

# Forecasting Satellite Attitude Volatility Using Support Vector Regression with Particle Swarm Optimization

Zuhua Zhong, Dechang Pi

**Abstract**—Forecasting the volatility of satellite attitude is a meaningful but complicated problem due to the complex non-linear characteristics of the standard deviation series, which reflects the volatility of satellite attitude. Support Vector Regression (SVR) is an efficient machine learning technique derived from statistical learning theory and has been successfully employed to solve regression problem of time series with nonlinearity in recent years. However, the generalization capacity of SVRs is greatly depend on their hyper-parameters and the process of tuning parameters manually is time-consuming. Particle Swarm Optimization (PSO) is a simple but effective optimization method inspired by social behavior of organisms such as bird flocking and fishing schooling. Thus, this paper proposes a hybrid PSO-SVR model to predict the volatility of these three attitude angles in satellites: Pitch Angle (PA), Roll Angle (RA) and Yaw Angle (YA), respectively. Thereinto, PSO is exploited to seek the optimal parameters for SVR to achieve satisfactory generalization capacity. The standard deviation series generated from telemetry data of Attitude Control System belonging to a Chinese satellite was used as experimental data to testify the performance of our proposed PSO-SVR model. The experimental results indicate that the hybrid PSO-SVR model can be a promising alternative to forecast the volatility of satellite attitude with relative high accuracy compared with grey model, residual grey model, and BP neural network.

**Index Terms**—Satellite Attitude, Volatility, Support Vector Regression (SVR), Particle Swarm Optimization (PSO)

## I. INTRODUCTION

Satellite attitude refers to the pointing direction of satellites flying in the orbit and the stabilization control of the attitude plays a crucial role in guaranteeing the successful operation of satellites [1]. In order to complete their missions, satellites have to meet various specified requirements, including these concerning the flying attitude. For instance, the antenna of telecommunication satellites should point at the earth all the time, whereas what earth observing satellites need to do is regulating the windows of their monitoring

equipment towards the earth all along. Thus, abnormal flying attitude will certainly interrupt the regular process fulfilling their tasks. According to statistics, from 1993 to 2013, there are approximately 300 catastrophic accidents or temporary malfunctions in total, resulting from different reasons, mainly including hitting by anomaly, solar array circuit failures, losing contact, power system failures, and attitude exceptions, etc. Among them, the attitude exceptions (pointing in wrong directions or fluctuating fiercely) contribute to an important part of them. (<http://www.sat-nd.com/failures/>). How to avoid satellite accidents via regulating the attitude is a challenging topic worth studying for researchers.

The stabilization control of satellite attitude is conducted by a vital subsystem named Attitude Control System (ACS). One of the commonly used implementation schemes of ACS is the three-axis stabilization scheme due to its extensive applicability and high pointing precision. The satellite attitude discussed in this paper refers to Pitch Angle (PA), Roll Angle (RA), and Yaw Angle (YA) in the three-axis stabilization scheme. In the current competitive context of aviation industry, intense attention involving satellite attitude has been mainly given to developing techniques for designing attitude control schemes [2]-[4], attitude determination or estimation [5], [6], and ACS fault diagnosis [7]-[9]. Great achievements have been obtained from these studies. However, previous researches have failed to throw light on the analysis of the inherent regularity of historical data and prevent latent satellite failures concerning attitude exceptions in advance. Interruption caused by attitude exceptions still cannot be effectively decreased.

For the prevention of malfunctioning processes, predicting the future values of key parameters is the first crucial procedure and trying to regulate the questionable component if predicted values exceed the given range is another important one. Time series prediction in which future parameter values are predicted as a function of the values in the past is an efficient approach to study the behavior of key parameters [10]. According to satellite-related knowledge, a lot of real-time telemetry attitude information, including PA, RA, and YA, is generated during the in-orbit monitoring and managing process and was stored in a huge database as time series for future analysis. It is well known that extensive researches concerning predicting the volatility of stock-market [11] are reported in the literature, as the volatility is able to reflect the stabilization level of stock-market. Predicting it in advance can provide some

Manuscript received July 21, 2014; revised August 10, 2014. This work was supported in part by the Fundamental Research Fund for the Central Universities (NZ2013306), the 333 project of Jiangsu Province, the technology Foundation of China (JSJC2013605C009).

