Proceedings of the 8th WSEAS International Conference on APPLIED MATHEMATICS, Tenerife, Spain, December 16-18, 2005 (pp184-188)

Application of Kalman Filter in Time Series Software Reliability Growth Model

GUO JUNHONG[1,2], LIU HONGWEI[1], YANG XIAOZONG[1], ZUO DE CHENG[1]
School of Computer Science and Technology
[1]Harbin Institute of Technology
No.92, West Da-zhi Street, Harbin, 150001
School of Computer Science and Technology
[2]Heilongjiang University
No.74, Xue Fu Road, Harbin, 150080
CHINA

Abstract: - Researches show that assumptions condition of existing software reliability growth models are difficult to be satisfied in actual projects which restrict the universality of models. Classical models neglect observation noise and its affection on accurate evaluation to software reliability. This paper proposes a time series software reliability growth model and transforms it into state space model and Kalman filter is used to reduce noise. Testing data of filtering noise can shows the essential rule of data better and improves goodness of fit. Simulation result shows the validity of this method.

Keywords: - Software reliability growth model; Kalman filter; Observation noise; Time series analysis; software reliability evaluation/prediction

1 Introduction

The applications of computer systems are so widely spread and the software systems play a more and more important role in the whole system. Software reliability is one of the essential factors that affect software performance. The definition of reliability for software is the probability of execution without failure for some specified interval of natural units or time [1,2].

Many software reliability growth models have been proposed to evaluate software reliability. Classical software reliability growth models have great influence on software reliability modeling research. Software reliability modeling has become one of the most important aspects in software reliability engineering since Jelinski-Moranda model appeared [3]. Software reliability modeling is often concerned with the behavior of software reliability and uses historical software reliability failure data to assess current software reliability status and forecast future software failures [4].

The assumptions conditions are the key factors of establishing software reliability growth model. There is a relation between assumptions chosen and modeling success. But in practical application, quite a few assumptions are not accordant with actual software development and test environment and these assumptions restricted the universality of models. So it has reached a common viewpoint in software reliability evaluation field that none of these models is able to cope properly with all the possible situations [5].

To apply these models, it is necessary to know how well the models suit an actual observation failure data set. Large disagreements sometimes appear among the software reliability predictions obtained from different software reliability growth models. So another important issue in software reliability modeling is to improve as much as possible the prediction accuracy [6].

Based on above discussion, this paper proposes a universal method for software reliability prediction by time series analysis. We establish a time series autoregression model and transform it to state space model. Using Kalman filter, more accurate prediction results are obtained.

This paper is organized as follows. Section 2 discusses observation noise of software testing data which has been neglected by many classical software reliability growth models. Section 3 shows the feasibility of software reliability modeling based on time series and the implemented algorithm is given. In section 4, the simulation analysis of testing data is provided. Finally, a brief conclusion is presented.

2 Observation Noise and Kalman Filter

Some former models neglect observation noise. In fact, many factors affect testing data. What you get are not real data, that is to say observation data exist disturbance, which is called "observation noise". Observation noise has white noise and colored noise. To compute simply, we treat disturbance as white noise with zero mean [8].

Kalman filter theory is based on a state-space approach in which a state equation model and an observation equation model are shown. In data processing, a filter is a function or procedure which removes unwanted disturbance. The concept of filtering and filter functions is particularly useful in engineering [9].

Kalman filter and the Wiener filter are two important linear filters for data estimation. During 1940s, in order to meet the requirements derived from World II war military technology, classical Wiener filtering theory was proposed by American scholars N. Wiener and Kolmogorov respectively. Wiener filtering theory was based on frequency domain method and suitable for stationary stochastic processes. The concepts of state variables and state space for systems were introduced by American scholars R.E. Kalman and R. S. Bucy in 1960. They presented the state space method on time domain that was called Kalman filtering theory. Considering the statistic characteristics of estimated variables and measurements, optimal recursive filtering algorithms were obtained by using the Kalman filtering theory. They were suitable for multi-variable systems, time-varying systems and non-stationary stochastic processes and easy for real-time implementation, thus overcame the shortcomings and limitations of classical Wiener filtering theory [10].

If we establish a time series autoregression model, we can transform it into state space model and use Kalman filter to reduce disturbance of failures data.

