Article citation info:
XU J, MENG Z, XU L. Integrated system of health management-oriented reliability prediction for a spacecraft software system with an adaptive genetic algorithm support vector machine. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2014; 16 (4): 571–578.

Jiuping XU
Zhiyi MENG
Lei XU

# INTEGRATED SYSTEM OF HEALTH MANAGEMENT-ORIENTED RELIABILITY PREDICTION FOR A SPACECRAFT SOFTWARE SYSTEM WITH AN ADAPTIVE GENETIC ALGORITHM SUPPORT VECTOR MACHINE

# ZORIENTOWANE NA ZINTEGROWANE ZARZĄDZANIE KONDYCJĄ SYSTEMU PROGNOZOWANIE NIEZAWODNOŚCI SYSTEMÓW OPROGRAMOWANIA STATKÓW KOSMICZNYCH Z WYKORZYSTANIEM OPARTEJ NA ADAPTACYJNYM ALGORYTMIE GENETYCZNYM MASZYNY WEKTORÓW NOŚNYCH

*Software reliability prediction is very important to minimize cost and improve software development effectiveness, especially in a spacecraft's software system. In this paper, a new spacecraft software system reliability definition is given and a new reliability prognostics-oriented life cycle integrated system health management for a spacecraft software system is focused on. Adaptive genetic algorithms are then combined with a support vector machine to build an adaptive genetic algorithm support vector machine reliability prediction model. This model attempts to overcome the genetic algorithm weaknesses, such as the local minima and premature convergence problems, and solves the parameter selection difficulties often encountered in a support vector machine. After construction, the proposed adaptive genetic algorithm support vector machine model is employed to predict the reliability of a spacecraft software system. Finally, a numerical example is given to show how the proposed approach has a superior prediction performance compared to a standard support vector machine and artificial neural network.*

**Keywords**: *spacecraft software system, reliability, integrated system health management, adaptive genetic algorithms support vector machine.*

*Przewidywanie niezawodności oprogramowania odgrywa ważną rolę w minimalizowaniu kosztów i poprawie efektywności tworzenia oprogramowania, zwłaszcza w odniesieniu do systemów oprogramowania statków kosmicznych. W niniejszej pracy, podano nową definicję niezawodności systemu oprogramowania statku kosmicznego koncentrując uwagę na opartym na prognozowaniu niezawodności oraz cyklu życia modelu zintegrowanego zarządzania kondycją systemu opracowanego dla systemu oprogramowania statku kosmicznego. Skonstruowano następnie model przewidywania niezawodności oparty na połączeniu adaptacyjnych algorytmów genetycznych oraz maszyny wektorów nośnych. Model ten stanowi próbę przezwyciężenia słabości algorytmów genetycznych, takich jak problem minimów lokalnych czy problem przedwczesnej zbieżności, a także rozwiązania trudności związanych z doborem parametrów, jakie często występują przy zastosowaniu maszyny wektorów nośnych. Skonstruowany model opartej na adaptacyjnym algorytmie genetycznym maszyny wektorów nośnych zastosowano do przewidywania niezawodności systemu oprogramowania statku kosmicznego. Wreszcie, przedstawiono przykład liczbowy, który pokazuje że opracowany model charakteryzuje się wyższą dokładnością prognozowania w porównaniu do standardowej maszyny wektorów nośnych oraz sztucznej sieci neuronowej.*

**Słowa kluczowe**: *system oprogramowania statku kosmicznego, niezawodność, zintegrowane zarządzanie kondycją systemu, adaptacyjne algorytmy genetyczne, maszyna wektorów nośnych.*

## 1. Introduction

As interest in the development of the space industry increases, the need for more reliable spacecraft is becoming crucial [15, 19, 23]. Future manned and unmanned space missions to the International Space Station, the Moon, Mars, and beyond means longer mission durations and a reliance on a more complex assemblage of components, both of which increase the probability of operational mission failures [30]. To ensure that the spacecraft operates as planned, every element of the vehicle as well as the completely assembled spacecraft itself must be tested on the ground under conditions simulating those it will face in space [31, 27, 25]. Consequently, a life cycle integrated system health management system which focuses on early spacecraft design, operations and general maintenance is required [28].

A spacecraft's software system is directly related to the possibility of mission failure because of the critical functions and complex operating environment. Spacecraft software system reliability (SSSR) is a critical index needed to ensure system reliability [18]. Although the importance of spacecraft software system reliability has long been realized, aerospace disasters caused by faults in the spacecraft software system (SSS) still occur. In 1996, a software failure in the Ariane 501 developed by the European Space Agency (ESA) resulted in a rocket explosion 40 seconds after launch, causing billions of dollars of economic losses. In 1999, software failures caused the landing engine on NASA's Mars Polar Lander to prematurely shut down resulting in a crash. Therefore, in spacecraft systems, any small software error may lead to entire mission failure, resulting in not only economic losses but also possible loss to human life and property. To deal with the safety and maintenance of a manned spacecraft, and especially that of the spacecraft software system reliability, a life cycle integrated system health management (ISHM) system which focuses on early design, operations and general maintenance is vital [9].

Generally, the ISHM consists of in situ monitoring, condition assessment, fault diagnosis, prognosis, and appropriate decision making, so it is a more comprehensive system than traditional prognostics and health management (PHM) systems. Health management has emerged as one of the key enablers for efficient system-level maintenance and lower life cycle costs. Diagnostics assesses the system's current reliability, and prognostics identify potential future failures. Using the prognostics information, health management systems maintain a system or equipment in working condition [32, 1]. Integrated system health management is a framework of integrated technologies that evaluates system reliability in actual life cycle conditions to determine the advent of failure and to mitigate system risks. ISHM does this by monitoring the health of a product or system, and then estimating reliability through an evaluation of the deviation or degradation from an expected health state and usage conditions [26]. The spacecraft software system life cycle includes early design, operation and maintenance over two stages.

