Evolutionary Modeling of Ordinary Differential Equations for Dynamic Systems

Hongqing Cao[1,2], Lishan Kang[1,3] and Yuping Chen[1,3]

[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, P. R . China
[2] Email: chq@rjgc.whu.edu.cn [3] Email: kang@whu.edu.cn

Abstract

This paper presents a new idea of modeling one-dimensional dynamic systems by higher-order ordinary differential equation (HODE) models in stead of by the ARMA models used in the traditional time series analysis. Accordingly, based on the idea of hybrid evolutionary modeling, we modify the HEMA algorithm in (Cao et al., 1998a; Cao et al., 1998b) to approach the modeling problem of HODEs for dynamic systems. Two applications of time series are used to demonstrate its effectiveness and advantages.

1 INTRODUCTION

Many complex systems and nonlinear phenomena which dynamically change with time exist in real world. It has long been a great issue for people to build dynamic mathematical models for those systems based upon the observed data (i.e., time series) so as to provide basis for system analysis, design and prediction. The three kinds of models people have used most in traditional time series analysis are Autoregressive (AR) Models, Moving Average (MA) Models and ARMA Models (Xiang et al., 1988; Gan, 1991). Some variations based on those three models are also studied, such as Autoregressive Integrated Moving Average (ARIMA) Models (Ozaki, 1977; Box & Jenkins, 1976), Bilinear Models (Granger & Andersen, 1978; Gabr & Rao, 1981), Threshold Autoregressive (TAR) Models (Tong, 1978) and TARMA Models (Wang et al., 1984). The traditional methods for solving the modeling problem of dynamic systems are to choose a model structure for the system initially, including the type and the order of the model, determine the parameters contained in the model subsequently, and test the validity of the model finally (Brockwell & Davis, 1986; Anderson, 1975). However, it is usually difficult for people to choose a suitable model structure without sufficient domain details and human expertise, and the determination of parameters also requires the modeler have rich mathematical knowledge and professional skills. Additionally, the traditional ARMA Models are linear models which can only reflect a very limited range of dynamic processes. Considering that the ARMA Models are linear difference equations in essence, we present a new idea of modeling one-dimensional dynamic systems by higher-order ordinary differential equation (HODE) models in stead of by the ARMA Models.

We have ever proposed a two-level hybrid evolutionary modeling algorithm called HEMA to approach the modeling problem of systems of ordinary differential equations (ODEs) whose main idea is to embed a genetic algorithm (GA) (Holland, 1975) in genetic programming (GP) (Cramer, 1985; Koza, 1992; Koza, 1994; Banzhaf et al., 1997) where GP is employed to optimize the structure of a model (upper level), while a GA is employed to optimize the parameters of a model (lower level) (Cao et al., 1998a; Cao et al., 1998b). Some numerical experiments were done to test HEMA's effectiveness. In this paper, based on the idea of HEMA, we make some modifications to this algorithm and adjust it to the modeling problem of HODEs by converting a HODE into a system of ODEs. The modified algorithm is applied to two practical examples of time series to testify its effectiveness.

This paper is organized as follows. The problem is defined in Section 2. In Section 3, we present the structure of HEMA for HODEs and give some detailed descriptions. In Section 4, two typical examples are used to test the effectiveness of the algorithm. Finally, conclusions are summarized in Section 5.

2 PROBLEM STATEMENT

Suppose that a series of observed data collected from a one-dimensional system X(t) at successive m time steps can be written as

$$\mathrm{X} = \begin{pmatrix} x(t_0) \\ x(t_1) \\ \vdots \\ x(t_m) \end{pmatrix} \quad (1)$$

where $t_i = t_0 + i*\Delta t$ ($i = 0, 1, 2, \ldots, m$), t_0 denotes the starting time, Δt denotes the interval between two observations. The modeling problem of HODEs for the dynamic system X(t)

is to find a model of nth-order ordinary differential equation (ODE)

$$x^{*(n)}(t) = f(t, x^*(t), x^{*\prime}(t), x^{*\prime\prime}(t), \ldots, x^{*(n-1)}(t)) \quad (2)$$

to describe the system such that $\| X^* - X \|$ is minimized and reliable predictions of the system can also be given based on the model. Here $\| X^* - X \|$ is defined as

$$\| X^* - X \| = \sqrt{\sum_{i=0}^{m} \left[x^*(t_i) - x(t_i)\right]^2} \quad (3)$$

f is composed of some elementary functions including triangle functions, exponential functions and power functions.

