Analysis on Evolution Trend of the Software Failure Data Based on Adaptive Volterra Model

Ting Lei
College of Science, Beijing Forestry University, Beijing 100083, China

HuiYang Xie*
College of Science, Beijing Forestry University, Beijing 100083, China
*Corresponding author(E-mail: yangrjw@126.com)

Abstract
This paper proposes the adaptive prediction model of software failure data analysis based on Volterra Series combined the theory of phase space reconstruction. The model makes full use of the advantages of small computational complexity, fast convergence speed and strong adaptive ability of Volterra series, to well reflect the dynamic characteristics of the software failure system. Numerical simulation results of practical failure data series show that Volterra model improves the prediction precision of traditional chaotic local model, also possesses a strong the anti-noise performance and can hold an optimistic prediction effect even if the data series have been polluted by the white noise.

Key words: Software, Failure Data, Chaos, Volterra Series, Prediction

1. INTRODUCTION
In recent years, reliability has become a research problem receiving increasing attention in the field of software research. Software failure can be caused by many factors, like software scale, development environment, software testing environment and development personnel’s technology and skills. Different factors lead to different software failure behaviors. The characteristics of the software itself determine its very complex failure behaviors. This kind of complexity easily results in the difference between the estimation results by the traditional model and the actual situation. A large number of studies show that software failure has the obvious feature of chaotic behavior rather than the traditional assumption of obeying non-homogeneous Poisson process and other random processes. Therefore, the independence hypothesis-based Markov process and other software reliability analysis models cannot effectively describe the dynamic behavior of software failure data. The chaos theory emphasizing system integrity provides a good idea for this kind of problems.

In the prediction field of chaotic time series, the local prediction model based on the phase space reconstruction technology has attracted more and more attention because of its good approximation function. The nonlinear Volterra adaptive prediction filter is very suitable for dealing with the real-time related problems due to its good mathematical properties, small computational complexity, good approximation effect, fast convergence speed and no need of the prior knowledge about the input signal. The paper proposes a prediction model of software failure data based on Volterra series. The model can make full use of the good mathematical properties of Volterra series and more comprehensively describe the dynamic properties of software failure data based on the good nonlinear approximation performance of the traditional chaotic local model. Thus, it can accurately predict the evolution trend of failure data in the future.

2. PHASE SPACE RECONSTRUCTION OF SOFTWARE FAILURE DATA
It is assumed that the software failure rate \( \kappa \) is proportional to the number of software defects \( x(t) \) contained in the software system:

\[
\kappa = \frac{dx(t)}{dt} = \kappa x(t)
\]

If the real function of the software failure system is expressed by \( F \), the time series of the software failure data \( \{x(k)\}_{k=1,2,...,n} \) is the sampling value of the system function \( f \) at the time \( t = 1,2,3,\cdots \). Since the relationship between the operation rule and the influential factors is not entirely clear, it is difficult to fully understand the physical mechanism of the actual software failure data, let alone precisely constructing the function \( f \) from the obtained software failure data. To solve this problem, the phase space reconstruction technology in the
The Volterra series model has many unique advantages compared with other nonlinear models. It has two kinds of forms, namely, time domain and frequency domain. The kernel functions in the time domain and frequency domain are the high order pulse response function and the generalized frequency response function, respectively. They both have clear physical meanings (Peng, Lang, 2007; Tomlinson, Manson, 1996).

### 3. PREDICTION FILTER BASED ON VOLTERA SERIES

#### 3.1. Volterra Series Theory

The Volterra series is an important mathematical tool for the study of nonlinear systems. It is the generalization of the convolution operation in the linear system theory in the analysis of nonlinear systems, which provides not only a new set of nonlinear theory but also the powerful tool for solving the nonlinear actual problem. The Volterra series model has many unique advantages compared with other nonlinear models. It has two kinds of forms, namely, time domain and frequency domain. The kernel functions in the time domain and frequency domain are the high order pulse response function and the generalized frequency response function, respectively. They both have clear physical meanings (Peng, Lang, 2007; Tomlinson, Manson, 1996).
Considering a linear system $y(t) = \int_{-\infty}^{\infty} b(t-\tau)x(\tau)d\tau$, where $x(t)$ is the input of the system and $y(t)$ is the input of the system, and $b(t)$ is the impulse response function. Obviously, the system can be determined by $b(t)$ uniquely. The system is transformed through Fourier transform to get the frequency domain expression of the system $Y(\omega) = B(\omega)X(\omega)$, where $Y(\omega)$, $B(\omega)$ and $X(\omega)$ are the Fourier transforms of $y(t)$, $b(t)$ and $x(t)$, respectively. All information of the system is contained in the response function $B(\omega)$ or $b(t)$.