As software size and complexity have increased, software development has moved toward modular designs [12]. This is especially true of the new generation spacecraft software systems monitored by life cycle integrated system health management. The life cycle integrated system health management discussed in this paper focuses on reliability prediction in the early design stage of the spacecraft software system. Basically, the spacecraft software system reliability mechanism is able to quantify the operational profile of the spacecraft software system. However, as software tends to develop defects and faults over time, software reliability also changes with time. Thus, as the number of faults grows, predicting software reliability over time becomes increasingly difficult. Consequently, as prognostics are the core of integrated system health management, the design of a life cycle integrated system health management with efficient prognostics technology is a very important research field [29].

There has been significant research into software reliability. Pietrantuono, Russo and Trivedi [22] proposed an architecture-based approach for software reliability and testing time allocation. Huang and Lin [13] presented an analysis of software reliability modelling by testing compression factors and failure-to-fault relationships. Amin, Grunske and Colman [18] outlined an approach to software reliability prediction based on time series modelling. Garg, Lai and Huang [10] studied a problem from the perspective of software reliability models which focused on when to stop testing. Some research has also specifically focused on spacecraft software system reliability. Wang [27] studied SSS design and performance tests from evaluation to release. However, few studies have focused on spacecraft software system reliability by investigating the life cycle integrated system health management.

As the spacecraft software system is very complex, there is a large quantity of random volatile fault data. Because of this, there is no single approach capable of resolving all software reliability life cycle integrated system health management problems as all approaches have both advantages and disadvantages. However, spacecraft software system reliability prediction, accuracy and timeliness are vital for decision-makers. Therefore, in this paper, adaptive genetic algorithms (AGA) [7, 18] combined with support vector machines (SVM) [33, 6, 14] are used to build an adaptive genetic algorithm-support vector machine prediction model. The adaptive genetic algorithm-support vector machine attempts to overcome the traditional weaknesses of genetic algorithms, such as the local minima and premature convergence problems, and solves support vector machine problems, such as parameter selection difficulty. This fusion approach is then used in a spacecraft software system reliability prediction case study. The remainder of this paper is organized as follows; Section 2 outlines the spacecraft software system reliability prediction problem and the fusion prediction approach. The proposed adaptive genetic algorithm-support vector machine is built and elaborated on in Section 3. In Section 4, a numerical example is given to show an application of the proposed model and algorithm. Concluding remarks are given in Section 5.

## 2. Spacecraft software system reliability prognostics

Software reliability is typically very complex [27] because of the number of features and the need to ensure a high level of safety and reliability, as shown in Fig. 1. Spacecraft software system reliability requires that interstellar functions and ground station functions work synchronously. Interstellar functions are made up of navigation calculations, housekeeping, fault monitoring, command processing, spacecraft subsystem management, general management and the communications payload. Ground station functions are made up of data processing, data compression and storage, spacecraft telemetry remote control, user interfaces, and operating condition monitoring and maintenance. Both interstellar and ground systems require high reliability, and this is particularly important for the interstellar software, which is typically an embedded real-time system. This complexity also leads to significantly higher software development costs. Therefore, to ensure the normal operation of a spacecraft and to avoid mission failure, it is necessary to focus on spacecraft software system reliability in the early design stage. This means that to develop highly reliable spacecraft software system reliability, verification techniques are necessary, so, besides the traditional techniques such as testing,

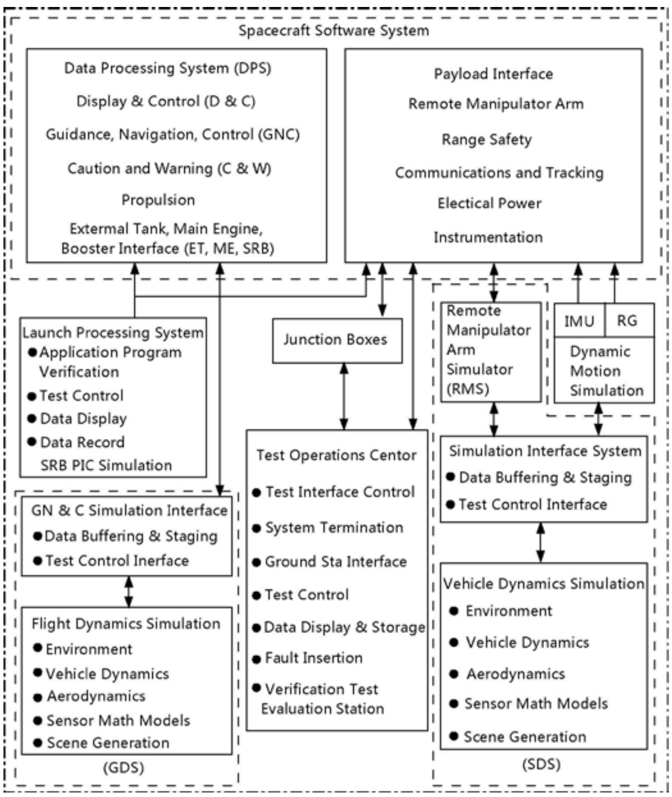


*Fig. 1. Integrated Spacecraft Software Systems*

automated verification techniques such as life cycle integrated system health management-oriented spacecraft software system reliability prediction are also critical [17].

Assessing the software reliability of life cycle integrated system health management is a complex task, because of the multiple time stages, the complexity of the system structures, the large number of parameters, the competing failure mechanisms and the presence of intermittent faults and failures [2, 29, 30]. The software reliability life cycle integrated system health management processes are shown in Fig. 2. Continuous health monitoring processes provide information about the system's performance, the environment and the operational loading, the data from which is required for life cycle integrated system health management data manipulation. The system's performance is then compared with a historical database, the faulty parameter isolated and the product damage assessed. Following this, parameter selection and isolation are carried out to identify the parameters that are contributing to the abnormal status of the system. The reliability is then assessed using diagnostic approaches, and, through the use of prognostic algorithms, the level of deviation or degradation is identified and the advent of failure is predicted by determining the distribution of remaining life.

