A Genetic Programming Approach to Modeling of Diffusion Processes by using the DT-CNN and its Applications to Control

時永, 祥三

https://doi.org/10.15017/1074

出版情報：經濟學研究. 69 (1/2), pp.73-86, 2003-01-30. 九州大学経済学会
バージョン：
権利関係：

A Genetic Programming Approach to Modeling of Diffusion Processes by using the DT-CNN and its Applications to Control

Shozo Tokinaga

Abstract

This paper deals with the application of the Genetic Programming (GP) to modeling of diffusion process by using the DT-CNN (Discrete Time Cellular Neural Network) and its application to the control of chaos in the cells. We try to approximate the dynamics of diffusion process by using the observation of time series based on the GP. In the GP, the system equations are represented by parse trees and the performance (fitness) of each individual is defined as the inversion of the root mean square error between the observed data and the output of the system equation. By selecting a pair of individuals having higher fitness, the crossover operation is applied to generate new individuals. Simulation studies for approximating known dynamics by using the observed time series show good estimation of the systems equations. The condition for the propagation failure of the autowave is also discussed based on the estimated equations. It is shown that the estimated threshold value for the propagation failure is close to the result of simulation study. Then, we apply the control method to stabilize the chaotic dynamics in the DT-CNN. In our control, since the system equations are estimated, we only need to change the input so that the system moves to the stable region. By assuming the targeted dynamic system $f(x(t))$ with input $s(t) = 0$ is estimated by using the GP (denoted $\hat{f}(x(t))$), then we impose the input $u(t)$ so that $x_f = \hat{x}(t+1) = \hat{f}(x(t)) + s(t)$ where x_f is the fixed point. Then, the next state $x(t+1)$ of targeted dynamic system $f(x(t))$ is replaced by $x(t+1) + s(t)$. The approximation and control method are applied to chaotic dynamics in DT-CNNs, and they are lead to a fixed point or limit cycle.

1 Introduction

In the last few years, there have been many advances in the study of pattern formation and wave phenomena in the fields such as physics and chemistry[1][2]. These studies have been carried out mostly experimentally and by simulation of nonlinear partial differential equations. An alternative approach is the direct analysis of nonlinear dynamics. For example, it is also showed that interconnections of a sufficiently large number of simple dynamic units such as cellular neural networks (CNNs) can exhibit extremely complex and self-organizing behavior [1]-[5].

The study of chaos has provided new conceptual and theoretical tools enabling us to understand complex behavior and control them [5][6]. Even though the chaotic behavior seems to be universal, and shows up the deterministic model in electrical circuits, lasers, chemical reactors and many other systems, but the applicability means that we learn about chaotic behavior by

studying simple mathematical models.

Specifically, in the social science, we must cope with the task to estimate the system dynamics, and it is a kind of inverse problem existing as a counter part of ordinary approach in numerical analysis where we analyze given dynamic system. Sometime, we can use only input and output pair to describe the system dynamics, and a kind of approximation is necessary.

This paper deals with the application of the Genetic Programming (GP) to modeling of diffusion process by using the DT-CNN (Discrete Time Cellular Neural Network) and its application to the control of chaos in the cells [7][8].

Even though the (analogue) CNN is capable of real time processing of digital data such as images, but for the modeling and simulation of diffusion process such as autowaves, the DT-CNN is promising to provide much tractable tool for computer simulations. Furthermore, the universality included in the CNN is succeeded to the DT-CNN wherein the system dynamics is described by the difference equations. In this paper we try to approximate the dynamics of diffusion process for the DT-CNN based on the observation of time series.

In previous works, we demonstrated that the GP provides very efficient tool to approximate chaotic dynamics by using the observed time series, and is also applicable to the control of chaos [9][10]. The estimation method has several advantages compared to conventional numerical methods, especially for the cases where the number of available observation is restricted. In the GP, the system equations are usually represented by parse trees (called individuals as well as in the Genetic Algorithm :GA) [11]-[15]. One parse tree corresponds to a system of dynamic equations. In the GP, the performance of each individual (called as fitness) is evaluated by comparing the output generated by the system equation corresponds to the individual with the observed data to be approximated. We apply the crossover operations to the individuals possessing relatively higher fitness in the reproduction phase. The mutation operation is also applied to reintroduce some diversity in an otherwise stagnant population.

In the DT-CNN, it is known that if the diffusion coefficients are small enough, the autowave is hindered to propagate further. We can apply the Keener's result to the approximated equations for the autowave propagation [17]. Since the approximated forms of system equation are obtained , it is easy to apply the estimation method for finding the condition for propagation failure rather than the simulation based studies.

In the next step, we utilize the approximated equations for the control of chaotic dynamics in the DT-CNNs. If a chaotic attractor is given, one approach for stabilizing the system performance is to make a time-dependent small perturbations in a accessible system parameters, rather than completely changing system dynamics by making large and costly alternation.

