Design and simulation of a software robustness testing model

WANG Xiang-gang

Vocational College of DongYing, Shandong 257091, China

Keywords: software robustness; multiple probability analysis; probabilistic robustness analysis

Abstract. The software robustness is affected by many factors, in order to improve the detection accuracy of software robustness, multiple probability analysis of computer control system is introduced, and a software robustness detection model based on probability robustness analysis is put forward. The computer control system of multiple probability analysis is utilized to detect the overall trend of software robustness, and achieve accurate detection of software robustness. The simulation results show that, compared to the traditional detection method, the software robustness detection model based on probability robustness analysis improves the detection accuracy of software robustness, provides some scientific, constructive suggestions for the work of software development, and improves the quality of software.

Introduction

Software robustness is regarded as an important index of the quality of the software, the accuracy of detection results has important significance on the software market decision [1- 2]. Aiming at the problem of software robustness detection, domestic and foreign scholars have carried out extensive research, great achievements have been made [3]. The traditional software robustness detection model mainly include the method of multiple linear regression, time series analysis method and grey model [4-6], however, the traditional single model have the problem of low detection accuracy. In recent years, combination optimization theory has been widely used, but hasn't been applied to software robustness detection.

In order to improve the detection accuracy of software robustness, with computer control system can describe the advantages of overall system variation trend, at the same time, using the probability robustness analysis to describe advantages of system dynamic change, a software robustness detection model based on probability robustness analysis is put forward, through the complementary advantages of these two to fully excavate the information in software robustness original data, and the performance of the combination model is tested through simulation experiments.

Computer control system robustness detection with the introduction of multiple probability analysis

Problem description. Large computer control system with uncertain linear time-varying can be described as:

$$\dot{x} = (A + \Delta A)x + (B + \Delta B)u \quad (6)$$

The name of large computer control system can be described as:

$$\dot{x} = Ax + Bu \quad (7)$$

In the formula, $A \in R^{n\times n}$, $B \in R^{n\times m}$, ΔA and ΔB are used to describe the perturbation of same dimension dynamic, and all belong to an uncertainty set U , i.e.:

$$U = \{(\Delta A, \Delta B) : \|\Delta A\| = \delta a, \|\Delta B\| = \delta b\} \quad (8)$$

Among them, $\|\cdot\|$ is used to describe the spectral norm of matrix. Assuming that the equivalent system after discretization on nominal control system is described by (9):

$$x\left[(k+1)T\right] = \Phi x(xT) + \Gamma u(kT) \quad (10)$$

In the formula, T is used to describe the sampling period, at the same time:

$$\Phi = e^{AT} \quad (11)$$

$$\Gamma = \int_0^T e^{A\tau} d\tau B \quad (12)$$

Assuming that the discrete time state feedback controller designed for the nominal control system is as follows:

$$u(kT) = Kx(kT) \quad (13)$$

On the basis of the controller designed with formula (13) to do the robustness detection for control system.

$$x\left[(k+1)T\right] = \left[e^{(A+\Delta A)T} + \int_0^T e^{(A+\Delta A)\tau} d\tau (B+\Delta B)K\right] x(kT) \quad (14)$$

Robustness detection method based on probability robustness analysis

The introduction of GM (1,1) model. Based on the relevant theories to shape GM (1,1) model, so as to analyze the large computer control system, and according to the obtained reference factors and evaluation results to provide the basis for the robustness detection of large computer control system. $X^{(0)} = \left\{x_t^{(0)}, t = 1, 2, \cdots, n\right\}$ is used to describe the original time series.