3 The Implemented Algorithms

Time series analysis theory is a method of describing statistics character of dynamic data, which can set up time series model from limited sample data. Its advantage is convenience and practicality. There are many contributions on estimation and prediction with autoregression time series models. Time series analysis method is well studied in some statistical literatures. However, its use in software reliability engineering is rather limited [11].

Time series is defined as an ordered sequence of values of a variable at equally spaced time intervals [12].

Based on software reliability analysis, input data are cumulative number of software failures or failure intervals mainly. That is to say, software reliability failure data are discrete data sequence. Whether it is steady or not, we can use the data to modeling and evaluate software reliability by applying proper time series method.

The cumulative number of failures $M(k)$ is increasing and trend to a fixed value. Considering observation disturbance, we can establish the following time series model:

$$M(k) = \theta(k)M(k-1) + \varepsilon(k) \tag{1}$$

where $\varepsilon(k)$ is zero mean white noise.

Here, it is assumed that the data is observed with white noise and the testing data is uncorrelated with the observation noise. From the autoregression model, we can establish the state space model as the following:

$$X(k) = \varphi(k)X(k-1) + w(k-1) \tag{2}$$

$$y(k) = X(k) + v(k) \tag{3}$$

where $X(k)$ is the system state vector, $y(k)$ is the observation vector, $w(k)$ is the process noise vector and $v(k)$ is the observation noise vector. $w(k)$ and $v(k)$ in this case are assumed to be mutually independent and zero mean white noise. The covariances of $w(k)$ and $v(k)$ are given as

$E[w(k), w(k)^T] = Q, E[v(k), v(k)^T] = R$.

In which, we can get the time series $\{\varphi(k)\}$ from $\{\hat{\theta}(k)\}$ by applying smoothing filter:

$$\varphi(k) = \lambda\varphi(k-1) + (1-\lambda)\hat{\theta}(k) \tag{4}$$

where the initial is $\varphi(1) = \hat{\theta}(1)$ and $\lambda = 0.87$ in the simulation. Series $\{\hat{\theta}(k)\}$ is calculated from equation (12) to (14).

The model system has the following Kalman filter equations [13,14]:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)\varepsilon(k+1) \tag{5}$$

$$\hat{X}(k+1|k) = \varphi(k)\hat{X}(k|k) \tag{6}$$

$$\varepsilon(k+1) = y(k+1) - \hat{X}(k+1|k) \tag{7}$$

$$K(k+1) = P(k+1|t)[P(k+1|k) + R]^{-1} \tag{8}$$

$$P(k+1|k) = \varphi(k)P(k|k)\varphi(k)^T + Q \tag{9}$$

$$P(k+1|k+1) = [I_n - K(k+1)]P(k+1|k) \tag{10}$$

$$\hat{X}(0|0) = \mu_0, P(0|0) = P_0 \tag{11}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K_{RLS}(k+1) [y(k+1) - \hat{\theta}(k)\hat{X}(k|k)] \tag{12}$$

$$K_{RLS}(k+1) = \frac{P_{RLS}(k)\hat{X}(k+1|k+1)}{\omega + P_{RLS}(t)\hat{X}(k+1|k+1)^2} \qquad (13)$$

$$P_{RLS}(k+1) = [1 - K_{RLS}(k+1)\hat{X}(k+1|k+1)] P_{RLS}(k)/\lambda \qquad (14)$$

To calculate above equations, for $\hat{X}(0|0) = 4$, $P(0|0) = 10000$, $Q = 2.845$, $R = 77.216$, $\omega = 0.56$.

4 Simulations and Analysis

Software testing data in table 1 comes from Data7 in chapter 17 in [15], where Day is the test time in days and CF is cumulative number of software failures.