After estimating the stable region (fixed point or limit cycle) based on the approximation, the perturbation is obtained to stabilize the chaotic dynamics [10]. We assume that the dynamic system $f(x(t))$ with input $s(t) = 0$ is estimated by using the GP, and is denoted as $\hat{f}(x(t))$. Then, the control method is derived straightforward by using the approximation. We impose the input $s(t)$ so that $x_f = \hat{x}(t)$ where $x(t)$ is governed by a difference equation and x_f is the fixed point.

In the following, in Sect.2, we show the fundamentals of the DT-CNN treated in the paper. In Sect.3, the approximation of the dynamics of the DT-CNN is shown based on the GP, and a simulation study is carried out by using the observation of state variables generated by known dynamics. Sect 4 shows the estimation of the diffusion coefficients with which the autowaves are hindered to propagate based on the approximated equations. In Sect.5, we show the control of chaotic dynamics of the DT-CNN by imposing small perturbation based on the approximated system equations.

2 Fundamentals of autonomous DT-CNN

In the computational tasks such as image processing, Chua and Roska proposed the CNN as a alternative with a clear mathematical description by partial derivatives and analogue VLSI implementation [1]-[3].

We have several types of CNN, namely autonomous CNN and non-autonomous CNN. In the following, we restrict ourselves to the autonomous CNN but having diffusion terms. As is pointed out, these CNN can simulate very wide class of partial differential equations [1].

Instead of the conventional CNN represented in a continuous time equations, we use discrete systems (DT-CNN) to provide much tractable tool for computer simulations. Because, for the modeling and simulation of diffusion process such as autowaves, the difference equations representing the behavior of DT-CNN is easy to implement for the simulation study. Furthermore, the universality of included in the CNN is succeeded to the DT-CNN.

In the following, it is assumed that the system equations are shown as

$$x_i(t+1) = F(x_i(t)) + D\nabla^2 x_i(t) \tag{1}$$

where $x_i(t)$ is n-dimensional vector at the location i, and $\nabla^2 x_i(t)$ is the diffusion term with diffusion coefficient D.

The diffusion term included in equation is discretized in a one-dimensional DT-CNN such as (in the following, the time t is omitted in the state variables if not necessary)

$$Diff(x_i(t)) = x_{i+1}(t) + x_{i-1}(t) - 2x_i(t) \tag{2}$$

In the same way, at each cell c_{ij} at location (ij) in a Cartesian coordinate system, we have

$$\begin{aligned} Diff(x_{ij}(t)) = {} & x_{i+1,j}(t) + x_{i-1,j}(t) + x_{i,j+1}(t) + x_{i,j-1}(t) \\ & -4x_{i,j}(t) \end{aligned} \tag{3}$$

At the beginning, we focus on the autonomous DT-CNN including Chua's circuit which exhibits extremely rich behavior via various bifurcations depending on the parameters. In the following, we consider two types of two dimensional DT-CNN namely, DT-CNN-1 and DT-CNN-3 having three state variables u_{ij}, v_{ij}, w_{ij} [1]-[3].

(DT-CNN-1)

$$u_{ij}(t+1) = u_{ij}(t) + h\alpha[v_{ij}(t) - f(u_{ij}(t))] + D_u Diff(u_{i,j}(t)) \tag{4}$$

$$v_{ij}(t+1) = v_{ij}(t) + h[u_{ij}(t) - v_{ij}(t) + w_{ij}(t)] \tag{5}$$

$$w_{ij} = w_{ij}(t) - h\beta v_{ij}(t) \tag{6}$$

$$f(x) = 0.5[(s_1 + s_2)x + (s_0 - s_1)(|x+1| - |x-1|) + \varepsilon \tag{7}$$

(DT-CNN-3)

$$\begin{aligned} u_{ij}(t+1) = {} & u_{ij}(t) + h\alpha[v_{ij}(t) - u_{ij}(t) - g(u_{ij})t)] \\ & + D_u Diff(u_{i,j}(t)) \end{aligned} \tag{8}$$

$$\begin{aligned} v_{ij}(t+1) = {} & v_{ij}(t) + h[u_{ij}(t) - v_{ij}(t) + w_{ij}(t)] \\ & + D_v Diff(v_{i,j}(t)) \end{aligned} \tag{9}$$

$$w_{ij}(t+1) = w_{ij}(t) - h\beta v_{ij} + D_w Diff(w_{i,j}(t))) \tag{10}$$

$$g(x) = s_1 x + 0.5(s_0 - s_1)(|x+1| - |x-1|) \tag{11}$$

where the notation $Diff(.)$ represents the diffusion term defined in equation(3).

2.1 DT-CNNs and propagation of autowave

Autowaves are the most frequently encountered wave phenomena in the natural phenomena. Since a CNN is a kind of excitable media in which each individual neuron exhibits one stable state, under excitation or forcing one or more cells change their values dramatically [1]-[3]. In a fluctuating discrete media, each neuron operates in a stable period regime, whereas in a chaotic media the regime of each cell is chaotic.

We will consider the modeling and analysis of media represented by bistable cells, used in the generation of traveling waves. It has been proved by Keener that propagation failure cannot observed in a continuous, one-variable, homogenous reaction-diffusion system. Therefore, in the following, we use a discrete system .