A cumulative operation can obtain 1-AGO sequence $X^{(l)} = \left\{x_t^{(l)}, t = 1, 2, \cdots, n\right\}$, $x_t^{(l)} = \sum_{i=1}^{t} x_i^{(0)}$, generating sequence $X^{(l)} = \left\{x_t^{(l)}, t = 1, 2, \cdots, n\right\}$, according to the sequence to derive the following first-order linear differential equations:

$$\frac{dX^{(1)}}{dt} + \alpha X^{(1)} = \beta \quad (7)$$

Among them, α, β are used to describe the developing gray and inner generated control gray respectively. After completing infinitesimal element discretization, the matrix form of differential equation parameter estimation can be described as:

$$\left[\hat{\alpha}, \hat{\beta}\right]^T = (B^T B)^{-1} B^T Y \quad (8)$$

$$B = \begin{bmatrix} -\left(x_1^{(l)} + x_2^{(l)}\right)/2 & 1 \\ -\left(x_2^{(l)} + x_3^{(l)}\right)/2 & 1 \\ \vdots & \vdots \\ -\left(x_{n-1}^{(l)} + x_n^{(l)}\right)/2 & 1 \end{bmatrix}, Y = \begin{bmatrix} x_2^{(0)} \\ x_3^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} \quad (9)$$

After the differential equation is solved, the time response sequence can be obtained:

$$\hat{x}_{l+1}^{(1)} = \left(x_1^{(0)} - \frac{\hat{\beta}}{\hat{\alpha}}\right) \exp\left(-\hat{\alpha} t\right) + \frac{\hat{\beta}}{\hat{\alpha}} \quad (10)$$

The restore operation is applied to acquire the fitting sequence:

$$\hat{Y}(t+1) = \hat{x}_{t+1}^{(0)} = \hat{x}_{l+1}^{(1)} - \hat{x}_l^{(1)} \quad (11)$$

The robustness value of different control system sub unit is described accurately.

Probability robustness analysis. This paper takes a control system of a large computer control system as an example for study. Figure 1 is a description of a typical closed-loop control system.

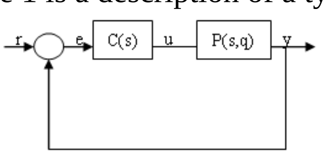

Figure 1 closed-loop control system

In the formula, $C(s)$ is used to describe the controller, $P(s,q)$ is used to describe the object, s is used to describe the nonlinear measurable variable function, q is used to describe the uncertain parameters of the historical definition in Q collection, $q = (q_1, q_2, \cdots, q_m)$.

The random vector in multivariate probability density function $f(q)$ is described by q, the sampling interval is described with N, the sampling points of independent and identical distribution is described with q, and $f(q)$ is used to describe the probability density function, assuming the sensitivity function is described as follows:

$$S(s,q)=\frac{1}{z+P(s,q)C(s)} \tag{12}$$

Supposing that $S(s,q)$ is a stable transfer function, that is:

$$u(q)=\|S(s,q)\|_{\infty}=\sup_{\omega}\left|S(j\omega,q)\right| \tag{13}$$

Assuming that the performance index $\gamma>0$, the probability is made to satisfy the following conditions:

$$u(q)\le\gamma \tag{14}$$

Give an indication function $I(q)$

$$I(q)=\begin{cases}1,u(q)\ge\gamma\\0,u(q)<\gamma\end{cases} \tag{15}$$

The probability of unacceptable performance can be described as:

$$P=P(u(q)\ge\gamma)=\int_{Q}f(q)I(q)dq \tag{16}$$

Through the Monte Carlo to simulate and calculate P, which is based on the given $f(q)$ to randomly access N sub samples, then arithmetic average value of statistics $I(q)$ can be acted as estimated amount of P, i.e.:

$$P=\frac{1}{N}\sum_{i=1}^{N}I(q_i) \tag{17}$$

If the estimation accuracy $\varepsilon\in(0,1)$ and confidence level $\delta\in(0,1)$ are known, $\hat{P}$ with a certain probability values converges to P, which makes the:

$$P\left\{\left|P-\hat{P}\right|\le\varepsilon\right\}\ge1-\delta \tag{18}$$

The reasonable minimum number of Monte Carlo random trials is:

$$N_{\min}\ge\ln\frac{1}{\delta}/\ln\frac{1}{1-\varepsilon} \tag{19}$$

The probability metric of system performance index reliability is called the degree of robustness, the formula can be described as follows:

$$\gamma_{2i}=\begin{bmatrix}t_{si} & \delta_i & t_{ai} & \eta_i & x_i\end{bmatrix}*\omega_i^T, i=(1,2,3,4,5) \tag{20}$$

In the formula, t_{si}, δ_i, t_{ai}, η_i, x_i are used to describe the adjustment time, overshoot, time back to zero, stability time and probability values of controllable degree, ω_i is used to describe the corresponding weight, through the hierarchy analyzing method it can be known that, $i=(1,2,3,4,5)$ is used to describe the number of evaluation index needs to be analyzed analysis in the calculation process of robustness, which can be increased based on the need.