The above system is correct for the analysis of linear systems, but it cannot be used for the analysis of nonlinear systems. The frequency response function can greatly simplify the analysis of the linear system, but unfortunately it cannot be directly applied in the analysis and research of the nonlinear vibration system. It cannot explain the frequency response phenomenon which is special for some nonlinear vibration systems, such as the resonance phenomenon, the high-order harmonic and subharmonic generation and the cross modulation phenomenon between frequencies.

Fortunately, like the linear system, for any continuous time-invariant nonlinear dynamic system, if both the input and output are the analytic functions, and if the input signal $x(t)$ adopted by the system is limited under the zero initial condition, i.e.:

$$\sqrt{\int_{-\infty}^{\infty} x^2(t)dt} < \infty$$

the nonlinear system can be represented by Volterra series.

Although Volterra series is an infinite order number, but studies have shown that a large class of nonlinear systems can be represented by Volterra series of finite order. With the relatively simple physical implementation, it is suitable for the modeling description of nonlinear processes. At the same time, under the condition of limited input signal energy, the causal continuous time invariant nonlinear system can be approximated at any degree by Volterra series. This excellent mathematical property lays a theoretical foundation for the application of Volterra series to the prediction of nonlinear systems.

$$y(t) = y_0(t) + y_1(t) + y_2(t) + \cdots + y_n(t) \quad (2)$$

where

$$y_i(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} b_i(\tau_1, \tau_2, \cdots, \tau_i) x(t-\tau_1)x(t-\tau_2)\cdots x(t-\tau_i)d\tau_1d\tau_2\cdots d\tau_i, \quad i = 1, 2, \cdots$$

In the expression, $b_k(j_1, j_2, \cdots, j_k)$ is the generalization of the linear impulse response function, representing the $k$-order Volterra kernel function. The kernel function contains all the information of the system. The Volterra kernel function is symmetric, namely,

$b_2(j_1, j_2) = b_2(j_2, j_1), \cdots, b_k(j_1, j_2, \cdots, j_k) = b_k(j_{i_1}, j_{i_2}, \cdots, j_{i_k})$

where $j_{i_1}, j_{i_2}, \cdots, j_{i_k}$ is a permutation of $j_1, j_2, \cdots, j_k$.

For the convergent Volterra series, the truncated Volterra series are used to approximately represent the relationship between the input and output of the nonlinear systems. That is:

$$y(t) = y_0(t) + \sum_{i=1}^{n} y_i(t)$$

where $n$ refers to the number of truncated orders.

3.2. Adaptive Evolution Model of the Time Series of the Software Failure Data

The observed software failure data sequence is assumed to be $\{x(t)\}(t = 1, 2, \cdots, n)$. The evolution trend of the software failure can be understood according to the data sequence.

The phase space reconstruction is carried out on the sequence first. According to Taken theorem, the appropriate delay time $\tau$ and embedding dimension $m$ of the phase space are selected. Then, $N = n-(m-1)\tau$ phase points of the sequence in $m$-dimension reconstructed phase space can be obtained:

$$Y(t) = [x(t), x(t-\tau), \cdots, x(t-(m-1)\tau)] \quad t = t_1, t_2, \cdots, t_N = n$$

where

$$Y(n) = [x(n), x(n-\tau), \cdots, x(n-(m-1)\tau)]$$

$Y(n+1)$ is called the prediction center, and other phase points are called the neighbor points.
The key to the prediction of the failure data sequence is to find out the mapping \( F : R^m \rightarrow R^m \) in Takens theorem and make it meet the condition \( Y(t+1) = F[Y(t)] \). Once the mapping \( F \) is finalized, the point \( Y(n+1) \) in the reconstructed phase space can be predicted by making use of the neighbor points, so as to get the predicting value of the time series \( x(n+1) \). This is because it is the first component of the phase point \( Y(n+1) \).

By this method, \( x(n+1), x(n+2), \cdots \) can be predicted by repeating the above steps and making iterations.