Here, we reproduce these phenomena and give an intuitive explanation of underlying process of autowave by using the known Chua's circuit [4]-[6]. Fig.1 shows the propagation of autowave on a homogenous DT-CNN cells in a 10×10 array having sufficiently large diffusion coefficients. At first, suppose that all of the cells are in the initial state S_A, and then introduce a constant forcing term R in the equation on the cell c_{11} as shown in Fig.1 (upper). The trajectory of c_{11} will settle down to a new equilibrium point S_B. Hence, by forcing the cell c_{11} for an appropriate time interval, we see the transient on cells as a traveling wave front whose global view is shown in Fig.1(lower). .

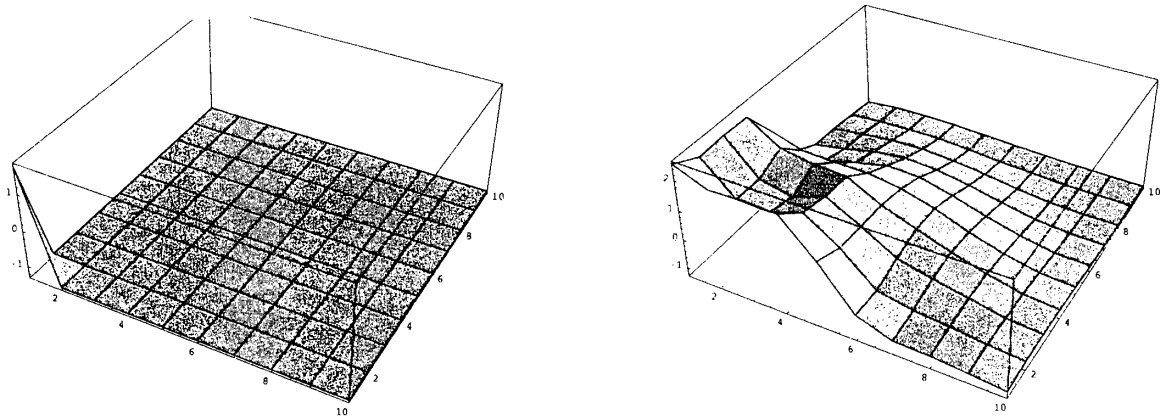

Fig.1 Porpagation of autowave(left:initial state, right:propagation)

In the following discussion, we use the GP to approximate the dynamics of homogenous DT-CNN by using the observation on a certain cell including the diffusion coefficients. Then, we can analyze the conditions for propagation and also the control of chaotic dynamics based on the approximation.

3 Approximation of system dynamics by the GP

3.1 Applying the GP

In real applications, usually it is assumed that only the observation of state variables are given and the functional form of the system dynamics is unknown, and should be estimated by a kind of approximation. The GP is an extension of Genetic Algorithm (GA), but its elements consist of arithmetic expression and variables[11]-[15]. A tree structure corresponds to a system dynamics and a set of trees structure (population of individuals) consist of search space for approximation.

The tree structure representing the $F(x_i(t))$ in equation (1) is, for example , $u_i(t) + v_i(t)$. But it is written usually in prefix form such as

$$(6.43 \times y_1 - y_2) \times (y_3 - 3.54) \rightarrow \times - \times 6.43 y_1 y_2 - y_3 3.54 \tag{12}$$

The equation represented by using the prefix are interpreted based upon the stack operation. We begin to scan the prefix representation, and if we meet a set of operator and two terminals (operands) then we perform the calculation and push down the result into the stack again.

For checking the validity of underlying parse tree, the so-called stack count (denoted as *StackCount* in the paper) is useful [15]. The *StackCount* is the number of arguments it places on minus the number of arguments it takes off from the stack. The cumulative *StackCount* never becomes positive until we reach the end at which point the overall sum still needs to be 1.

By using the *StackCount* we can see which loci on the prefix expression is available to cut off the tree for the crossover operation, and we can validate whether the mutation operation is allowed. If final count is 1, then the prefix representation (tree) corresponds properly to a system equation. Otherwise, the tree structure is not relevant to represent the equation.

Usually, we calculate the root mean square error ($rmse$) between $x(t)$ and $\tilde{x}(t)$ where $\tilde{x}(t)$ is the prediction of $x(t)$, and use it as the fitness. By selecting a pair of individuals having higher fitness, the crossover operation is applied to generate new individuals.

Crossover operations

Contrary to the operation in GA, the crossover operation in GP is applied to restricted cases. Then, we can not choose arbitrary loci in the string of individuals and replace the parts of two tree structures. The two subtrees are extracted and swapped each other.

To keep the crossover operation always producing syntactically and semantically valid programs, we look for the nodes which can be a subtree in the crossover operation and check for no violation. By using the *StackCount* already mentioned, we know the subtrees which are the candidate for the crossover operation. The basic rule is that any two loci on the two parents genomes can serve as crossover points as long as the ongoing *StackCount* just before those points is the same. The crossover operation creates new offsprings by exchanging sub-trees between two parents.

Mutation

The goal of the mutation operation is the reintroduction of some diversity in an population. Two types of mutation operation in GP is utilized to replace a part of the tree by another element.