Table 1 A set of software failure data

Day	CF	Day	CF	Day	CF	Day	CF
0	4	**38**	186	**65**	374	**87**	494
2	11	**40**	193	**66**	379	**88**	496
4	21	**41**	200	**67**	386	**89**	497
9	34	**43**	205	**68**	393	**90**	508
11	42	**45**	212	**69**	407	**91**	509
16	55	**48**	218	**70**	420	**92**	511
17	59	**49**	224	**71**	434	**93**	513
20	66	**50**	228	**72**	445	**94**	517
22	74	**51**	240	**73**	447	**95**	518
23	75	**52**	246	**74**	451	**96**	522
24	81	**53**	253	**75**	455	**97**	523
26	94	**54**	261	**76**	458	**98**	524
27	101	**55**	272	**77**	464	**99**	526
28	110	**56**	278	**78**	470	**100**	527
29	118	**57**	287	**79**	473	**101**	528
31	123	**58**	294	**80**	476	**102**	529
32	133	**59**	306	**81**	480	**103**	530
33	140	**60**	318	**82**	481	**104**	532
34	151	**61**	333	**83**	483	**105**	533
35	156	**62**	347	**84**	484	**107**	535
36	164	**63**	354	**85**	486		
37	177	**64**	363	**86**	491		

Table 2 Comparison of observed data and estimated data of Goel-Okumoto (GO) model and the autoregression model with kalman filter

Day	Observed Data	GO Estimated Data	GO Estimated Error	Kalman Estimated Data	Kalman Estimated Error
5	21	64.1607	43.1607	27.2545	6.2545
15	42	172.2497	130.2497	46.2239	4.2239
25	81	258.0441	177.0441	84.8134	3.8134
35	156	326.1426	170.1426	159.4130	3.4130
45	212	380.1950	168.1950	209.6867	-2.3133
55	272	423.0986	151.0986	268.0939	-3.9061
65	374	457.1528	83.1528	380.9347	6.9347
75	455	484.1830	29.1830	460.1247	5.1247
85	486	505.6379	19.6379	485.0410	-0.9590
95	518	522.6676	4.6676	520.8737	2.8737

Table 3 Comparison of observed data and predicted data of Goel-Okumoto (GO) model and the autoregression model with Kalman filter

Day	Observed Data	GO Predicted Data	GO Predicted Error	Kalman Predicted Data	Kalman Predicted Error
98	524	527.0553	3.0553	524.2106	0.2106
99	526	528.4516	2.4516	525.4241	-0.5759
100	527	529.8160	2.8160	526.6403	-0.3597
101	528	531.1492	3.1492	527.8593	-0.1407
102	529	532.4520	3.4520	529.0812	0.0812
103	530	533.7251	3.7251	530.3059	0.3059
104	532	534.9691	2.9691	531.5334	-0.4666
105	533	536.1846	3.1846	532.7638	-0.2362
106	533	537.3725	4.3725	533.9970	0.9970
107	535	538.5332	3.5332	535.2331	0.2331

Table 4 Comparison of relative error

Day	Estimated Relative Error		Day	Predicted Relative Error	
	GO model	With Kalman		GO model	With Kalman
5	2.0553	0.2978	**98**	0.0058	0.0004
15	3.1012	0.1006	**99**	0.0047	-0.0011
25	2.1857	0.0471	**100**	0.0053	-0.0007
35	1.0907	0.0219	**101**	0.0060	-0.0003
45	0.7934	-0.0109	**102**	0.0065	0.0002
55	0.5555	-0.0144	**103**	0.0070	0.0006
65	0.2223	0.0185	**104**	0.0056	-0.0009
75	0.0641	0.0113	**105**	0.0060	-0.0004
85	0.0404	-0.0020	**106**	0.0082	0.0019
95	0.0090	0.0055	**107**	0.0066	0.0004

$$\text{Estimated relative error} = \frac{\hat{M}(k) - M(k)}{M(k)}$$

$$\text{Predicted relative error} = \frac{\hat{M}(N+p|N) - M(N+p)}{M(N+p)}$$

Table 5 Comparison of SSE

Estimated SSE		Predicted SSE	
GO model	With Kalman	GO model	With Kalman
1358400	2934.3	109.5439	1.9473