Simulation experiment

Data sources. The experimental data is from robustness test data in a supermarket management system, the data is divided into training set and test set, the last 10 data is viewed as the test set, the rest are training sets, which can be used to establish software reliability prediction model.

Data preprocessing. Because the changing range of software reliability data is large, equation (9) is carried on for pretreatment, to make the range between [0 1].

$$x'=\frac{x-x_{\min}}{x_{\max}-x_{\min}} \tag{9}$$

In the formula, x is the original data, $x_{\max}$ and $x_{\min}$ is the maximum value and the minimum value.

Results of robustness detection. Using computer control system on the training set to learn the software reliability, software robustness detection model is established, and then use the model to detect the test set, test results obtained as shown in table 1.

Table 1 the test results of robustness analysis

The total number of errors	The predicted value	The prediction accuracy
347	395	86.17%
389	375	96.40%
342	331	96.78%
377	386	97.61%
424	393	92.69%
406	393	96.80%
395	377	95.44%
381	406	93.44%
475	501	94.53%
401	415	96.51%
The average prediction accuracy		94.64%

Modification of probability robustness analysis on detection results. According to the relative value and the actual situation, the system is divided into 3 states respectively, E_1 , E_2, E_3, including: $E_1 \in (86.17\%, 93.44\%]$, $E_2 \in (94.53\%, 95.44\%]$, $E_3 \in (96.40\%, 97.61\%]$, and then according to the state corresponding to the detection point, the state transition probability matrix of each step can be acquired:

$$P^{(1)} = \begin{bmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 0 \\ 1 & 0 & 1/2 \end{bmatrix} \quad P^{(2)} = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 1/4 & 1/4 \\ 1/2 & 1/2 & 0 \end{bmatrix} \quad P^{(3)} = \begin{bmatrix} 1/2 & 1/2 & 1/4 \\ 1/4 & 1/8 & 1/8 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

The first data point being detected as an example, because the state is divided into 3, so 3 data close to the first data point are selected to make the prediction table, transfer steps are 1, 2, 3, in the transfer probability matrix corresponding to transfer step number, corresponding line vector of initial state is obtained to get the first data point detection table.

The maximum value of the sum of transfer probability is 1/4, the corresponding state is E_1, total number of discovered errors of the first data point in this software system is at the state E_1, according to the total number of discovered errors of one data point in this software system to obtain the detection value 395, according to equation (10) can get the detection value of probability robustness analysis 361, predicted detection accuracy is 95.97, compared to the single grey model, the detecting precision of the proposed method has been improved, in the same way, next detection value of other points can be calculated, the detection results are shown in table 2.

Table 2 test results of probabilistic robustness analysis

The total number of errors	The test value	The test accuracy
347	351	98.86%
389	395	98.48%
342	333	97.37%
377	386	97.67%
424	419	98.82%
406	399	98.28%
395	382	96.71%
381	368	96.59%
475	488	97.34%
401	410	97.80%
The average test accuracy		97.79%

Testing results of probability robustness analysis. The obtained results are shown in table 3. From table 3 we can see, the prediction accuracy of a single Markov chain model is significantly lower than that of the proposed method.

Table3 testing results of probability robustness analysis detection model

The total number of errors	The test value	The test accuracy
347	344	99.14%
389	385	98.97%
342	338	98.83%
377	394	95.69%
424	430	98.60%
406	402	99.01%
395	387	97.97%
381	376	98.69%
475	473	99.58%
401	392	97.76%
The average test accuracy		98.42%

Results and analysis. The detection residual error of grey model, Markov chain model, probability robustness analysis detection model is calculated, the comparison graph of testing value and real value of three kinds of models as shown in Figure 2 and the absolute value of the detection residual error, as shown in figure 3.