(Global mutation :G-mutation)

Generate a individual I_s, and select a subtree which satisfies the consistency of prefix representation. Then, select at random a terminal in the individual, and replace the terminal by the subtree of the individual I_s.

(Local mutation:L-mutation)

Select at random a locus in a parse tree to which the mutation is applied, we replace the place by another value (a primitive function or a variable).

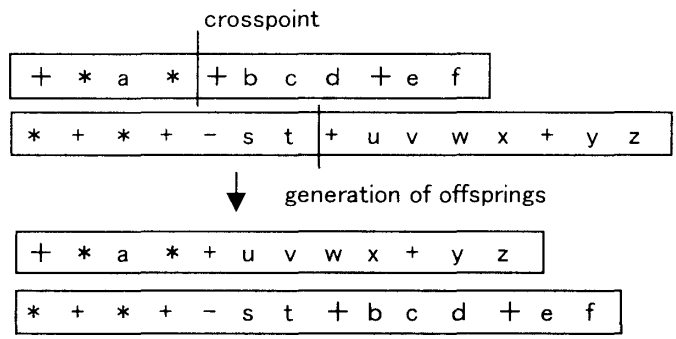

Fig.2-Schematic diagram of crossover operation.

3.2 Optimizing the constants

Even though the structure of the equations describing the system dynamics is improved by the GP, but the constants included in the prefix representation are usually only swapped from one individual to another individual, and never changed from initial value. Besides the mutation operations, there is no way to optimize the constants. In previous works, we utilized the GA (Genetic operation) as one procedure in the GP to optimize the constants in the individuals, but it is time consuming [9][10].

In the paper, we optimize the constants by using conventional steepest descent algorithm to simplify the procedure [8]. However, the steepest descent algorithm for optimizing the constants is applied only once for each GP iteration. Because the individuals having higher fitness will remain in the pool for a long time, and it is expected that sufficient times of incremental change of the constants are applied to these individuals iteratively. On the other hand, it is not useful to optimize the constants in the individuals with lower fitness which are ultimately removed from the pool.

Define the difference between the observation of the time series $x(t)$ and the prediction $y(t,a)$ obtained by interpreting a certain individual as follows where the difference is accumulated for $t = 1, 2, ..., T$.

$$H(a) = \sum_{t=1}^{T} [y(t,a) - x(t)]^2 \tag{13}$$

where $a = (a_1, a_2, ..., a_m)$ are the constants included in the individual. The incremental value Δa to optimize the constants a are given by

$$\Delta a_i = -\alpha \partial H(a)/\partial a_i \tag{14}$$

where α is used to accelerate the convergence.

The partial derivative is obtained by interpreting the prefix representation. Then, in each GP operation, each individual is interpreted three times to evaluate $y(t,a)$, and Δa_i , and another $y(t,a)$ after the incremental change of a.

From many experimental data we can recognize that the problem arises from the convergence to the local minimum can be avoided in the tasks treated in the paper.

3.3 Algorithm of the GP

We iteratively perform the following steps until the termination criterion has been satisfied.
(Step 1)

Generate an initial population of random composition of possible functions and terminals for the problem at hand. The random tree must be syntactically correct program.
(Step 2)

Execute each individual (evaluation of system equation) in population by applying the optimization of the constants included in the individual. Then, assign it a fitness value giving partial credit for getting close to the correct output. Then, sort the individuals according to the fitness S_i.
(Step 3)

Select a pair of individuals chosen with a probability p_i based on the fitness. The probability p_i is defined for ith individual as follows.

$$p_i = (S_i - S_{min}) / \sum^{N} (S_i - S_{min}) \tag{15}$$

where S_{min} is the minimum value of S_i, and N is the population size.
(Step 4)

Then, create new individuals (offsprings) from the selected pair by genetically recombining randomly chosen parts of two existing individuals using the crossover operation applied at a randomly chosen crossover point. The algorithm is the same as the roulette strategy. If the individual having highest fitness is not included, then we apply the strategy of elite preservation. Iterate the procedure several times to replace individuals with lower fitness.
(Step 5)

If the result designation is obtained by the GP (the maximum value of the fitness become larger than the prescribed value), then terminate the algorithm, otherwise go to Step 2.

We can evaluate the approximation of chaotic dynamics generated by known system equations. As a result, the approximation error of system equations for the artificial chaotic time series generated by known 1-D and 2-D dynamics such as (Logistic map, Henon map, and Ushiki map) are about 1.0e-7 and are very small [9][10].

3.4 Approximation of Known dynamics

The method is at first applied to the estimation of known dynamics of DT-CNN by using the observation of autowaves. For the first example of approximation, the functions included in the dynamics of DT-CNNs are selected from Chua's circuits. The functions $f(u)$ and $g(u)$ in equations (7) and (11) are characterized by three linear segments on the u-axis.
(DT=CNN-1)
$f(x) = 0.5[(s_1 + s_2)x + (s_0 - s_1)(|x+1| - |x-1|) + \varepsilon$
$h = 0.1, \alpha = 9, \beta = 30, s_1 = s_2 = 2/7$
$s_0 = -1/7, \varepsilon = 1/14$

(DT-CNN-3)
$g(x) = s_1 x + 0.5(s_0 - s_1)(|x+1| - |x-1|)$
$h = 0.1, \alpha = 9, \beta = 19, s_0 = -1.143, s_1 = 0.714.$

It is seen that the DT-CNN-1 have two stable points, $S_A = (u_a, v_a, w_a) = (-1.25, 0.1.25)$ $S_B = (u_b, v_b, w_b) = (1.25, 0. - 1.25)$
and the DT-CNN-3 have a limit cycle. Since the cells of the DT-CNN are homogeneous, it is relevant to approximate the system function on a certain cell by using the observations of state variables on the cell.