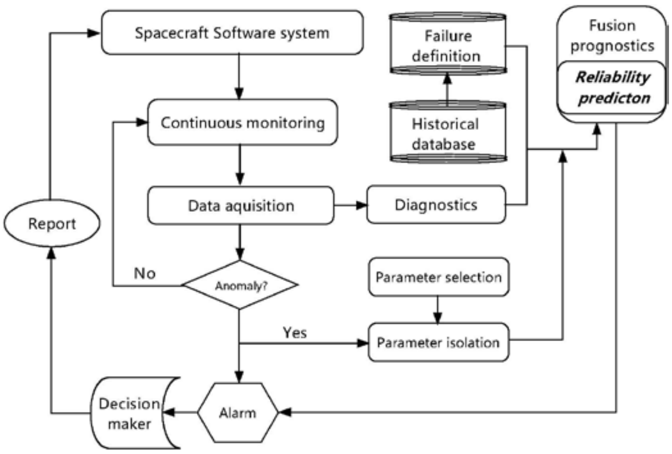


*Fig. 2. ISHM-oriented fusion prognostics framework for SSS*

Spacecraft software system reliability is defined as the probability of a failure-free software operation for the operational phase in a specified environment [31], and has quality characteristics, as it is able to quantify the operational profile of a spacecraft software system. Spacecraft software system reliability generally changes over time leading to the appearance of defects and faults, which means that the failure space time (FST) is the key to an accurate assessment of spacecraft software system reliability. As a result, before the spacecraft software system is put into operation the proposed life cycle integrated system health management for spacecraft software system reliability prediction is able to identify faults and remove them in the early design stage. Spacecraft software system reliability in the early design stage is focused on in this paper, as shown in Fig. 3.

For complex spacecraft software systems, the application of intelligent technology assists in parameter optimization and accuracy improvements. Support vector machines have a good generalization ability as they utilise a statistical learning theory based on dimension theory and structural risk minimization rules [32], rather than traditional empirical risk minimization principles. The support vector machines' basic idea is the mapping of the data to a larger dimensional

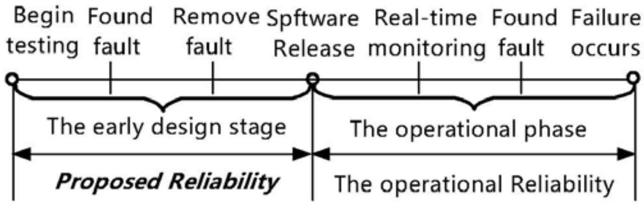


*Fig. 3. The proposed SSS reliability*

feature space where a linear regression is conducted using nonlinear mapping. This has been shown to have a positive effect in small sample, high dimension, non-linear prediction areas. However, parameter selection can have a significant influence on the prediction effect. Because adaptive genetic algorithms have a strong global optimization ability, a type of automatic parameter selection method using adaptive genetic algorithms is established in this paper. Here, we propose an intelligent fusion prediction model based on support vector machine prediction and adaptive genetic algorithmic theory. Fusion prediction, a synthesis of these two different theories, has the ability to strengthen the positive aspects of both.

## 3. AGA-SVM prognostics model

The fusion method used in the paper is based on support vector machines and adaptive genetic algorithms. The adaptive genetic algorithm support vector machine combines adaptive genetic algorithms with support vector machines as described in the following. The original sequence is first transformed into a new sequence of data using an accumulation operation, and the prediction model is established using the support vector machines to generate the data sequence. Then, an adaptive genetic algorithm method is employed to select the best parameters for the adaptive genetic algorithm support vector machine prediction model. Finally, a prediction value can be determined using the inverse accumulation generation prediction result. The essence of the adaptive genetic algorithm support vector machine model is as shown in Fig. 4. The adaptive genetic algorithms use a global automatic optimization ability which intelligently finds the best parameters as well as the optimal parameters for the support vector machine's kernel function, all of which makes the calculations easier and saves prediction time. The complete process is as in the following:

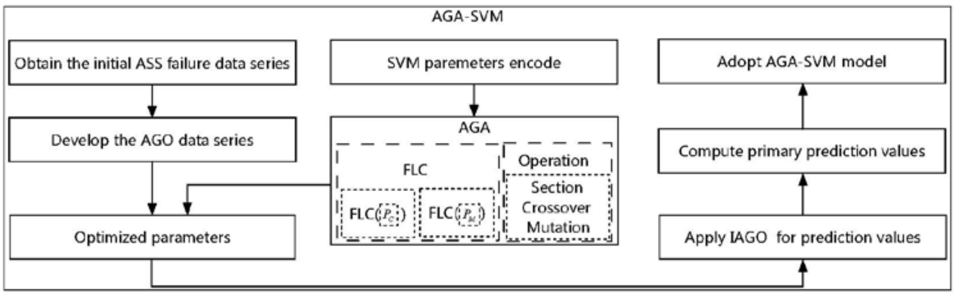


*Fig.4. AGA-SVM operating flow diagram*

**Step 1. Initial data pre-processing of spacecraft software system failure data**

The initial collected spacecraft software system failure data is expressed as the $R^{(0)}$ series:

$$R^{(0)} = \{r^{(0)}(1), r^{(0)}(2), \cdots, r^{(0)}(n)\} \tag{1}$$

where $r^{(0)}(i) > 0 (i = 1, 2, \cdots, n)$ denotes the $i$th failure values.