Z. H. Zhong is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016 China. (E-mail: zzhnuaa@163.com).

D. C. Pi is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016 China. (E-mail: dc.pi@nuaa.edu.cn).

assistance for decision making for stock-market participant. As there also exists specific relationship between the volatility and the state of satellites according to expert knowledge and statistical analysis (see Fig. 1), we can regard the volatility of satellite attitude as the key parameter similarly. Generally speaking, the volatility of three-axis attitude usually belong to a given range when satellites running regularly. As shown in Fig. 1, the value of PA volatility under attitude exception situation is much larger than that in the normal situation. Therefore, if we forecast the volatility of satellite attitude according to recent historical data, its developing trend and incipient attitude failures will be found early if the predicted future value of volatility exceeds a specified threshold. Impact caused by attitude exceptions can be greatly controlled. In this paper, we take standard deviation as the indicator representing the volatility according to knowledge on statistic and we regard them as equivalent terms in the following paper.

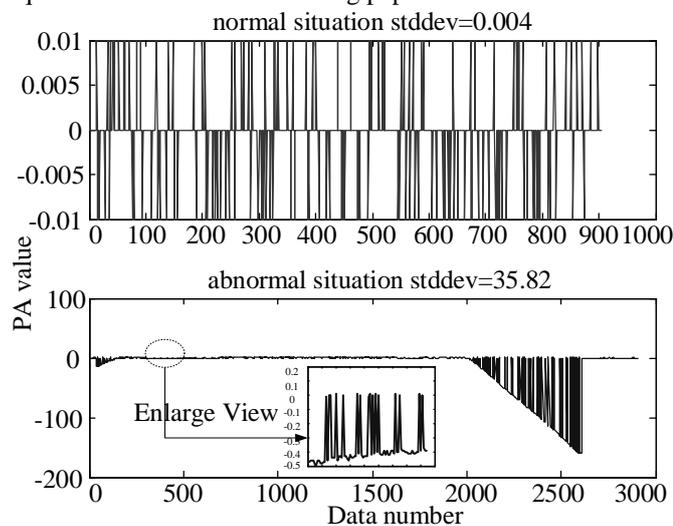


Fig. 1. Volatility comparison between normal and abnormal situation.

In recent decades, time series prediction technology has been developed into two broad categories: traditional linear time series prediction and nonlinear time series prediction. The grey model [12] (GM) proposed by Professor Deng Julong, as a typical linear model, has been widely employed in short-term forecasting of monotonously increasing or decreasing data [13]. But their predicting accuracy will be greatly reduced if the objective data is highly nonlinear. Thus, it is not suitable to be applied in predicting the volatility of satellite attitude in consideration of the complex non-linear characteristics of the volatility series. Besides, various effort has been devoted into studying the artificial neural networks [14] (ANN) for that it can effectively solve function estimation problem with nonlinearity [15]. However, the *empirical risk minimization principle* followed by ANN make ANNs fail to get rid of some inherent deficiencies, e.g., the danger of over-fitting, the probability of getting stuck in local optimal, and slow convergence velocity. Support vector machine (SVM) is a novel and efficient machine learning technique proposed by Vapnik in 1995 [16], [17] originally for classification purposes [18], [19]. The *structural risk minimization (SRM) principle* implemented by SVMs aims to minimize the sum of empirical risk and confidence interval. This principle grants SVMs several merits compared with

ANN, such as obtaining global, unique solution as modelling SVMs is dealing with a linearly constrained quadratic programming problem. Another advantage of this principle is that SVMs can effectively avoid over-fitting because SVMs are able to keep a balance between the empirical risk and confidence interval. Years later, the basic theory of SVMs was extended to support vector regression (SVR) [20], [21] to cope with regression problem and has exhibited desirable performance in time series prediction problem from various domain [22-25]. However, the difficulty in selecting proper SVR hyper-parameters substantially slow down the pace of resolving practical problems utilizing SVRs. What's more, mature theoretical guidance for choosing SVR parameters is still in absence and tuning these parameters manually is time-consuming.