For the DT-CNN-1, it is assumed that at the beginning, all of the cells have the same initial state, say S_A, then we change the state of cell c_{11} from S_A to S_B at time $t = 0$. The state S_B propagate to another cells, and the change of the state of a certain cell over the time $t = 0$ to $t = T$ is used to approximate the system dynamics of the DT-CNN-1. In the same way, the propagation of autowave on the DT-CNN-3 is used to approximate the dynamics.

The parameters for the simulation study are as follows.
operators:$||, +, -, \times$
population size=1000
maximum length of array for u,v,w=90
data length of time series=20

Table 1 shows the final value of $n - rmse$ (defined as the $rmse$ divided by the standard deviation of the time series) for the approximation of system equations.

$$n - rmse = [\sum (x(t) - \tilde{x}(t))^2]^{1/2} / N_s \sigma \tag{16}$$

Table 1 also shows the number of iterations denoted as N_p at which the point we have almost the same functional form of equations as the ultimately obtainable results.

Concerning the iteration of algorithm to terminate the approximation, about after 50 generations of GP we have good approximation (identification) for the chaotic data.

Table 1-Approximation error

name	n-rmse	N_p
CNN-1	0.0002	59
CNN-3	0.0003	75

4 Propagation failure of autowave

4.1 Diffusion coefficients and propagation failure

Then, the result is used to estimate the threshold value for propagation of autowaves on DT-CNNs. Since the diffusion coefficient affects the propagation and propagation failure of autowave in DT-CNNs, we must know the minimum (threshold) value of diffusion coefficient from estimated system equations.

Fig.3 shows an example of propagation failure on a DT-CTT whose cells have sufficiently large diffusion coefficients expect for the cells located in the center of the plane. As is seen from Fig.3, the autowave is hindered on the cells which have small diffusion coefficients, and the autowaves are forced to propagate along a kind of detour bypassing the close obstacle, but fills in the open one on the plane for further propagation. .

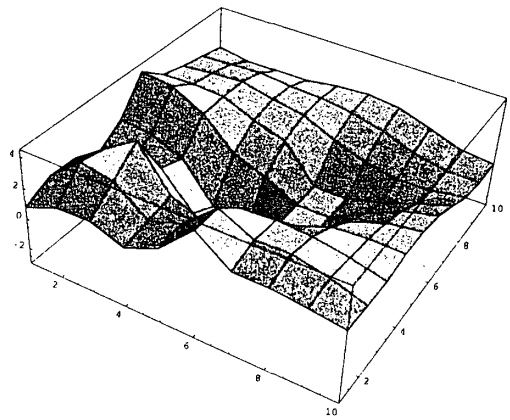

Fig.3-Propagation failure of autowave

Originally, it is vary hard to calculate the threshold value from the observation, however, since we know the functional form, we can easily estimate the threshold for diffusion coefficients based on Keener's result [16].

Supposing that the variable x is normalized in $[0,1]$, and

$f(0) = f(\alpha) = f(1) = 0$

$f(x) \neq 0, \quad for \quad x \neq 0, \alpha, 1 \quad (17)$

$f(x) < 0 \quad for \quad 0 < x < \alpha$

$f(x) > 0 \quad for \quad \alpha < x < 1 \quad (18)$

$f'(x_0) = f'(x_1) = 0, 0 < x_0 < \alpha < x_1 < 1$

$f'(x) \neq 0 \quad for \quad \neq x_0, x_1 \quad (19)$

Then, if two values x_0, x_1 satisfying following relations are found, then the propagation wave is hindered (stable nonuniform standing solutions are found).

(i) $2(x_0 - 1) - f(x_0)/d = 0 \quad (20)$

(ii) $2x_1 - f(x_1)/d = 0 \quad (21)$

(iii) $f'(x) < 2d \quad on \quad 0 \leq x \leq x_0, x_1 \leq x \leq 1 \quad (22)$

We can extend the result for three state variable system. At first, consider the propagation of autowave on a cell $c(i,j)$. It is easily observed that the propagation in the direction of i axis in independent of the direction for j axis, since we assume that the structure of DT-CNN is homogenous. Since the wave front propagation is essentially unaffected by the variable $w(t)$, we consider the propagation of waves in the reduced system equations where the state w is set to be a constant such as w^0.

Moreover, we are focusing on the propagation of autowave on a plane of cells, the changes of state variables are tentatively ignored. Then, we observe that the equation for the variables $u(t)$ and $v(t)$ is uniquely solvable for $v = v(u)$, and finally we obtain the function for testing the propagation of autowave represented in a single time-independent variable u.