From the regular accumulation of the initial data series, a data series is generated:

$$R^{(1)} = \{r^{(1)}(1), r^{(1)}(2), \cdots, r^{(1)}(n)\} \tag{2}$$

Where $r^{(1)}(k) = \sum_{i=1}^{k} r^{(0)}(i)$ , $i = 1,2,\cdots,n, k = 1,2,\cdots,n$ . The new series is taken as a learning sample for the support vector machines.

The failure data is random and disordered because of the complexity of the spacecraft software system and the complicated relationship between each module or component. Consequently, the accumulated generation operation is employed on the original disordered data to find the hidden internal relationships. To do this, the adaptive genetic algorithm support vector machine prediction model is established using the generated data.

**Step 2. Selection of kernel function for the prognostics model**

Different kernel functions and parameters have a significant effect on the support vector machine prediction model's performance. Selecting the appropriate kernel function can be done relatively easily from the predicted results, which can, to some extent, overcome the negative effects caused by the unbalanced samples. Common kernel functions have polynomial kernel functions, and radial basis function (RBF) kernel functions, and each different kernel function determines a different nonlinear transformation and feature space, which have different classification effects.

The common kernel functions are as follows:

(1) Inner product kernel function, $k(r_i, r) = (r_i \cdot r)$ ;

(2) Polynomial kernel function, $k(r_i, r) = [(r_i \cdot r) + 1]^q$ ;

(3) RBF kernel function, $k(r_i, r) = \exp\{-|r_i - r|^2 / 2\sigma^2\}$ ;

(4) Sigmoid kernel function, $k(r_i, r) = \tanh(a(r \cdot r_i) + b)$ .

After a comparative analysis of the different kernel functions, and by taking into account the complexity of the spacecraft software system and the large amount of data from the numerous sensors, the RBF kernel function is chosen here to support the support vector machine prognostics model, as it has a strong nonlinear prediction ability, which is able to achieve better prediction results. The parameter $\sigma$ is chosen as in the following step.

**Step 3. Select parameters using adaptive genetic algorithms**

Support vector machine parameter selection, such as the kernel function parameter $\sigma$, the regularization parameter $C$ and the regression approximation error control parameter $\varepsilon$ is very important, as it has a significant influence on support vector machine performance.

There has been significant research focused on support vector machine parameter selection. An expression for calculating $C$ and ε was suggested by Cherkassky and Ma, who also provided an effective solution to the selection problem [6]. Cristianini et al. used a kernel calibration method to quickly determine the kernel parameters, but the selection of $C$ and $\sigma$ were not involved [5]. Keerthi and Lin found that there was a functional relationship between the kernel parameters and $C$ , and converted a two-dimensional optimization problem into two one-dimensional optimization problems [14].

Genetic algorithms (GA) have also been used to select optimal parameters [21]. One of the main problems related to genetic algorithms is in finding the optimal control parameter values. Further, different control parameter values may be necessary during the course of a run. The main weaknesses of genetic algorithms are that they can be ineffective and time-consuming because of complexity, and therefore can be costly to the SSS. Consequently, an adaptive genetic algorithm is built so that the selected control parameters can be dynamically adjusted during the problem solution evolution [7]. Reference [20] described the main scheme of this concept using two fuzzy logic controls (FLC): the crossover FLC and the mutation FLC. These two FLCs were implemented independently to adaptively regulate the crossover and mutation operator rates during the genetic search process. The fitness evaluation function in this paper is defined as $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{R - \hat{R}}{t}\right|$, where $R$ , $\hat{R}$ represent the initial values and prediction values [8].

**Step 4. Adopt support vector machine regression model**

$R^{(1)} = \{r^{(1)}(1), r^{(1)}(2), \cdots, r^{(1)}(n)\}$ is the given generated data series, where $r_t$ can be used to predict $r_{t+1}$ by mapping $f: D^m \to D$ , $r_{t+1} = f(r_t, r_{t-1}, \cdots, r_{t-(m-1)})$ , and $m$ is the embedded dimension, namely the model order. Consequently, the learning samples for prediction can be obtained after the transformation. Then, the final prediction error (FPE) is employed to assess the model error and to select the value for $m$ .
where:

$$\mathrm{FPE}(m) = \frac{d+m}{d-m}\sigma_a^2 \tag{3}$$

$\sigma^2 = E(\mathrm{a}_d) = \frac{1}{d-m}\sum_{t=m+1}^{r}[\mathrm{d}_t - (\sum_{i=1}^{d-m}(\alpha_i - \alpha_i^*)K(\mathrm{r}_i, \mathrm{r}_t) + \mathrm{b})]^2$ , $d$ is the number of training samples, $\alpha$ and $\alpha^*$ are the Lagrange multipliers, and $K$ is the inner product function.

After the topological structure of the support vector machine prediction is determined, the learning samples that the support vector machines use are trained, and the values for $\alpha$ , $\alpha^*$ and b are derived.

From this, the regression function [33] is available:

$$f(r) = \sum_{SV}(\alpha_i - \alpha_i^*)K(r_i, r) + b \tag{4}$$

where $t = m+1, \cdots, d$ .

Accordingly, the values for $\alpha$ , $\alpha^*$ and $b$ are put into Eq. (4) and a definitive regression function is determined.

**Step 5. Compute prediction values**

The data series $R^{(1)}$ is put into the prediction steps above and $\hat{R}^{(1)}$ is determined. An L-step prediction model is then computed:

$$\hat{r}_{d+1} = \sum_{i=1}^{d-m}(\alpha_i - \alpha_i^*)K(r_i, r_{d-m+l}) + b \tag{5}$$

where $r_{d-m+l} = \{r_{d-m+l}, \cdots, \hat{r}_{d+1}, \cdots, \hat{r}_{d+l-1}\}$ .