The chaotic polynomial local model is commonly used to construct the nonlinear mapping \( F \):

\[
Y(t+1) = F[Y(t)] = a_0 + a_1Y(t) + a_2Y^2(t) + \cdots + a_nY^n(t)
\]

However, the model is prone to be affected by the computational error in the case of high order and large amount of data, and the prediction results are lack of stability.

Since the Volterra series has a good approximation effect on nonlinear systems, the paper will design the appropriate Volterra filter to establish the adaptive prediction model of the software failure data sequence evolution based on Volterra series and phase space reconstruction theory.

The purpose of our prediction is to obtain the predicted value of the first component of the phase point \( Y(n+1) \). Corresponding to Equation (1), the following mapping \( f : R^m \rightarrow R \) can be obtained, meeting:

\[
x(t+1) = f[x(t), x(t-(1)), x(t-(2)), \cdots, x(t-(m-1))] \quad i = 1, 2, \cdots, M
\]

The function \( f \) on the right is considered to be carried out according to the Volterra series. Because the software failure data sequence is discrete, and Series equation (2) is expressed in the form of continuous Volterra series form, which cannot be directly used for the discrete data analysis, it needs to discretize Equation (2). The discrete Volterra series is used to express the function \( f \). The following can be obtained:

\[
x(t+1) = f[x(t), x(t-(1)), x(t-(2)), \cdots, x(t-(m-1))] = b_0 + \sum_{j_1=0}^{\infty} b_1(j_1)x(t-j_1) \\
+ \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} b_2(j_1, j_2)x(t-j_1)x(t-j_2) + \cdots + \\
\sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \sum_{j_3=0}^{\infty} b_3(j_1, j_2, j_3)x(t-j_1)x(t-j_2)x(t-j_3) + \\
\cdots + \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \sum_{j_3=0}^{\infty} \sum_{j_4=0}^{\infty} b_4(j_1, j_2, j_3, j_4)x(t-j_1)x(t-j_2)x(t-j_3)x(t-j_4)
\]

It is the Volterra filter used for predicting the software failure data. In the formula, \( b_k(j_1, j_2, \cdots, j_k) \) represents the \( k \)-order Volterra kernel function. Obviously, with the increase of the series \( k \), the input parameters and algorithm complexity of the filter signal will increase according to the exponential rate. For this reason, the second-order truncation summation is carried out on the above filter to get the Volterra prediction filter form of the time series.

\[
x(t+1) = f[x(t), x(t-(1)), x(t-(2)), \cdots, x(t-(m-1))] = b_0 + \sum_{j_1=0}^{n_1-1} b_1(j_1)x(t-j_1) + \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} b_2(j_1, j_2)x(t-j_1)x(t-j_2)
\]

According to the Takens theorem, the dimension of the original nonlinear system is assumed to be \( D \). To make the reconstructed phase space of a chaotic time series describe the information of the sequence complete, the embedding dimension \( m \) of the reconstructed phase space needs to meet \( m > 2D+1 \). Thus, in Equation (6), \( n_1 = n_2 = m \) can be taken. Then, the adaptive prediction model of the software failure data based on Volterra series can eventually be expressed as:
$$x(t + 1) = f \left[ x(t), x(t - \tau), x(t - 2\tau), \ldots, x(t - (m - 1)\tau) \right]$$
$$= b_0 + \sum_{j=0}^{m-1} b_1(j)x(t-j_1\tau) + \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{m-1} b_2(j_1, j_2)x(t-j_1\tau) \cdot x(t-j_2\tau)$$  \hspace{1cm} (7)

The model belongs to a two-order adaptive nonlinear predictive FIR filter. Given:

$$B(t) = \left[ b_0, b_1(0), b_1(1), \ldots, b_1(m - 1), b_2(0, 0), b_2(0, 1), \ldots, b_2(m - 1, m - 1) \right]$$
$$X(t) = \left[ 1, x(t), x(t - \tau), \ldots, x(t - (m - 1)\tau), x^2(t), x(t)x(t - \tau), \ldots, x^2(t - (m - 1)\tau) \right]$$

where \( B(t) \) represents the coefficient vector of the filter; \( X(t) \) represents the input vector of the filter. For the continuous input and output nonlinear system, the predictive filter can be expressed in the following matrix form:

$$x(t + 1) = X^T(t) \cdot B(t)$$  \hspace{1cm} (8)

### 3.3. Determination of Volterra Filter Coefficient

It can be seen from Equation (7) that as long as the coefficient of the filter is solved, the adaptive prediction model of the time series of the software failure data can be established.