For simplicity, the nth-order ordinary differential equation model with the form of (2) is called ODE(n) model in following sections.

3 HEMA FOR HODES

Base on the idea of two-level evolutionary modeling, the structure of HEMA for HODEs can be described in pseudo code as follows:

```
Procedure HEMA for HODEs;
begin
    input original data X(0);
    preprocess X(0) and get X(1);
    input the order of ODE models;
    compute the conversion matrix Y of X(1);
    initialize the ODE model population P(0);
    evaluate P(0);
    s :=0;
    repeat
        simplify P(s);
        normalize P(s);
        for i: = 1 to popsize do
        begin
            if(structure(pi) ∉ optimized)
            begin
                check out all the constants contained in pi;
                initialize the parameter population Q(0);
                evaluate Q(0);
                t := 0;
                repeat
                    select Q(t+1) from Q(t) according to fitness and selection strategy;
                    recombine Q(t+1) by using genetic operators(crossover, mutation and reproduction);
                    evaluate Q(t+1);
                    t := t+1;
                until termination criterion II;
                replace all the parameters in pi with the best individual in Q(t);
            end
        end
        select P(s+1) from P(s) according to fitness and selection strategy;
        recombine P(s+1) by using genetic operators(crossover, mutation and reproduction);
        evaluate P(s+1);
        s := s+1;
    until termination criterion I;
    make system prediction based on the best individual in P(s);
end
```

To avoid repeated descriptions, we will give some explanations about those details which differ from the HEMA for system of ODEs in following subsections. Interested readers are strongly recommended to refer to (Cao et al., 1998a; Cao et al., 1998b) to get more details.

3.1 DATA PREPROCESSING

As for the original data, we apply low-pass filtering to eliminate noise at high frequencies by means of the discrete Fourier transform.

3.2 CONVERSION OF HODE

Suppose that a HODE has the form of

$$x^{(n)}(t) = f(t, x(t), x'(t), x''(t), \ldots, x^{(n-1)}(t)) \quad (4)$$

In order to calculate the approximate values of x in a time series from t_0 to t_m by means of numerical integration of system of ODEs, thus to evaluate the fitness of the model subsequently, we first convert it into a set of n coupled first-order ordinary differential equations having the form of

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \vdots \\ y_{n-1}' = y_n \\ y_n' = f(t, y_1, y_2, \ldots, y_n) \end{cases} \quad (5)$$

by the replacement of variables

$$y_1 = x, \ \ y_2 = x', \ \ y_3 = x'', \ \cdots, \ \ y_n = x^{(n-1)} \quad (6)$$

and compute the conversion matrix Y of $X^{(1)}$:

$$Y = \begin{pmatrix} y_1(t_0), & y_2(t_0), & \cdots & , y_n(t_0) \\ y_1(t_1), & y_2(t_1), & \cdots & , y_n(t_1) \\ \vdots & \vdots & & \vdots \\ y_1(t_m), & y_2(t_m), & \cdots & , y_n(t_m) \end{pmatrix} \quad (7)$$

If we denote $Y_i = (y_i(t_0), y_i(t_1), \cdots, y_i(t_m))^T$ then $Y_1 = X^{(1)}$, and Y_i for i=2, 3, $\cdots$, n, which are the (i-1)st-order derivatives of x in a time series from t_0 to t_m respectively, can be figured out approximately by means of numerical differentiation. For example, for $n \leqslant 4$, we can use the following formulae of order h^2 error:

• forward difference formula:

$$
\begin{aligned}
x_i' &= \frac{-x_{i+2} + 4x_{i+1} - 3x_i}{2h} \\
x_i'' &= \frac{-x_{i+3} + 4x_{i+2} - 5x_{i+1} - 2x_i}{h^2} \\
x_i''' &= \frac{-3x_{i+4} + 14x_{i+3} - 24x_{i+2} + 18x_{i+1} - 5x_i}{2h^3}
\end{aligned}
\tag{8}
$$

• central difference formula:

$$
\begin{aligned}
x_i' &= \frac{x_{i+1} - x_{i-1}}{2h} \\
x_i'' &= \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} \\
x_i''' &= \frac{x_{i+2} - 2x_{i+1} + 2x_{i-1} - x_{i-2}}{2h^3}
\end{aligned}
\tag{9}
$$