The data series $\hat{R}^{(1)}$ in Eq. (5) are the prediction values for the accumulated generation data series $R^{(1)}$ .
The inverse accumulated generation operation (IAGO) to $\hat{R}^{(1)}$ is initiated, and the prediction model for the original data series $R^{(0)}$ is obtained as follows:

$$\hat{R}^{(0)}(k+1) = \hat{r}^{(1)}(k+1) - r^{(1)}(k), k = n+1, n+2, \cdots \tag{6}$$

Table 1. 100 historical $FST_t$ data (seconds)

| No. | Value | No. | Value. | No. | Value | No. | Value | No. | Value | No. | Value |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8.63 | 18 | 9.38 | 35 | 9.49 | 52 | 12.61 | 69 | 12.28 | 86 | 11.38 |
| 2 | 9.15 | 19 | 8.61 | 36 | 8.13 | 53 | 7.16 | 70 | 11.96 | 87 | 12.21 |
| 3 | 7.96 | 20 | 8.78 | 37 | 8.68 | 54 | 10.01 | 71 | 12.02 | 88 | 12.28 |
| 4 | 8.64 | 21 | 8.04 | 38 | 6.46 | 55 | 9.86 | 72 | 9.30 | 89 | 11.37 |
| 5 | 9.98 | 22 | 10.91 | 39 | 8.01 | 56 | 7.87 | 73 | 12.50 | 90 | 11.41 |
| 6 | 10.19 | 23 | 7.56 | 40 | 4.71 | 57 | 8.64 | 74 | 14.56 | 91 | 14.42 |
| 7 | 11.76 | 24 | 11.04 | 41 | 10.01 | 58 | 10.58 | 75 | 13.33 | 92 | 8.34 |
| 8 | 11.67 | 25 | 10.12 | 42 | 11.02 | 59 | 10.93 | 76 | 8.95 | 93 | 8.08 |
| 9 | 6.94 | 26 | 10.18 | 43 | 10.87 | 60 | 10.67 | 77 | 14.78 | 94 | 12.21 |
| 10 | 7.49 | 27 | 5.92 | 44 | 9.48 | 61 | 12.51 | 78 | 14.89 | 95 | 12.79 |
| 11 | 10.63 | 28 | 9.50 | 45 | 11.03 | 62 | 11.37 | 79 | 12.14 | 96 | 13.16 |
| 12 | 7.86 | 29 | 9.62 | 46 | 10.86 | 63 | 11.92 | 80 | 9.79 | 97 | 12.76 |
| 13 | 8.69 | 30 | 10.43 | 47 | 9.48 | 64 | 9.58 | 81 | 12.11 | 98 | 10.36 |
| 14 | 9.29 | 31 | 10.64 | 48 | 6.67 | 65 | 10.46 | 82 | 13.12 | 99 | 13.85 |
| 15 | 8.35 | 32 | 8.34 | 49 | 9.31 | 66 | 12.73 | 83 | 12.30 | 100 | 12.49 |
| 16 | 9.11 | 33 | 10.39 | 50 | 10.36 | 67 | 12.61 | 84 | 12.72 | | |
| 17 | 9.61 | 34 | 11.32 | 51 | 10.11 | 68 | 12.10 | 85 | 14.21 | | |

Table 2. Samples in validation set (seconds)

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Value | 12.02 | 9.30 | 12.50 | 14.56 | 13.33 | 8.95 | 14.78 | 14.89 | 12.14 | 9.79 |
| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Value | 12.11 | 13.12 | 12.30 | 12.72 | 14.21 | 11.38 | 12.21 | 12.28 | 11.37 | 11.41 |
| No. | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Value | 14.42 | 8.34 | 8.08 | 12.21 | 12.79 | 13.16 | 12.76 | 10.36 | 13.85 | 12.49 |

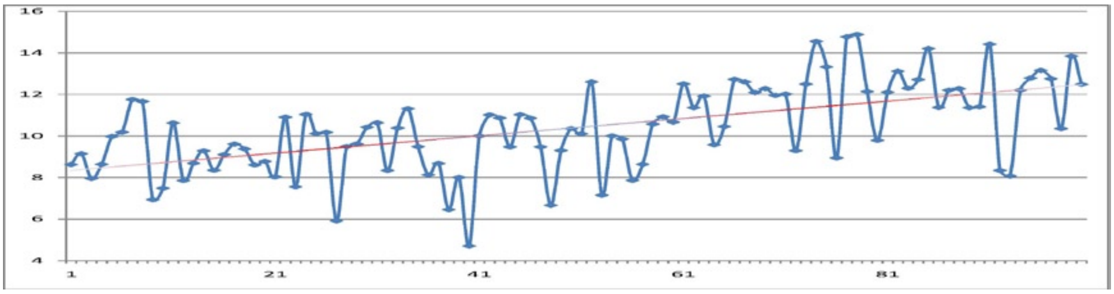


Fig. 5. Historical FST data Trends

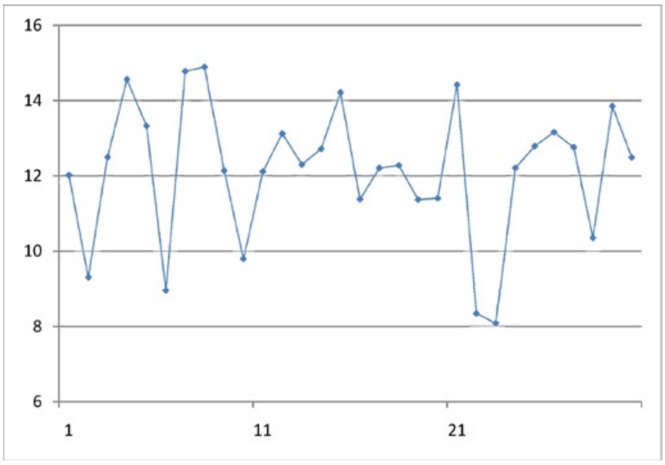


Fig. 6: Validation set Samples Trend

where $\hat{R}^{(0)}$ are the prediction values for $R^{(0)}$.