We can translate the condition for functions $F(u)$ with $F(u^-) = F(u^+) = 0$ where u^- and u^+ are the smallest and largest zeros of $F(u)$, respectively. If we make the appropriate change of scale, we found that the conditions are hold if there are values u^0, and u^1 for which

$$F(u_0)/2(u_0 - u^+) = d, F(u_1)/2(u_1 - u^-) = d \quad (23)$$

4.2 Applications

The threshold value of the diffusion coefficient is estimated by the Keener's conditions based on the approximated system equations. Then, we evaluate the theoretical value by comparing it with the result of simulation study. As already mentioned, the DT-CNN-1 has two stable state S_A and S_B, and if the propagation is hindered the initial state of the cells are not unchanged , even though the state of the cell c_{11} is changed from S_A to S_B and the source of a autowave is generated. Otherwise, the deviation of the state variable from the initial state defined by the following equation will be large.

$$r_d = \sum[(u(t) - u_a)^2 + (v(t) - v_a)^2 + (w(t) - w_a)^2] \tag{24}$$

By observing the deviation r_d for several value of diffusion coefficients, we can estimate the threshold value of diffusion coefficients which determine the propagation of autowaves.

Table 2 shows the comparison of the diffusion coefficients for the DT-CNN-1 obtained from the approximated equations and the simulation studies. As a result, we have very good estimation for the diffusion coefficients with which the autowave is hindered to propagate.

In case of DT-CNN-3, we must consider the combination of the diffusion coefficients D_u, D_v, D_w for the variables u, v, w. However, in these cases, the functional form is not able to be reduced to a simple form of single variable, and the Keener's result is not applicable. Therefore, we restirct ourselves to the case where we assume a single diffusion coefficient is not zero, and other diffusion coefficients are zero.

Table 3 shows the result of estimation of the diffusion coefficients for these resrticted cases. As is seen from Table 3, the result obtained from the approximated system equations are close to the result estimated by the simulation studies. Especially, it is seen that the diffusion coefficient D_w play very small role in the propagation failure of the autowave.

Table 2. Estimatation of D_u (DT-CNN-1)

	by approximated function	by simulation
D_u	0.301	0.292± 0.01

Table.3 Estimation of D_u, D_v, D_w(DT-CNN-3)

	by approximated function	by simulation
D_u	3.1	3.2±0.15
D_v	2.7	2.73±0.12
D_w	120.0	130.3±0.5

5 Control of chaos

5.1 Control using approximated dynamics

Finally, the approximation method of system dynamics is applied to the control of cells in DT-CNNs to lead them into a stable region (fixed points and limit cycle). Our method is simple and easy to apply compared to conventional methods such as the OGY method [17].

The OGY method is well known control scheme by using the linearlization of a trajectory $x(t+1) = f(x(t), s(t))$ where $s(t)$ is a perturbation input for control [17]. Then, we have a linearlization at a certain fixed point x_f as

$$x(t+1) - x_f = A(x(t) - x_f) + bs(t) \tag{25}$$

where $A = D_x f(x_f, 0), b = D_s f(x_f, 0)$ and are able to be experimentally determined. We then change $s(t)$ slightly from zero to some value (determined by the eigenvalues and eigenvectors of the matrix A) so that the state moves to a stable manifold.

In previous works, we demonstrated that by comparing the result of control by the OGY method with our control scheme, it is revealed that the OGY method still effective if the noise is relatively small, but in the region of with higher level of noise the control fails almost always, while our method still provide in some extent an effective control. Furthermore, the time to complete the control is relatively large in the OGY control compared to the case using the control of our method.

In the continuous chaotic system, the feedback control is applicable to stabilize the chaotic dynamics by imposing the feedback to the sysmte equations [18].

$$dx(t)/dt = f(x(t)) + K[\hat{x}(t) - x(t)] \tag{26}$$

where $\hat{x}(t)$ is a fixed point or a limit cycle, and K is a feedback gain. However, the gain for control must be determined experimentally, while the system equation is usually unknown.

In our system the system equations are approximated and estimated, the input for control is easily obtained. The method is applied to the chaotic DT-CNN to prove the effectiveness of the method. Consider a non-linear dynamical system

$$x(t+1) = f(x(t)) + s(t) \tag{27}$$

where we assume that $x(t)$ is the multi-dimensional state vector, and $s(t) = (s_u(t), s_v(t), s_w(t))$ is the only available control (multi-dimensional) parameter which we allow to vary in a range. We assume further that we are not far apart from the neighborhood of some steady state x_f (fixed point), which we want to stabilize by choosing an appropriate sequence of admissible input.

Off course, if we need to control the state to a limit cycle, then we replace the fixed point x_f by the trajectory of the limit cycle. Therefore, in the following discussion we treat the case where the target of the control is a fixed point x_f.