• backward difference formula:

$$
\begin{aligned}
x_i' &= \frac{3x_i - 4x_{i-1} + x_{i-2}}{2h} \\
x_i'' &= \frac{2x_i - 5x_{i-1} + 4x_{i-2} - x_{i-3}}{h^2} \\
x_i''' &= \frac{5x_i - 18x_{i-1} + 24x_{i-2} - 14x_{i-3} + 3x_{i-4}}{2h^3}
\end{aligned}
\tag{10}
$$

3.3 ENCODING OF THE MODEL POPULATION

Once a HODE is converted into a system of ODEs, we notice that the only difference between two HODE models is the nth equation $y_n' = f(t, y_1, y_2, \ldots, y_n)$, namely the right-hand function of HODE. When initializing the model population, the THEMA generates *popsize* such individuals randomly and each individual is represented as a binary tree. For example, given a fourth-order ODE $x^{(4)} = 3x''' + \sin(x'') - tx' + xe^t$, its corresponding equation $y_4' = 3y_4 + \sin(y_3) - ty_2 + y_1e^t$ can be represented as a binary tree shown in Fig. 1. Besides this, the maximum depth of per tree is restricted by a constant D and the complexity of a model is measured by the number of nodes contained in each tree.

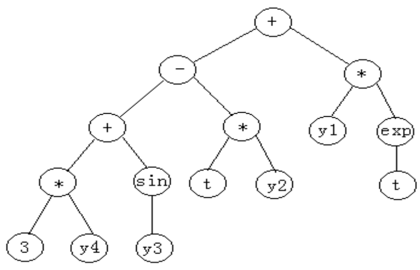

Fig. 1. An example of the representation of a HODE model

3.4 FITNESS EVALUATION OF THE MODEL POPULATION

Suppose that the corresponding system of ODEs of an arbitrary individual p_i in the model population has the general form of (5), the fitness of p_i can be calculated as follows:

```
Procedure  cal_fitness;
begin
    let X* and ΔX be both (m+1)-dimensional column vectors, Y* be a (m+1)×n empty matrix,
    assign the first row of Y to that of Y*;
    for i := 2 to m+1 do
    begin
        integrate the system (5) for a step with some numerical methods by taking the (i-1)st row of Y as the initial conditions;
        assign the solution to the ith row of Y*;
    end
    X*:=Y1*;                 { Y1* denotes the vector composed of the first column of Y*}
    ΔX := X(1) -X*;
    fitness(pi) := || ΔX ||;{The notation "|| ||" represents the norm of a vector }
end
```

Obviously, here the lower the fitness is, the better the individual is. During the fitness evaluation, we use the fourth-order Runge-Kutta method with fixed stepsize 0.01 to integrate the system and build ODE(1) models, ODE(2) models, ODE(3) models and ODE(4) models for two examples of time series respectively.

3.5 SIMPLIFICATION AND NORMALIZATION OF MODELS

The simplification of models is that of simplifying the tree structures of each individual in the model population by replacing some subtrees which consist of some arithmetic operations between constants with calculable values. This operation is performed on all individuals in every generation which will affect the number of parameters to optimize but not change the fitness of individuals.

The normalization of models is that of adjusting the structure of such subtree in the model whose root is "+" (plus) or "*" (multiplication) and whose left branch or right branch is a constant to ensure that the constant always lies on the right of "+" or "*" in the S-expression of the model. This operation is helpful to distinguish the model structures correctly so that "*a*+*x*" and "*x*+*a*" or "*a***x*" and "*x***a*" will not be regarded as different structures to do the optimization process redundantly.

3.6 SYSTEM PREDICTION

Once the best evolved model is obtained in one run, we then take the last row of Y as the initial conditions, integrate the corresponding system of ODEs for several steps by using the fourth-order Range-Kutta method with stepsize 0.01 and get the predicted series of Y^*. The first column of Y^* is just the predicted series of the dynamic system based on the model.

4 COMPUTATIONAL EXPERIMENTS

4.1 PARAMETER SETTINGS AND MEASURES

To examine the effectiveness of the HEMA for HODEs, we apply it to two practical examples of time series and build ODE(1) models, ODE(2) models, ODE(3) models and ODE(4) models for each example respectively. Ten runs are conducted independently for ODE models with different order. All the experiments are performed on Pentium II (266Mhz) using Visual C++ Compilers. The parameter settings are as follows:

- For the evolutionary modeling process:

We use the function set F = {+, −, *, /, ∧, *sin*, *cos*, *exp*, *ln*} where *x^n* symbolizes x^n ($0<n<5$), the terminal set T = {y_1, ⋯, y_n, t, c} where n is the order of ODE and c is a random constant, a population size of 50, a maximum tree depth of 4 and a maximum of 50 generations per run.