## 4. Numerical example

In this section, a numerical example is given to illustrate the predictive performance of the proposed adaptive genetic algorithm support vector machine model. As described in Section 2, as the SSSR generally changes over time, the FST is an exemplification of an SSSR for an SSS and is closely connected with defects and faults. Consequently, the FST for the SSS are taken as the prognostic samples as shown in Table 1. The data contains 100 observations of a times series (t, $FST_t$) pertaining to an SSS prototype. Here, $FST_t$ represents the failure space time of the software after the tth modification has been made. Then, the $FST_t$ data are divided into two with seventy items in the training set, and thirty items in the validation set as shown in Figures 5 and 6. Through calculation, 30 $FST_t$ values were derived as shown in Table 2.

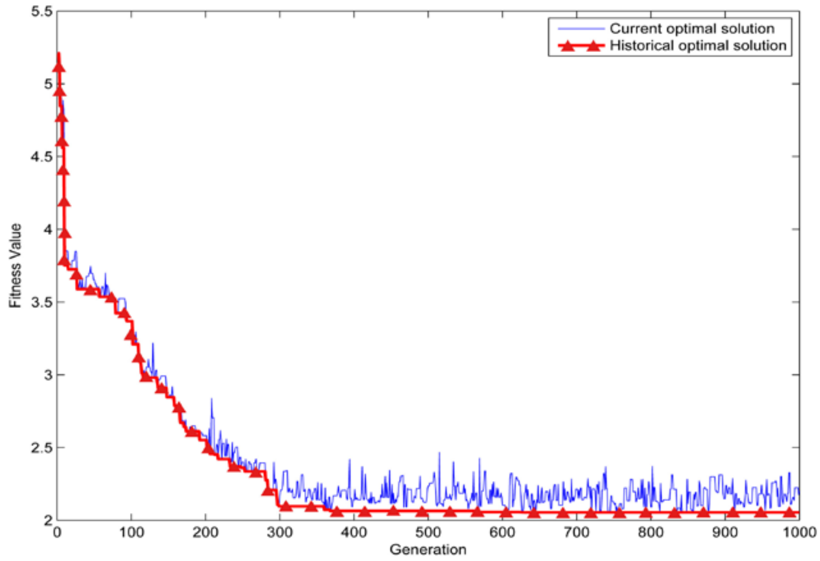


*Fig. 7. The comparison of the current optimal solution and the historical optimal solution*

Table 3. Predicted values using different approaches

| t | Orignal | AGA–SVM | ANN | GA-SVM |
|---|---|---|---|---|
| 1 | 12.02 | 12.81 | 13.68 | 13.96 |
| 2 | 9.30 | 10.80 | 8.23 | 7.64 |
| 3 | 12.50 | 12.78 | 10.45 | 14.28 |
| 4 | 14.56 | 13.99 | 15.72 | 15.73 |
| 5 | 13.33 | 13.67 | 14.26 | 10.67 |
| 6 | 8.95 | 9.65 | 8.46 | 8.12 |
| 7 | 14.78 | 14.01 | 12.46 | 12.46 |
| 8 | 14.89 | 15.71 | 15.81 | 15.36 |
| 9 | 12.14 | 12.55 | 11.64 | 9.43 |
| 10 | 9.79 | 9.7 | 9.01 | 8.19 |
| 11 | 12.11 | 11.11 | 13.24 | 14.56 |
| 12 | 13.12 | 13.68 | 14.68 | 10.38 |
| 13 | 12.30 | 12.23 | 10.37 | 11.78 |
| 14 | 12.72 | 12.45 | 13.01 | 10.39 |
| 15 | 14.21 | 14.38 | 13.83 | 12.41 |
| 16 | 11.38 | 11.64 | 10.65 | 8.06 |
| 17 | 12.21 | 12.89 | 13.99 | 14.67 |
| 18 | 12.28 | 13.01 | 12.67 | 11.29 |
| 19 | 11.37 | 10.34 | 10.94 | 13.79 |
| 20 | 11.41 | 11.63 | 11.38 | 12.08 |
| 21 | 14.42 | 12.45 | 15.06 | 10.49 |
| 22 | 8.34 | 9.34 | 8.96 | 11.94 |
| 23 | 8.08 | 8.09 | 7.68 | 9.05 |
| 24 | 12.21 | 11.79 | 13.54 | 11.52 |
| 25 | 12.79 | 13.11 | 12.03 | 8.42 |
| 26 | 13.16 | 12.42 | 12.88 | 15.28 |
| 27 | 12.76 | 10.43 | 11.06 | 12.48 |
| 28 | 10.36 | 8.34 | 9.25 | 7.49 |
| 29 | 13.85 | 13.65 | 14.78 | 14.27 |
| 30 | 12.49 | 12.78 | 13.6 | 13.53 |

### 4.1. Results of fusion prognostics

In the adaptive genetic algorithm support vector machine, the 100 historical values were divided into two parts: seventy in the training set, and thirty in the validation set. The adaptive genetic algorithms were employed to solve the problems, and the parameters used were set according to reference [20] as follows: population size $N = 50$, crossover probability $p_c = 0.6$, mutation probability $p_m = 0.2$, maximum number of generations $\max Gen = 1000$. In addition, the fitness evaluation function was defined as $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{R-\hat{R}}{t}\right|$, as outlined in Step 3 of Section 3. After a run of the adaptive genetic algorithms, the fitness function had a significantly better convergence as shown in Fig.7, with the best fitness value for each generation represented in the vertical axis, and the iterations in the horizontal axis. Consequently, the optimal parameters $\sigma = 0.66$, $\varepsilon = 0.0001$, $C = 1000.26$ were chosen. The FPE method was used for prediction after the optimal length of the historical original data had been selected. The adaptive genetic algorithm support vector machine updated the historical original data using an adaptive add or subtract strategy, and accordingly, dynamic prediction was executed.