The Volterra filter coefficient is described by the time domain orthogonal algorithm as follows:

The estimated error of the filter is defined:

$$e(t) = x(t) - \hat{x}(t)$$

where \( \hat{x}(t) \) is the predictive value.

The output response of the filter:

$$x(t + 1) = X^T(t) \cdot B(t)$$

The adaptive adjustment process of the filter coefficient:

$$B(t + 1) = B(t) + \rho \cdot e(t)X(t)$$

$$X^T(t + 1)X(t + 1)$$

where \( \rho \) represents the control parameter, used to adjust the convergence property of the filter.

Compared with the minimum mean square error algorithm and other adaptive algorithms which apply the space average, the TDO algorithm is based on the time average of the error square. Thus, it can select the convergence control parameter in a wider scope, which is more suitable for the nonlinear models with high time complexity. The TDO algorithm can more accurately depict the dynamic evolution of the software failure data sequence, so as to adaptively train and adjust the filter coefficients and get the best prediction performance.

### 4. Numerical Simulation

Now, the actual software failure data are used to verify the performance of the Volterra prediction model by numerical simulation.

The data of the array simulation are from the Handbook system which collects the software failure data. The data set CSR1 for experimental analysis has recorded the failure data of 397 kinds of software. The time series formed by these data is \( \{x(t), t = 1, 2, \ldots, 397\} \). 380 data will be used to train the model and establish the Volterra prediction filter. The remaining 17 data will be used to test the prediction effect of the filter.

### 4.1. Chaotic Behavior Analysis of the Failure Data

The power spectrum is applied to judge whether the software failure data has the chaotic characteristics from the point of qualitative analysis. Solving the power spectrum from Fourier analysis of the time series has recorded the failure data of 397 kinds of software. The time series formed by these data is \( \{x(t), t = 1, 2, \ldots, 397\} \). 380 data will be used to train the model and establish the Volterra prediction filter. The remaining 17 data will be used to test the prediction effect of the filter.
Next, the software failure data sequence is used for the phase space reconstruction. First of all, the phase space reconstruction is carried out on the sequence. According to the Takens theorem, two important parameters, namely, the embedding dimension \( m \) and the delay time \( \tau \), are required when reconstructing the phase space. \( m \) can be determined by CAO algorithm(Cao, Liangyue, 1997; Yue, Li, 2015), as shown in Fig. 2. Its abscissa is the embedding dimension and its ordinate is the energy functions \( E_1 \) and \( E_2 \). The value of \( m \) which makes \( E_1 \) value smooth is the embedding dimension needing to solve. After calculation, \( m = 5 \). Thus, according to the design scheme of the model, the order number of the Volterra predictive filter also takes 5. The parameter \( \tau \) is determined by the mutual information method. As shown in Fig. 3, the abscissa is the delay time and the ordinate is the mutual information. The value of \( \tau \) which makes the mutual information reaches the minimum value point for the first time is the delay time. After calculation, \( \tau = 6 \). With the values of \( m \) and \( \tau \), the reconstructed phase space of the software failure data can be obtained based on the reconstruction technology of Takens theorem, so as to solve the Lyapunov exponent of the failure data.

Then, the Lyapunov exponent \( \lambda \) of the flow series is calculated.

With two parameters of the embedding dimension \( m \) and the delay time \( \tau \), the Lyapunov exponent of the time series can be calculated by Wolf method(Kantz, Schreiber, 1997; Sprott, 2003). The chaotic degree of a system can be characterized by the largest Lyapunov exponent. When \( \lambda > 0 \), it indicates that it is a chaotic software failure data. The greater \( \lambda \) is, the higher the degree of chaos of the failure data is. As shown in Fig. 4, corresponding to embedding dimension \( m = 5 \), the Lyapunov exponent \( \lambda = 0.0131 \) of the failure data sequence can be derived, which is consistent with the previous judgment that the failure data sequence has the chaotic characteristics.

Next, the Volterra filter is used to predict the failure data.

### 4.2 Time Series Prediction of the Failure Data

In order to compare the prediction results, the chaotic polynomial local model (3) with the order \( n = 1 \) is also applied to predict the software failure data.

In order to evaluate the prediction effect of the models, the following error indices are defined.

Absolute error of each prediction point: \( E_i = |x(t) - \hat{x}(t)| \).

Average Error: \( AE = \frac{1}{n} \sum_{i=1}^{n} E_i \).