- For the parameter optimization process:

We use a population size of 20, a 60% crossover rate, a 30% mutation rate and a 10% reproduction rate, and the termination criterion of that the fitness value of the best individual has remained unchanged for 3 generations.

In addition, the following measures are applied to compare the modeling results of HODEs with different order for two examples:

- fitting error (FE) and prediction error (PE) which are defined as

$$\mathrm{FE} = \sqrt{\sum_{i=1}^{m}(\hat{x}_i - x_i)^2} \qquad \mathrm{PE} = \sqrt{\sum_{i=m+1}^{m+n}(\hat{x}_i - x_i)^2} \qquad (11)$$

where x_i denotes the observed value, $\hat{x}_i$ denotes the fitting value and the predicted value of the HODE model for FE and PE respectively, m is the number of observed data to build a model and n is the number of time steps required to predict. To be specific, for Example I, m=110, n=4; for Example II, m=220, n=6.

- average fitting error (AFE) and average prediction error (APE) which are the mean values of FE and PE of the HODE models obtained in ten runs respectively.
- average number of nodes (AN_{nodes}) which is the mean value of the number of nodes of the HODE models obtained in ten runs.
- number of success (N_{succ}), namely the number of runs in which the best evolved model can give reasonable predictions. If the best evolved model in one run can not make system predictions at all or its prediction error are enormously large, we declare it a failure; or else a success.

4.2 EXAMPLE I: LEOPARD CAT QUANTITY OF CANADA

The experimental data are cited from (Xiang et al., 1988) which are about the quantity of the leopard cats of Canada from 1831 to 1944. Many statisticians have ever shown great interests to the dynamic data. We now take the observed data of the first 110 years as history data to build HODE models and predict the values of the last four years.

As the amplitude of original data leaps greatly, we first compute their values of common logarithm and take the transformed data as input to build HODE models.

The statistical results of ten runs for Example I are shown in Table 1.

Table 1. The Statistical Results of Modeling by ODE Models for Example I (10 runs)

Model	ODE(1)	ODE(2)	ODE(3)	ODE(4)
AFE	3.561151	1.243644	0.124922	0.507806
APE	0.852183	0.522207	0.264809	0.574626
AN_{nodes}	9.2	8.4	8.4	8.8
Mean Time(sec.)	2280	3064	2325	1397
$\mathbf{N_{succ}}$	10	10	10	10

Obviously, the ODE(3) model is most desirable to describe the system as both its AFE and APE are much smaller than those of other three models. We show the best ODE(3) model in ten runs in Table 2 and illustrate its curves of fitting and prediction in Fig. 2.

Table 2. The Best ODE(3) Model in Ten Runs for Example I

Evolutionary Solution	$\begin{cases} dy_1/dt = y_2 \\ dy_2/dt = y_3 \\ dy_3/dt = y_3/(y_2-(-0.143929)) \end{cases}$			
Equivalent ODE	$x''' = x''/(x'+0.143929)$			
FE	0.123900			
PE	0.264711			
Observed Value	3.000000	3.201397	3.424392	3.530968
Predicted Value	2.999852	3.219896	3.482772	3.788497

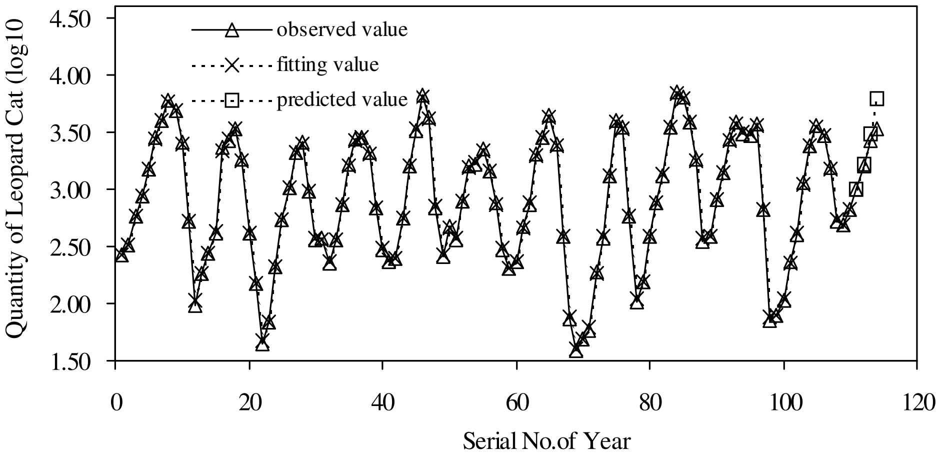

Fig. 2. The fitting and prediction curves of the best ODE(3) models for Example I

We can see that both the fitting values and the predicted values are pretty good and, more importantly, the model is quite simple in structure. In (Xiang et al., 1988), a TAR(2, 2; 8, 3) model has been built for the dynamic data, but its structure is rather complicated.

