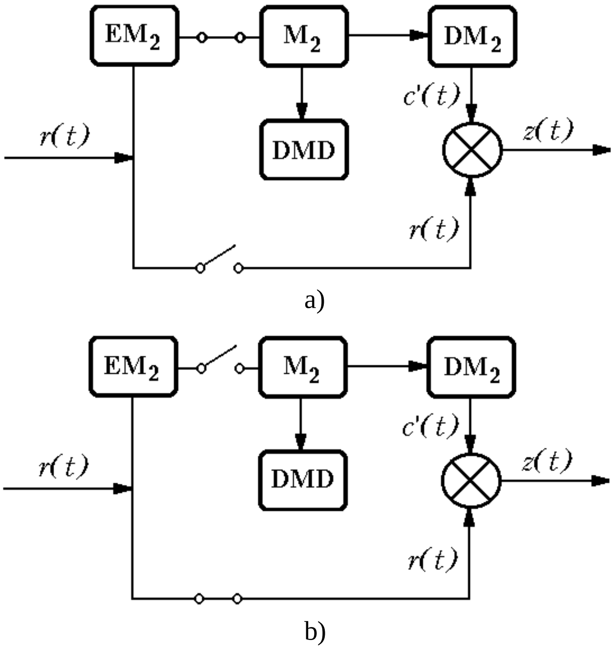

Figure 3: Synchronization scheme.

Let us consider an application of the maps with the code sequences stored as stable cycles. In this case, the synchronization of the transmitter and receiver can be organized as follows (Fig. 3).

Let the code sequences be stored in the maps of the transmitter and receiver at level q. At the beginning, a nonmodulated binary signal of the transmitter is fed to the channel. At this time, the receiver is in the state of waiting for synchronization (Fig. 3a). The binary signal coming to the receiver input is transformed by the encoder of the map EM_2 into a piece of a sequence represented by the symbols of the map alphabet. The initial condition for the map is formed then from this fragment. For this purpose, we take the first q elements of this piece and check if they make an initial condition corresponding to an information region of the map. When storing code sequences on the map, the slopes of all information regions are set equal. Other (noninformation) segments of the map have different slopes, so the test for the information segment is reduced to the test of the slope value of the linear segment of the map function corresponding to the initial condition.

If the initial condition corresponds to the information region, we iterate the map once, and compare the obtained value of the mapping variable to the value corresponding to the q elements of the sequence piece beginning from the second element $(a_2...a_{q+1})$. If the test is successful, we make a few other tests (usually no more than two–three), then the decision-making (DMD) decides that the synchronization is established.

If the first or one of the subsequent tests give negative answer, i.e., if the point corresponding to a q-element substring of the sequence piece do not coincide with a point of the iterated map, then the DMU makes decision on the absence of the synchronization. Then we shift by one element along the sequence piece and repeat the above operations. In the absence of noise, the synchronization is inevitably established in a few steps of the above operations, as is proved in [17].

These investigations indicate that the discussed procedure of the search for the initial conditions for the synchronization reliably operates even in the presence of external noise. However, in this case, the mean time of the establishment of synchronization increases.

The synchronization establishment time was estimated as the length of the received sequence, necessary to find the initial condition for the map. In the case of no external noise in the channel, the synchronization occurs very quickly, i.e., in less than ~ 30 symbols (the symbols are assumed to come one per time step). Indeed, the presence of a noise (Gaussian zero-mean noise) changes the situation, but, the system srill demonstrates good tolerance to external noises. The mean time of the synchronization establishment as a function of the noise level σ is shown in Fig. 4.

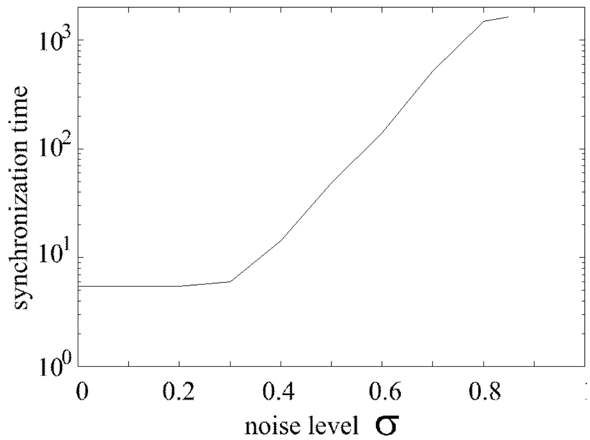

Figure 4: Synchronization establishment time versus the noise level σ.

As is seen from Fig. 4, up to $\sigma \approx 0.3$, the system is practically insensitive to the noise in the channel. Then the sync time grows exponentially, and at values close to $\sigma \approx 0.85$ and higher, synchronization becomes practically impossible.

At lower noise levels, synchronization occurs quickly, mostly after reception of a few symbols, as is seen from

Fig. 5, where the probability density distribution of the synchronization times is presented for $\sigma = 0.3$.