Considering the dynamic prediction latency, this model adopts a multi-step prediction strategy, with 5 prediction steps. The schemes cited above not only assure a highly precise prediction, but also reduce the amount of calculations needed.

The predicted values using the adaptive genetic algorithm support vector machine, genetic algorithm support vector machine, and an artificial neural network (ANN) [11] are shown in Table 3. The thirty samples in the validation set were used for comparison and evaluation.

### 4.2. Performance analysis

A comparative study of the predictive performance of the other models was conducted. The results from the adaptive genetic algorithm support vector machine model were compared with the results from an artificial neural network model and a genetic algorithm support vector machine model as shown in Fig. 8. The vertical axis represents the value of the failure space time, and the horizontal axis represents the sequence number. It can be seen that the artificial neural network model results had a serious distortion in the prediction for temporal data aggregation, as it is only able to predict exponential data series. However, the adaptive genetic algorithm support vector machine avoided this problem, as it enhances the regularity of data through the accumulated generating operation which weakens the random disturbance on the original data and finds the hidden internal relations from the disordered original data. In addition, the advantages of the support vector machines are also incarnated in the genetic algorithm support vector machine, such as the small sample learning. Consequently, the adaptive genetic algorithm support vector machine was shown to have a better predictive performance than either the genetic algorithm support vector machine or the artificial neural network.

Based on the prognostics results, the prediction model was analyzed and evaluated using the following measures; the mean absolute percentage error (MAPE), the normalized root mean squared errors (NRMSE) and the root mean squared relative error (RMSRE). MAPE can be used to analyze and evaluate the approximation ability of the prediction model, and NRMSE and RMSRE are used to assess the

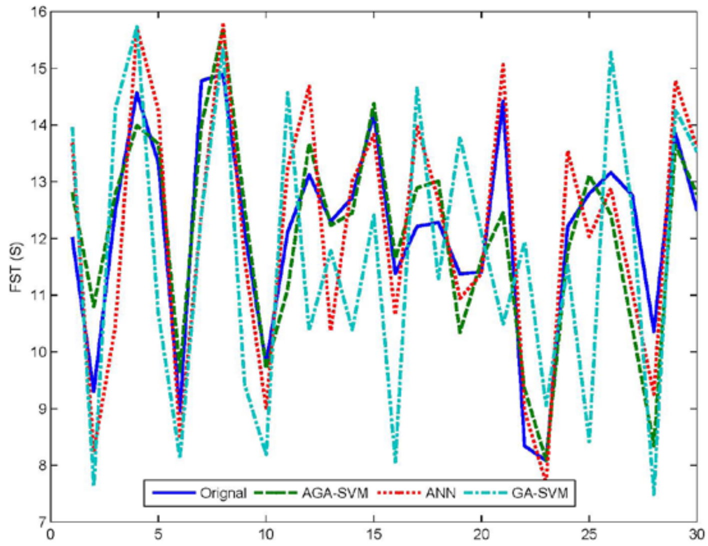


*Fig. 8. Comparison of predicted results for different methods*

*Table 4. Efficacy evaluation of prognostics models*

| Prognostics models | Evaluation index | | |
|---|---|---|---|
| | MAPE | NRMSE | RMSRE |
| ANN | 43.104 | 0.264 | 0.618 |
| GA-SVM | 22.146 | 0.582 | 0.326 |
| AGA−SVM | 10.601 | 0.791 | 0.041 |

ability of the models to simulate realistic, observed variability. The evaluation results for the prediction models are shown in Table 4, from where it can be seen that: $ANN_{(MAPE)}>GA\text{−}SVM_{(MAPE)}>AGA\text{-}SVM_{(MAPE)}$, $ANN_{(NRMSE)}<GA\text{-}SVM_{(NRMSE)}<AGA\text{-}SVM_{(RMSRE)}$, $ANN_{(RMSRE)}>GA\text{-}SVM_{(RMSRE)}>AGA\text{-}SVM_{(RMSRE)}$, which demonstrates that the AGA-SVM had the best performance for the MAPE, NRMSE and RMSRE. These results not only show the close relationship between the actual and predicted values, but also indicate that the AGA-SVM model is valid for predicting the SSSR.

## 5. Conclusion

An integrated system health management for a spacecraft software system was proposed to assist in avoiding catastrophic software failure by providing ongoing reliability monitoring as well as by predicting failure and providing warnings. Spacecraft software system reliability prediction is a critical process in integrated system health management. Many methods have been used to predict spacecraft software system reliability, but a single prediction method is unable to meet the requirements of modern complex space avionics systems. In this paper, a prognostics model for the prediction of spacecraft software system reliability was demonstrated. The adaptive genetic algorithm support vector machine combines adaptive genetic algorithms with standard support vector machines. A numerical example demonstrated that the use of the adaptive genetic algorithm support vector machine model in selecting the support vector machine parameters increases the predictive performance of the genetic algorithm support vector machine, as it makes the prediction process faster, which is very important for integrated system health management. A comparative study of the predictive performance of other models was conducted, and from this it could be seen that the proposed model has a better performance than either an artificial neural network or standard support vector machines. Therefore, the model was proved to be an efficient reliable modeling technique for spacecraft software system engineering.

In this paper, the focus was on finding methods to effectively use reliable system-specific information and improve prognostics performance. In the future, we plan to study investigative techniques that can fuse spacecraft software system reliability estimates, save time and provide accurate, more efficient optimization algorithms to select the parameters for the support vector machine model to predict spacecraft software system reliability.