We assume that the dynamical system $f(x(t))$ with $s(t) = 0$ is estimated by using the GP, and is denoted as $\hat{f}(x(t))$. Then, the control method is derived straightforward by using the approximation. Since we can obtain the estimated value $\hat{x}(t+1)$ for the next state, we impose the input $s(t)$ so that

$$x_f = \hat{x}(t+1) + s(t) = \hat{f}(x(t)) + s(t) \tag{28}$$

The control of chaos is attained by moving the next state $x(t+1)$ to $x(t+1) + s(t)$, while $\hat{f}(x(t))$ is very close to $f(x(t))$, and $\hat{x}(t+1)$ is very close to $x(t+1)$.

We summarize the algorithm of control to stabilize the system into a steady state as follows.
(Step 1)finding steady state

Since we know the functional forms of system equations obtained by the approximation using the GP, we can find the steady state x_f by numerical computation.
(Step 2) calculation of input

We determine an appropriate value of input $s(t)$ so that

$$x_f = \hat{f}(x(t)) + s(t) \tag{29}$$

where $x(t)$ is the observed current state. This implies that x_f will be stabilized for the system equations and arbitrary initial conditions if and only if the next state $\hat{x}(t+1)$ is equal to x_f.
(Step 3) replace next state$x(t+1)$ by $\hat{f}(x(t)) + s(t)$

Then, we replace the next state $x(t+1)$ of the original system by $\hat{x}(t+1) = \hat{f}(x(t)+s(t)$ even if the original system $f(x(t))$ leads the next state to $x(t+1) = f(x(t))$. In the simulation study, the next state $x(t+1)$, for example, generated by the Henon map, is replaced by $x(t+1)+s(t)$. The process forces the observed state move to the steady state.

(Step 4) iterate until completion of control

Iterate the procedure Step 2 through Step 3 until the system fall into the region of stabilization.

Since the system is assumed to be nonlinear, the application of linear input will succeed only in a neighborhood of around x_f. Due to the bound, we have to correct the parameters in the next iterations according to the same scheme where we hope to need one iteration of the new set less than the step before.

Since we see a dependency between the maximum parameter perturbation and the expected time to achieve control, there is a trade-off between the maximum allowed parameter changes and the expected time to achieve the target. Since the input $s(t)$ may not be too large, there exist some lower limit of expected time which may still be large.

5.2 Applications

Fig.4 shows the process of control of DT-CNN-1 whose initial values are given by using the uniformly distributed random number. The upper diagram of Fig.4 shows the transition of the state $u(t)$ in a certain cell. The lower side of Fig.4 denotes the corresponding input signal $s(t)$ to control the chaotic behavior of the DT-CNN-1. As is seen from Fig.4 , the state of DT-CNN-1 is moved to a fixed point after imposing a input $s(t)$ at a time point.

In a similar manner, the process of control of the DT-CNN-3 is depicted in Fig.5, where the upper diagram is the controlled state of $u(t)$ in a certain cell, and the lower signal means the corresponding input. The state $u(t)$ is moved to a limit cycle, and we need only a small input at a time point to complete the control.

Table 3. summarizes the mean value of the absolute value of the input signal , and the time steps necessary for controlling the state to the steady state.

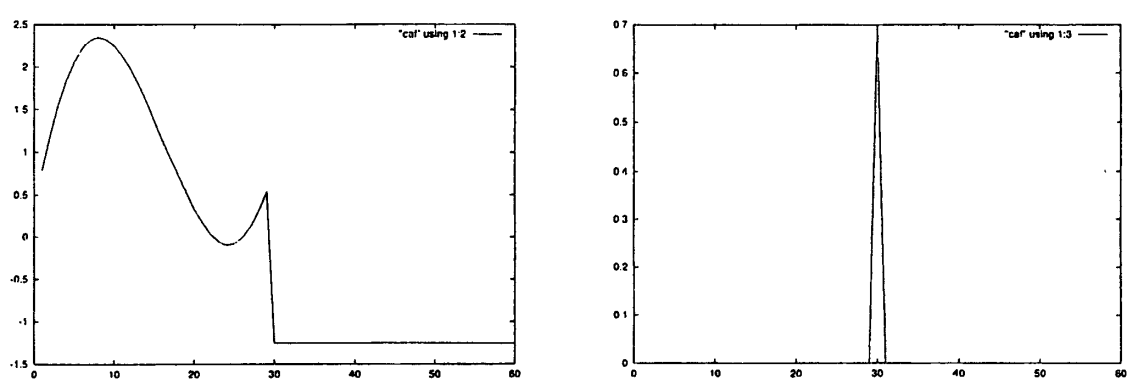

Fig.4-Control of DT-CNN-1 (left:controlled signal($u(t)$),right:input $s_u(t)$)