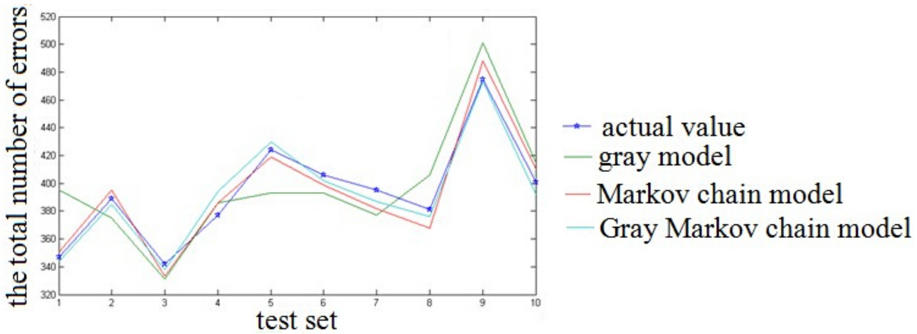

Figure 2 The test data in a supermarket management system

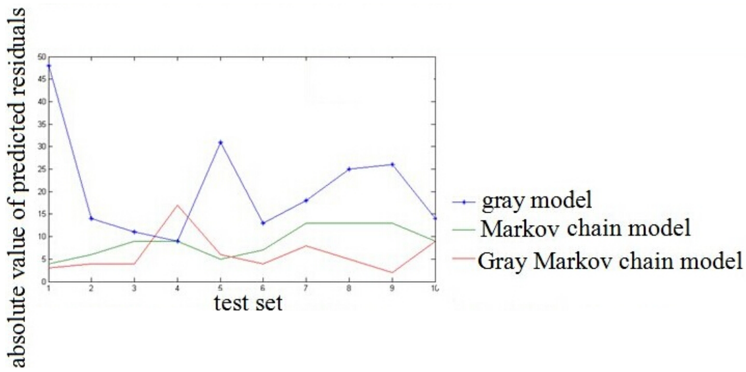

Figure 3 The test data in a supermarket management system

After comparing figure 2 and 3, it can be known that compared with the traditional detection model, the proposed detection and analysis model has higher detection accuracy and smaller error.

Conclusions

In view of the features of software robustness, like many affecting factors, strong uncertainty, great stochastic volatility, first using the computer control system to detect the software robustness, then probability robustness analysis model is utilized to modify test results. The simulation results show that, comparing to the single model or the traditional model, system and the combined model of probability robustness analysis and computer control system realizes the complementary advantages, can reveal the overall change trend of the robustness, and also overcome the influence of randomly fluctuated data on the detection accuracy, the defection accuracy of software robustness is higher, so as to achieve the goal of scientifically detecting software robustness, and provides the powerful technical support for the detection of software development.

References

[1] Huang Xizi. Software reliability, safety and quality assurance [M]. Beijing: Publishing House of electronics industry, 2002

[2] Zhang Xuesong, Guo Ping. Prediction of software reliability based on combined neural network [J]. Journal of Beijing Normal University (SCIENCE EDITION) 2005, 41 (6): 599-604.

[3] Wang Hao. Research on software reliability model of Based on data mining technology [D]. Beijing: Dissertation of Beihang University, 2002

[4] Zou Fengzhong, Xu Renzuo. Multiple model comprehensive evaluation of software reliability [J]. Journal of Tongji University, 2002, 30 (10): 1183-1185.

[5] Ma Sasa, Chen Zili, Zhao Shouwei. Research of evaluation and prediction of software reliability based on clustering [J]. Computer application, 2005, 25 (12): 269-372.

[6] Hu Qiusuo, Zhong Cheng. Software module risk prediction model based on neural network [J]. Computer engineering and application, 2007, 43 (18): 106-110.