4.3 EXAMPLE II: CHEMICAL REACTION TEMPERATURE

The experimental data are cited from (Box & Jenkins, 1976) which are about the centigrade temperature of some chemical reaction process recorded every other minute. In this experiment, we take the recorded data of the first 220 minutes as history data to build HODE models and predict the values of the last 6 minutes.

The statistical results of ten runs for Example II are shown in Table 3.

Table 3. The Statistical Results of Modeling by ODE Models for Example II (10 runs)

Model	ODE(1)	ODE(2)	ODE(3)	ODE(4)
AFE	3.272348	0.938791	0.083298	0.387948
APE	3.773765	1.679618	3.648096	2.104190
AN_{nodes}	7	7	8.4	6.6
Mean Time(sec.)	4303	5299	4432	3488
N_{succ}	10	10	10	7

From the results we see that of the four models, the ODE(1) model is the worst as it has the largest AFE and APE; the ODE(3) model has a minimal AFE but its APE is large; for the ODE(4) model, its AFE and APE seem good, but it has only a 70% rate of success. As a tradeoff of fitting and prediction, we think that the ODE(2) model is superior to other models in describing this time series whose order is identical to the AR(2) model built in (Gan, 1991)

$$x_t = 1.806681x_{t-1} - 0.80668163x_{t-2} + a_t(\sigma_a^2 = 0.02774) \quad (12)$$

The results of the best ODE(2) model in ten runs are shown in Table 4 and its fitting and prediction curves are illustrated in Fig. 3.

Table 4. The Best ODE(2) Model in Ten Runs for Example II

Evolutionary Solution	$\begin{cases} dy_1/dt = y_2 \\ dy_2/dt = y_2/\cos(y_2 * 1.219242) \end{cases}$					
Equivalent ODE	$x'' = x'/\cos(1.219242\,x)$					
FE	0.943623					
PE	1.566485					
Observed Value	20.2	19.7	19.3	19.1	19.0	18.8
Predicted Value	20.250847	19.711464	19.179573	18.653486	18.121416	17.589474

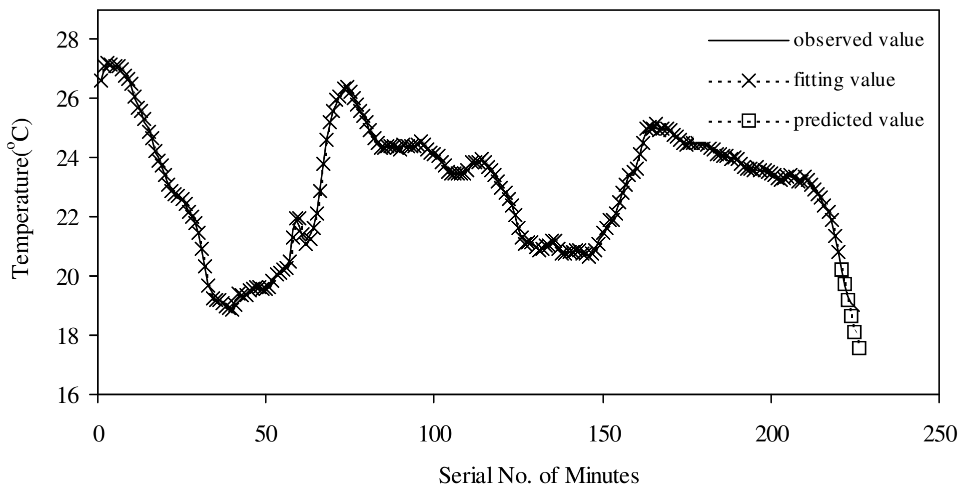

Fig. 3. The fitting and prediction curves of the best ODE(2) model for Example II

It is surprising to see that the fitting values of the model can coincide with the observed data so well and its predicted values are also good. Moreover, the model is a complex nonlinear differential equation which contains cosine function in the expression. In fact, by running the HEMA for HODEs, the computer can search out many such complex models whose structures are usually unimaginable to human minds.