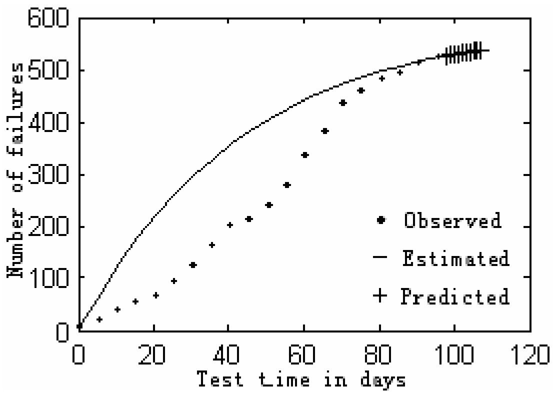

Fig.1 Goel-Okumoto model

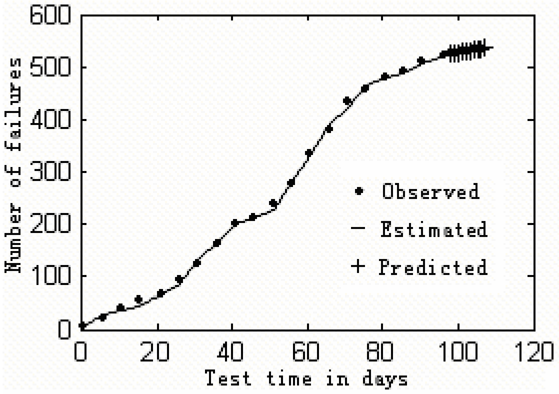

Fig.2 $M(k)$, $\hat{M}(k)$ **and** $\hat{M}(N+p\,|\,N)$

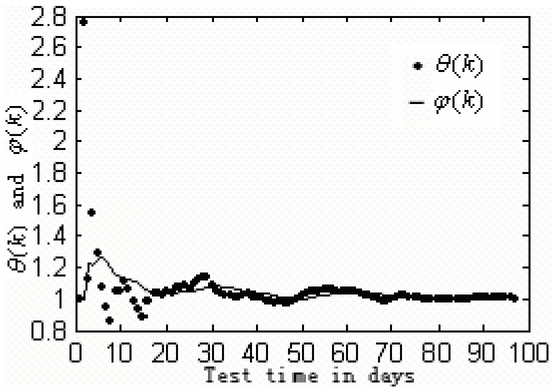

Fig.3 $\theta(k)$ **and** $\varphi(k)$

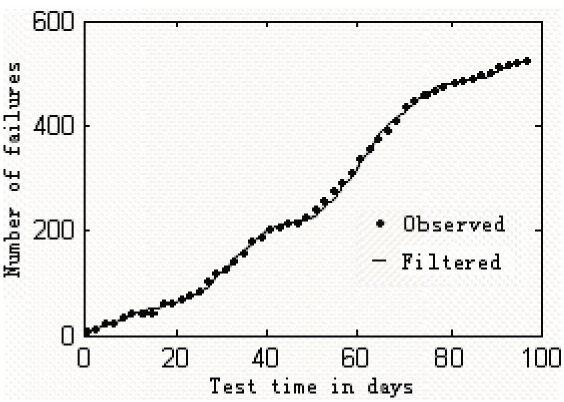

Fig.4 $X(k)$ **and** $\hat{X}(k\,|\,k)$

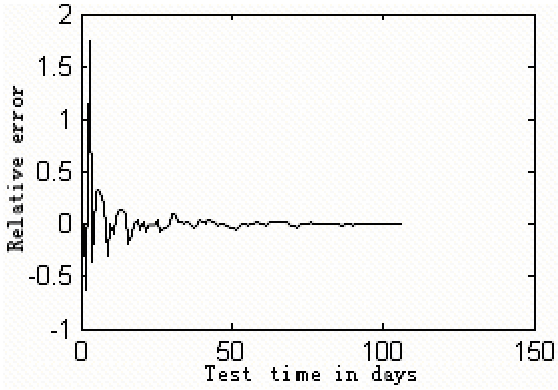

Fig.5 Relative error

In Table 2, we can see that the estimated data of the autoregression model with Kalman filter are more close to real observation data. The estimated errors are far less than Goel-Okumoto model's. Goel-Okumoto nonhomogeneous Poisson process model has a strong influence on software reliability modeling. So we illustrate the proposed model and Goel-Okumoto model. To verify fitting quality and prediction quality of the proposed model, this paper divides software failures into two parts, the first part data are treated as fitting data. According to the fitting results, we can predict the second part failures data. The comparing result of observed data and predicted data shows that the prediction quality of the proposed model is validated.