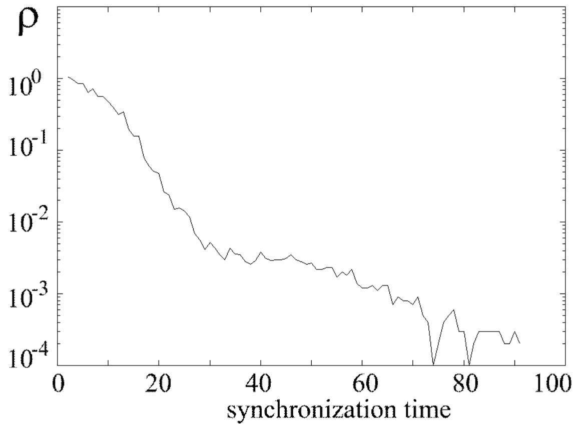

Figure 5: Distribution of the synchronization establishment time.

These results can be explained by the fact that at low noise levels the received sequence is corrupted but slightly, i.e., the corrupted symbols are seldom in it. This is illustrated in Fig. 6, where the average portion of the corrupted symbols in the received sequences is presented as a function of the noise level σ. At $\sigma \approx 0.85$, the average length of undistorted code sequence fragments becomes less than some critical length necessary for the operation of the associative memory, and synchronization becomes improbable.

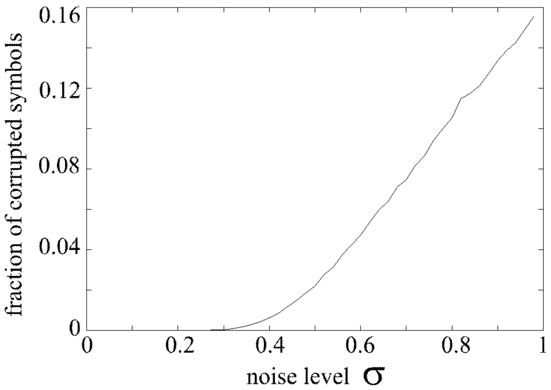

Figure 6: Number of the symbols corrupted in the channel as a function of the noise level σ.

Let us consider now the problem of synchronization of the transmitter and receiver in the case when the code sequences are stored in the maps of the receiver as unstable cycles. Such a map generates chaos. Hence, here we can apply the principle of synchronization of chaotic systems: any own trajectory of the chaotic attractor of a map can be synchronized by means of external stimulation of this map by a signal representing this very trajectory. In the case of the map with stored information, the most interesting is synchronization of special trajectories, i.e., the unstable cycles corresponding to the code sequences.

The dynamics of the autonomous map is described by the following equation

$$x_{k+1} = F(x_k). \tag{5}$$

When this map affects another map, the signal of this map x_k is mixed with some weight with the signal of the other map y_k. The weight α (coupling coefficient) determines the degree of the external effect

$$y_{k+1} = F(\alpha x_k + (1 - \alpha)y_k). \tag{6}$$

There is the minimum value of the coupling coefficient necessary for the synchronization of the stored code signal by this external stimulation. This minimum (threshold) value is uniquely determined by the relation [18]–[20]

$$|1 - \alpha| < \exp(-\lambda), \tag{7}$$

where

$$\lambda = \ln\left(\left[\prod_{k=1}^{N}\left|F'\left(x_k\right)\right|^{\frac{1}{N}}\right]\right) \tag{8}$$

is the Lyapunov exponent of the corresponding unstable cycle. If δ is the slope of information regions of the map function, then the condition for synchronization (6) can be rewritten as

$$|1 - \alpha| < 1/|\delta|. \tag{9}$$

Note, that for a stable cycle the threshold is zero, i.e., synchronization is possible for no matter how small coupling coefficient value. But for the unstable cycle, the coupling coefficient threshold value is positive and increases with increasing cycle instability. Thus, by external stimulation of the map with the information stored as unstable cycles we can synchronize this cycle by the external signal composed of this cycle points.

Consider one of the concrete schemes of synchronization of the transmitter and receiver using the principle of chaotic synchronous response (Fig. 7).

The signal in the channel consists of the fragments of the code sequence, multiplied either by +1 or by –1. It comes to the receiver input, where it is split in two. One copy of the signal is multiplied by –1, and then encoded using the generalized alphabet of the receiver map M_2, and the second copy is encoded using the generalized alphabet of the same map without multiplication (as if it were multiplied by +1). Then, both signals are fed to the input of the maps M_2.

Let at this time interval the signal of the code sequence multiplied by +1 be coming from the channel. Then the symbol sequence at the output of the first map will be same as that at its input. After decoding of this sequence into binary form and multiplying it with the incoming binary sequence, we obtain a sequence of ones. Averaging of this sequence over the length of the code message by the integrator, even in the case of distortions in the initial binary sequence allows us to recognize this “one” in the decision-making unit. The output sequence after the map M_2 of the signal passing through the second branch of the receiver will not coincide with the input sequence. So, by decoding it into binary sequence and multiplying by the input binary sequence, we will have a pseudorandom sequence of +1 and –1. Averaging of this sequence by the integrator gives the signal close to zero. Now the decision-making unit has to distinguish the signal close to one from the signal close to zero.