Root mean square error: \( SR = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} [x(t) - \hat{x}(t)]^2} \), used to describe the average deviation degree between the predicted value and the actual value.

Mean absolute percentage error: \( SM = \frac{1}{n} \sum_{i=1}^{n} \frac{|x(t) - \hat{x}(t)|}{x(t) + \hat{x}(t)} \), used to describe the evolution trend of the failure data sequence and measure the accuracy of the model.

Regularized mean square error: \( MSN = \frac{\sum_{i=1}^{N} [x(t) - \hat{x}(t)]^2}{\sum_{i=1}^{N} x^2(t)} \), used to describe the ratio between the absolute error and the real value.

The prediction results are shown in Fig. 5. Obviously, the prediction results of Volterra prediction filter and the traditional local model are close to the actual value of the failure data. Table 1 shows the numbers of predicting points with the relative errors of two models falling in a range and the predicted average relative errors of the models. Table 2 records the comparison results of the performance indexes \( SR \), \( SM \) and \( MSN \) of two prediction models.

From these results, compared with the traditional local linear model, the proportion of the predicting points of the Volterra prediction filter with the relative error falling in the small value range is high. From the perspective of relative error, the improved model does improve the prediction performance of the model. It indicates that using Volterra filter can achieve a better effect in predicting the software failure data.

Finally, the anti-noise capacity of the model is discussed. The white noise sequence is added to the above software failure data sequence, in order to simulate the noise in the actual observation sequence. The noise power is from \( 4.87 \times 10^{-5} \) to 4.8. The failure data sequence with the noise is modeled to compare the difference between the errors of the prediction models before and after adding the noise. The results are shown in Fig. 6. In the figure, the transverse axis is the noise power, and the longitudinal axis is the relative errors of different points. It can be
seen that the errors of the prediction models before and after adding the noise increase when the noise power rises. However, compared with the Voltrra prediction filter, the local model has more obvious changes. Its prediction effect is more sensitive to the noise. In the process that the noise power is increased, the error rate of the coefficient is high. When the noise power is increased to 0.0054, the difference between the model coefficients before and after adding the noise has been very obvious. However, for the Voltrra prediction filter, the increasing rate is relatively slow. From this point of view, the Voltrra prediction filter has better anti-noise performance.

Figure 1. Power Spectrum of Failure Data

Figure 2. Determining the Embedding Dimension of the Reconstructed Phase Space

Figure 3. Determining the Delay Time
Figure 4. Calculating the Lyapunov Exponent

Figure 5. Prediction Effect of the Models and Effect Comparison

Figure 6. Relationship of the Noise Power and the Model Coefficient Error

Table 1. Relative Error Ranges of Two Prediction Model

<table>
<thead>
<tr>
<th>Error range</th>
<th>Local Model</th>
<th>Volterra Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0.05</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>[0.05,0.06]</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>[0.06,0.07]</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Local Model</td>
<td>Volterra Model</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>SR</td>
<td>102.687</td>
<td>157.630</td>
</tr>
<tr>
<td>SM</td>
<td>0.02679</td>
<td>0.04673</td>
</tr>
<tr>
<td>MSR</td>
<td>0.0028</td>
<td>0.0067</td>
</tr>
</tbody>
</table>

Table 2. Errors of Two Prediction Models

5. CONCLUSIONS

At present, due to the influence of software scale, development environment, test technology and other factors, the software system has had the characteristics of nonlinear complex systems. As a result, the traditional software failure data prediction model cannot describe the evolution trend of failure data objectively. The paper studies the prediction of failure data. It combines the phase space reconstruction technique of the chaotic theory to propose the adaptive prediction model of the software failure data based on the Volterra series. The model makes full use of the good mathematical properties of small computational complexity, fast convergence speed and strong adaptive ability of Volterra series as well as the advantages of the nonlinear approximation of the local prediction model to accurately reveal the inherent relationship and evolution mechanism of the software failure data system, which overcomes the disadvantages of large computational amount and poor anti-noise performance of the traditional local prediction model when the polynomial has high orders and the sample data is large. The model is proved to be successful through the simulation verification of the actual failure data.

ACKNOWLEDGEMENTS

This work was supported by national natural science foundation of china(Grant No. 61370193).

REFERENCES


Tomlinson G.R., Manson g.(1996) “A Simple Criterion for Establishing an Upper Limit to The Harmonic