## References


1. AM Aziz, An iterative method for decision fusion in multiple sensor systems. Aerospace Science and Technology 2010; 14: 487-493.
2. Amin A, Grunske L, A Colman, An approach to software reliability prediction based on time series modeling. Journal of Systems and Software 2013; 86(7): 1923-1932.
3. Bastière A, Fusion methods for multi-sensor classification of airborne targets Méthodes de fusion pour la classification multi-senseurs de cibles aériennes. Aerospace Science and Technology 1997; 1: 83-94.
4. Chu D, Deshpande A, Hellerstein JM, Hong W, Approximate data collection in sensor networks using probabilistic models, in proceedings of the 22nd international conference on data engineering, Atlanta 2006: 48-59.
5. Cristianini N, Kandola J, A Elisseeff J Shawe-Taylor, On kernel target alignment, in: Studies in Fuzziness and Soft Computing 2006; 194: 205-256.
6. Cherkassky V, Ma Y, Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 2004; 17: 113-126.
7. Chamnanlor C, Sethanan K, Chien CF, Gen M, Re-entrant flow shop scheduling problem with time windows using hybrid genetic algorithm based on auto-tuning strategy. International Journal of Production Research 2013: 1-18.
8. Ding C, Xu J, Xu L, ISHM-based intelligent fusion prognostics for space avionics. Aerospace Science and Technology 2013; 29(1): 200-205.
9. Figueroa F, Holland R, Schmalzel J. Duncavage, D. Integrated system health management (ISHM): systematic capability implementation. In Sensors Applications Symposium, 2006. Proceedings of the 2006 IEEE: 202-206.
10. Garg M, Lai R, Huang SJ, Colman, When to stop testing: a study from the perspective of software reliability models. IET software 2011; 5(3): 263-273.
11. Guo Y, Wang X, Liu C, Zheng Y, Cai X, Electronic System Fault Diagnosis with Optimized Multi-kernel SVM by Improved CPSO. Eksploatacja i Niezawodnosc-Maintenance and Reliability 2014; 16(1):85-91.
12. Hsu CJ, Huang CY, An adaptive reliability analysis using path testing for complex component-based software systems. Reliability, IEEE Transactions on 2011; 60(1): 158-170.

13. Huang CY, Lin CT, Analysis of software reliability modeling considering testing compression factor and failure-to-fault relationship. Computers, IEEE Transactions on 2010 59(2): 283-288.
14. Keerthi SS, Lin CJ, Asymptotic behavior of support vector machines with Gaussian Kernel. Neural Computation 2003; 15: 1667-1689.
15. Liu X, Li W, Wang Y. Manned space flight launch vehicle CZ-2F rliability growth management. In Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on. IEEE. 378-381.
16. Musa JD, A theory of software reliability and its application. Software Engineering, IEEE Transactions on 1975; 3: 312-327.
17. Martin L, Schatalov M, Hagner M, Goltz U, O Maibaum, A methodology for model-based development and automated verification of software for aerospace systems. In Aerospace Conference 2013 IEEE: 1-19.
18. Michael R, Lyu. Handbook of Software Reliability Engineering: Introduction. IEEE Computer Society Press and McGraw-Hill Book Company, 2005.
19. NASA. Remarks by the president on space exploration in the 21st century, http://www.nasa.gov/news/media/trans/obama_ksc_trans.html.
20. Neungmatcha W, Sethanan K, Gen M, Theerakulpisut S, Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem. Computers and Electronics in Agriculture 2013; 98: 85-99.
21. Peng W, Huang, HZ, Zhang X, Liu Y, Li Y, Reliability based optimal preventive maintenance policy of series-parallel systems. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2009: 4–7.
22. Pietrantuono R, Russo S, Trivedi KS, Software reliability and testing time allocation: an architecture-based approach. Software Engineering, IEEE Transactions on 2010; 36(3): 323-337.
23. Qi F. China's manned space development reviews and future scenarios. Spacecr Environ Eng 2011; 28: 1–4 (in Chinese).
24. Sun S, Multi-sensor optimal information fusion Kalman filters with applications. Aerospace Science and Technology 2004; 8: 57-62.
25. Sim L, Cummings ML, Smith CA. Past, present and future implications of human supervisory control in space missions. Acta Astronautica 2008; 62(10): 648-655.
26. Vichare N, Pecht M, Prognostics and health management of electronics. IEEE Transaction on Components and Packaging Technologies 2006; 29: 222-229.
27. Wang G, Cui Y, Wang S, Meng X, Design and performance test of spacecraft test and operation software. Acta Astronautica 2011; 68(11): 1774-1781.
28. Wang Z, Huang HZ, Du X, Reliability-based design incorporating several maintenance policies. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2009; (4): 37-44.
29. Xu J, Xu L, Integrated system health management-based condition assessment for manned spacecraft avionics. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 2013; 227(1): 19-32.
30. Xu J, Xu L, Health management based on fusion prognostics for avionics systems, Journal of Systems Engineering and Electronics 2011; 22: 428-436.
31. Yu T, Cui W, Song B, Wang S, Reliability growth estimation for unmanned aerial vechicle during flight-testing phases. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2010; (2): 43-47.
32. YG Bagul, I Zeid, SV Kamarthi, A framework for prognostics and health management of electronic systems. IEEE Aerospace Conference 2008.
33. Zhang JF, Hu SS, Chaotic time series prediction based on multi-kernel learning support vector regression, Acta Phys. Sin 2008; 57: 2708-2713.



**Jiuping XU**
**Zhiyi MENG**
Department of Uncertainty Decision-Making Laboratory
Sichuan University, No.24 South Section 1, Yihuan Road
Chengdu, China, 610065

**Lei XU**
Department of Business Management
Xihua University
Chengdu, China, 610039

E-mails: xujiuping@scu.edu.cn, zhiyimengscu@sina.com, leihsu@163.com