5 CONCLUSIONS

This paper presents a new idea of modeling one-dimensional dynamic systems by higher-order ordinary differential equation (HODE) models in stead of by the ARMA Models used in the traditional time series analysis. Accordingly, based on the idea of two-level hybrid evolutionary modeling, we modify the HEMA algorithm in (Cao et al., 1998a; Cao et al., 1998b) to approach the modeling problem of HODEs for dynamic systems. This modified algorithm has some advantages compared with the traditional modeling methods used for time series analysis:

1)It has broken through the limitation of linear models in traditional ARMA Models, and is capable of building complex non-linear HODE models for one-dimensional dynamic systems.
2)The optimization of the model structure and the optimization of the parameters of the model could be done simultaneously by using the two-level processes in the HEMA for HODEs.
3)It depends very little on domain details and human expertise. The whole modeling process is carried on automatically. By running the algorithm, the computer can even find some excellent HODE models whose structures are unimaginable to human minds.

Acknowledgements

The authors would like to thank the anonymous referees for their comments on the paper. This work was supported in part by National Science Foundation of China (No. 69635030) and National 863 High Technology Project of China.

References

Anderson (1975), *Time Series Analysis and Forecasting: the Box-Jenkins Approach*. London: Butterworths.W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone (1997), *Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann.

G. E. P. Box and G. M. Jenkins (1976), *Time Series Analysis, Forecasting and Control*. San Francisco: Holden Day.

P. J. Brockwell and R. A. Davis (1986), *Time Series: Theory and Methods*. New York: Spring Verley.

H. Q. Cao, L. S. Kang, Z. Michalewicz and Y. P. Chen (1998a), "A Hybrid Evolutionary Modeling Algorithm for System of Ordinary Differential Equations," *Neural, Parallel & Scientific Computations*. Vol.6, No.2, pp. 171-188, June, 1998, Dynamic Publishers, Atlanta.

H. Q. Cao, L. S. Kang, Z. Michalewicz and Y. P. Chen (1998b), "A Two-level Evolutionary Algorithm for Modeling System of Ordinary Differential Equations," *In Genetic Programming 1998: Proceedings of the Third Annual Conference*, David E., Iba, Hitoshi, and Riolo, Rick L. (editors). pp.17-22. University of Wisconsin, Madison, Wisconsin. San Francisco, CA, July 22-25, 1998: Morgan Kaufmann.

N. L. Cramer (1985), "A representation for the adaptive generation of simple sequential programs," In *Proceedings* *of an International Conference on Genetic Algorithms and Their Applications*, Grefenstette, J. J. (editor). pp.183-187, Carnegie-Mellon University, Pittsburgh, PA, USA, 24-26, July, 1985.

M. M. Gabr and T. S. Rao (1981), “The estimation and prediction of subset bilinear time series models with applications,” *Journal of Time Series Analysis*, no. 2, pp. 155-171.

R. C. Gan (1991), *The Statistical Analysis of Dynamic Data*. Beijing: Beijing University of Science and Technology Press (in Chinese).

C. W. Granger and A. P. Andersen (1978), *An Introduction to Bilinear Time Series Models*. Vandenhoeck and Ruprecht: Gottingen.

J. H. Holland (1975), *Adaptation in Natural and Artificial System*. Ann Arbor, MI: Univ. Of Michigan Press.

J. R. Koza (1992), *Genetic Programming：on the Programming of Computers by Means of Natural Selection*. Cambridge, MA：MIT Press.

J. R. Koza (1994), *Genetic Programming II：Automatic Discovery of Reusable Programs*. Cambridge, MA：MIT Press.

T. Ozaki (1977), “On the order determination of ARIMA models,” *Application Statistics*, vol. 26, no. 3, pp. 290-301.

H. Tong (1978), “On a threshold model,” *Pattern Recognition and Signal Processing*, NATO ASI Series E: Applied Science, no. 29.

W. Y. Wang, J. G. Du and J. T. Xiang (1984), “Threshold autoregressive moving average models: TARMA,” *Computational Mathematics*, no. 4 (in Chinese).

J. T. Xiang, J. G. Du and J. E. Shi (1988), *Dynamic Data Processing: Time Series Analysis*. Beijing: Meteorology Press (in Chinese).