Table 3 shows the similar contrast about the predicted data of the two models. We find the relative errors of the autoregression model with Kalman filter are very small. Further, Table 4 shows the comparison of relative errors of filtered model and Goel-Okumoto model. From this table, we can get the conclusion that both relative errors of estimated and predicted data with Kalman filter are far less than Goel-Okumoto model's. The sums of square errors (SSE) [16] are calculated in Table 5. As can be seen from the table, SSE of the observed data, the estimated data and the predicted data are illustrated to show the predictive validity of the new model with Kalman filter. A SSE value closer to zero indicates a better fit.

Fig.1 and Fig.2 illustrate corresponding data of the Goel-Okumoto model and the new model with Kalman filter, including observed data, estimated data and predicted data. We can see that the proposed model can fit the failures data better because it uses weighted least squares method and emphasizes the effect of current failures data.

Fig.3 gives the curve of parameter $\theta(k)$ and $\varphi(k)$. Actually, $\varphi(k)$ is the filter data of $\theta(k)$. The filtered data can improve the goodness of fitting model.

In Fig. 4, we can see that $X(k)$ are observation failures data and $\hat{X}(k\,|\,k)$ are filtered data using

Kalman filter. The calibrated failures data can filter disturbance and reflect the nature of data. Fig.5 illustrates the relative errors and shows good evaluation results of the new model.

5 Conclusion

According to the character of time series, an autoregression model is established and transformed to state space model. In the model, the using of Kalman filter reduces observation noise. The model with Kalman filter can represent the actual software failures data relatively.

The new model considers the disturbance noise of observed data and has no need for some unrealistic assumptions condition. All these accord with the characteristic of real projects.

As the parameters are time-varying, the proposed model with Kalman filter can suit different software testing data and has more widely applicability. By compare with Goel-Okumoto model, the proposed model fits the real project fairly well. The simulation experiments verify the accuracy and efficiency of this new model.

References:

[1] Huang Xizi, *Software Reliability*、*Security and Quality Assurance*, Publishing House of Electronics Industry, 2002.

[2] FL. Popentiu, D.N. Boros, Software Reliability Growth Supermodels, *Microelectronics and Reliability*, Vol.36(4), 1996, pp. 485-491.

[3] J.Musa, A.Jannino, and K.Okumoto, *Software Reliability- Measurement, Prediction, Application*, McGraw Hill, New York, 1987.

[4] PK Kapur, S. Younes, Modelling an imperfect debugging pheonmenon in software reliability, *Microelectronics and Reliability*, Vol.36, 1996, pp. 645-650.

[5] Amerit L Goel., Software reliability models: Assumptions,limitations and applicability, *IEEE transaction on software engineering*, SE11 (12), 1985, pp. 1411~1425.

[6] PK Kapur, S. Younes, Software reliability growth model with error dependency, *Microelectronics and Reliability*, Vol.35 (2),1995, pp. 273-278.

[7] Z. Jelinski and PB Moranda, Software Reliability Research, *Statistical Computer Performance Evaluation*, Academic, New York, 1972, pp. 465-485.

[8] Seiichi Nakamori, Filtering algorithm based on innovations theory for white Gaussian plus colored observation noise, *Journal of Information Science and Engineering*, Vol.5, 1989, pp.157-166.

[9] Seiichi Nakamori, Estimation technique using covariance information in linear discrete-time systems, *Signal Processing*, vol.43(2), 1995, pp. 169-179.

[10] Giuliano De Rossi, Kalman filtering of consistent forward rate curves: a tool to estimate and model dynamically the term structure, *Journal of Empirical Finance*, vol.11(2), 2004, pp. 277-308.

[11] Wang ZL., *Time series Analysis*, Publishing House of Chinese Statistics, 2000.

[12] Seiichi Nakamori, Estimation of multivariate signal by output autocovariance data in linear discrete-time systems, *Mathematical and Computer Modelin*g, vol.24(1), 1996, pp.97-114.

[13] Dengzili, *Modern Time Series Analysis and Application*, Publishing House of knowledge, 1989.

[14] Dengzili, *Theory and Application of Optimal Filtering: Modern Time Series Analysis Method*, Publishing House of Harbin Institute of Technology, 2000.

[15] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*, McGraw Hill, 1996.

[16] Takamasa Nara, Masahiro Nakata, Akihiro Ooishi, Software Reliability Growth Analysis-Application of NHPP Models and Its Evaluation, *IEEE Transactions on Reliability*, HASE 1996, pp. 222-227.