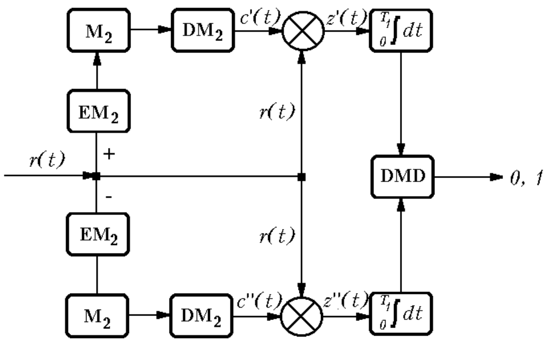

Figure 7: Self-synchronized receiver.

If at this time interval, a signal of the code sequence multiplied by –1 is coming from the channel, the similar consideration shows that at the integrator output of the first branch there will be a signal close to zero, while at the output of the second branch the signal will be close to –1.

Thus, at the output of the decision-making unit, a binary sequence corresponding to the transmitted information signal will be retrieved.

Both synchronization schemes do not require precisely synchronized clocks in the transmitter and receiver. The synchronization is established regardless of which fragment of the code sequence (cycle) is used. It is provided for any cycle used as a code sequence which allows organization of individual private communications channels.

The time of the synchronization establishment in the absence of interference is no longer than the duration of a few binary ones of the code sequence, and in the presence of not very large noise, the length of a single binary message.

ACKNOWLEDGMENT

This work is supported in part by the Russian Foundation for Basic Research, Grant No.97-01-00800 and the INTAS Grant (INTAS - 94 -2899).

References

[1] L. Pecora, T. Caroll. *Phys. Rev. Letters*, vol. 64, no. 8, p. 821. 1990

[2] L. Kocarev, K.S. Halle, K. Eckert, L. Chua, U. Parlitz. *Int. J. Bifurcation and chaos*, vol. 2, no 3, p. 709, 1992.

[3] U. Partlitz, L. Chua, L. Kocarev, K. Halle, A. Shang. *Int. J. Bifurcation and Chaos*, vol. 2, no. 4, p. 973, 1992.

[4] K. Cuomo, A. Oppenheim. *Phis. Rev. Lett.*, vol. 71, no 1, p. 65, 1993

[5] Yu. Belsky, A. Dmitriev. *Radiotechnika I Elektronika,* vol. 38, no 7, p. 1310, 1993.

[6] A. Volkovsky, N. Rul’kov, *Pis’ma v GTF*, vol. 19, no 3, p. 71, 1993.

[7] A. Kozlov, V. Shalfeev. *Pis’ma v GTF*, vol. 19, no 23, p. 83, 1993.

[8] H. Dedieu, M. Kennedy, M. Hasler, *IEEE Trans. Circuits Syst.*, vol. CAS-40, no 10, p. 634, 1993.

[9] K.S. Halle, C.W. Wu, M. Itoh, L.O. Chua. *Int. J. Bifurcation and Chaos*, vol. 3, no 2, p. 469. 1993.

[10] A.S. Dmitriev, A.I. Panas, S.O. Starkov, *Int. J. Bifurcation and Chaos*, vol. 5, no 3. p. 371. 1995.

[11] A.S. Dmitriev. *Radiotechnika i Elektronika*, vol. 36, no 1, p. 101. 1991.

[12] A.S. Dmitriev, A.I. Panas, S.O. Starkov S.O. *Phys. Lett. A*, vol. 155, no 8,9, p. 494. 1991.

[13] Yu.V. Andreyev, A.S. Dmitriev, L.O. Chua, C.W. Wu. *Int. J. Bifurcation and Chaos*, vol. 2, no. 3, p. 483, 1992.

[14] Yu.V Andreyev, A.S. Dmitriev. *Radiotechnika i Elektronika, vol*. 39, no 1, p. 104, 1994.

[15] Yu.V. Andreyev, Yu.L. Belsky, A.S. Dmitriev. *Radiotechnika I Elektronika, vol*. 39, no 1. p. 114, 1994.

[16] Yu.V. Andreyev, Yu.L. Belsky, A.S.Dmitriev, D.A. Kuminov. *IEEE Trans. on Neural Networks*, vol. 7, no. 2, .p. 290, 1996

[17] Yu.V. Andreyev, A.S. Dmitriev, D.A. Kuminov, L.O. Chua, C.W. Wu. *Int. J. Bifurcation and Chaos*, vol. 6, no 4, p. 627, 1996.

[18] A.Dmitriev, M.Shirokov, S. Starkov. *Proc. of NDES’95*, p. 277, Doublin, Ireland, 1995.

[19] A.S.Dmitriev, S.O.Starkov, M.E.Shirokov. *Izvestiya VUZov “PND”*, vol. 4,5, p. 40, 1996.

[20] S.Starkov, Yu.Andreyev, A.Dmitriev, M. Matveev, M. Shirokov. *Proc. of NDES’96*, p. 299, Sevillia, Spain, 1996.