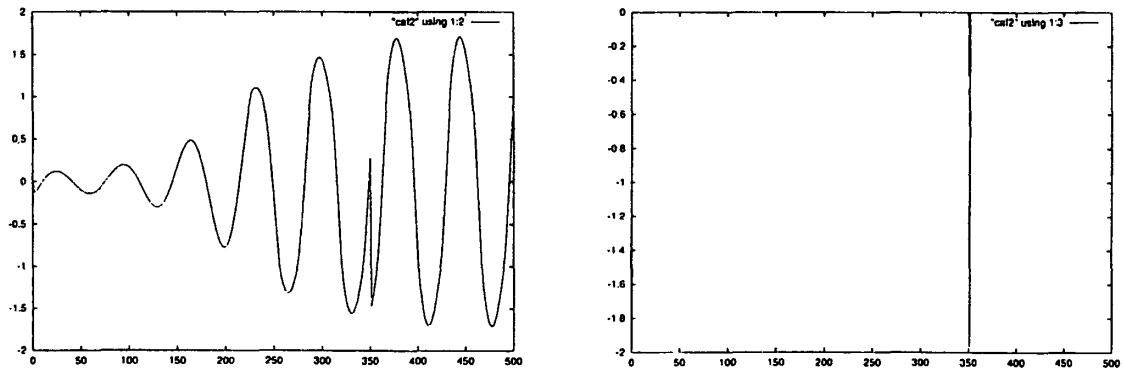

Fig.5-Control of DT-CNN-3 (left:controlled signal($u(t)$),right:input $s_u(t)$)

Table 4. Control result (mean inputs, steps for control)

name	s_u	s_v	s_w	N_c
DT-CNN-1	0.30	0.01	0.27	1
DT-CNN-3	1.5	0.02	1.8	1

6 Conclusion

This paper showed an estimation method for the dynamics of the DT-CNN by using the GP, and its applicxation to the control method of chaotic dynamics by using the approximated system equations. In the GP, the system equations were represented by parse trees and the performance (fitness) of each individual was defined as the inversion of the root mean square error between the observed data and the output of the system equation.

In the simulation study, the system dynamics of known DT-CNNs are estimated by using the observation of state variables, and the prediction error was discussed. The condition for the propagation of autowaves is also estimated by the approximation. In our control, since we know the approximation of the functional form of the dynamics, we only need to impose the input so that the system moves to stable region.

The problems remained to be solved include the extension of the method to various real time series and the improvement of control, and further works will be continuously done by the authors.

References

[1] L.O.Chua and L.Young: "Cellular neural network:Theory and practice", IEEE Trans.,Cirsuits Syste., vol.35,no.10, pp.1257-1290,1988.

[2] L.O.Chua and T.Roska: "The CNN paradigm", IEEE Trans.,Cirsuits Syste., vol.40,no.3, pp.147-156, 1993.

[3] L.O.Chua et.al: "Autonomous cellular neural networks: A unified paradigm for pattern formation and active wave propagation ",IEEE Trans. Circuits Syst., vol.42, no.10,pp.559-578,1995.

[4] L.Pivka:Autowave and spatio-temporal chaos in CNNs-Part I and Part II", IEEE Tanns.Circuits Syst., vol.42, no.10, pp.638-650, 1995.

[5] L.Pivka: "Autowave and spatio-temporal chaos in CNNs- Part II", IEEE Tanns.Circuits Syst., vol.42, no.10, pp.638-650, 1995.

[6] G.Hu, L.Pivka and A.L.Zheleznyak: "Synchronization of a one-dimensional array of Chua's circuits by feedback control and noise", IEEE Trans. Cirsuits Syst., vol.42, no.10, pp.736-741, 1995.

[7] S.Tokinaga and M.Yababe: "On the modeling of diffusion process of payment system based on the CNN and its applications ", Technical Report of IEICE, vol. NLP-200-141,pp.9-16,2001.

[8] S.Tokinaga and M.Yababe: "Applying the Genetic Programming to modeling of diffusion processes by using the CNN and its applications to control ", Proc. NOLTA2001, vol.2,pp.222-333, 2001.

[9] Y. Ikeda and S.Tokinaga: "Approximation of chaotic dynamics by using smaller number of data based upon the genetic programming", Trans. IEICE, vol.E83-A,no.8, pp.1599-1607, 2000

[10] Y.Ikeda and S.Tokinaga: "Controlling the chaotic dynamics by using approximated sysmtem equations obtained by the genetic programming ", Trans. IEICE vol.E84-A,no.9, pp.1599-1607, 2001

[11] J.R.Koza:Genatic Programming,MIT Press, 1992

[12] J.Koza:"Genetic programming: "A paradigm for genetically breeding populations of computer programs to solve problems", Report No.STAN-CS-90-1314, Dept.of.Computer Science Stanford University, 1990

[13] J.Koza: "Evalution and subsumption using genetic programming", Proc of the First European Conference on Artificial Life, MIT Press, 1991.

[14] J.Koza and J.P.Rice: "Automatic programming of robots by using genetic programming", Proc. of 10th AAAI, 1992.

[15] M.J.Keith and M.C.Martin: "Genetic programming in C++: Implementation issues", in (ed) K.E.Kinnerar,Jr.,Advance in Genetic Programming MIT Press, 1994.

[16] J.P.Keener: "Propagation and its failure in coupled systems of discrete excitable cells," SIAM J. Appl. Math, vol.47, no.3, pp.556-572,1987.

[17] E.Otto, C.Grebogi and J.A.Yorke: "Controlling chaos", Physical Review Letters, vol.64,no.11,pp.1198-1199,1990.

[18] K.Pyragas: "Continuous control of chaos by self-controlling feedback ", Physics Letters, A,vol.170,pp.421-428, 1992.