Thus, this paper proposes a hybrid PSO-SVR model to predict the volatility of satellite three-axis attitude in which Particle Swarm Optimization (PSO) [26], [27] is used to select optimal SVR parameters. This forecasting model aims at providing assistance for monitoring the stabilization of satellite attitude. The performance of PSO-SVR is compared with the well-known existing method, GM, Residual GM, and BP neural network in order to demonstrate the superiority of our proposed model. The remaining of this paper is organized as follows: Section 2 provides a brief introduction to support vector regression. Basic theory and algorithm of particle swarm optimization is elaborated in Section 3. Section 4 describes the process of SVR parameters optimization using PSO. Section 5 presents the procedures of employing PSO-SVR model in forecasting the volatility of satellite attitude and testifies the forecasting capability of our proposed PSO-SVR model with real telemetry dataset from an anonymous satellite ACS. Finally, Section 6 draw some conclusions.

## II. SUPPORT VECTOR REGRESSION

Regression problems aim to determine a proper function  $f(x)$  which accurately describes the relationship between input vector  $x$  and output value  $y$ . SVRs regard these models with minimum sum of empirical risk and confidence interval as the optimal ones. The core concept of SVR is firstly to map the original data into a high-dimensional feature space non-linearly, then to find an optimal linear regression function in this feature space [20], [21], see Fig. 2. The following table list the symbols concerning data in this paper.

TABLE I  
NOTATIONS CONCERNING DATA

$G = \{(x_i, y_i)\}$	Dataset
$x_i$	Input vector
$y_i$	Corresponding output
$l$	Total number of data patterns

According to Statistical learning theory, SVRs approximate the regression function taking the following form:

$$f(x) = w \bullet \Phi(x) + b, \quad w \in R^n \quad (1)$$

where  $\Phi$  is the non-linear mapping function,  $w, b, \cdot \cdot \cdot$  denotes the weight vector, bias term, and inner product, respectively. The values of  $w$  and  $b$  are estimated by minimizing the following formula based on Structural Risk Minimization Principle:

$$R_{reg} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l L_{\varepsilon}(x_i, y_i, f) \tag{2}$$

Equation (2) mentioned above describes the regularized risk function. The first term  $\frac{1}{2} \|w\|^2$  is the regularization term which reflects the complexity of the regression solution and corresponds to the confidence interval. While the second part is the empirical risk usually measured by the  $\varepsilon$ -insensitive loss function defined as:

$$L_{\varepsilon}(x_i, y_i, f) = \max(0, |y_i - f(x_i)| - \varepsilon) \tag{3}$$

Besides, the positive constant  $C$  termed penalty factor keeps a balance between empirical risk and confidence interval. The value of  $C$  determines the importance attached to these two items. Increasing the value may lead to pay more attention on the empirical risk, otherwise on the confidence interval. Usually the larger the value of  $C$ , the greater the likelihood of over-fitting. Choosing a suitable value of  $C$  is crucial during the establishment of a favorable SVR model. Equation (3) indicates that the loss will be ignored if the difference between predicted value and actual value is less than  $\varepsilon$ , otherwise the loss equals the absolute difference between the predicting error and the radius of the  $\varepsilon$ -tube [20] shown in Fig. 2.  $\xi_i$  ( $\xi_i^*$ ) termed slack variables, are used to measure the distance between observed value and the upper(lower) boundary of the  $\varepsilon$ -tube (see Fig. 2). Then, (2) can be reorganized as function given by (4):

$$\begin{aligned} \text{Minimize } R_{reg} &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s.t. } &\begin{cases} y_i - f(x_i) \leq \varepsilon + \xi_i \\ f(x_i) - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \tag{4}$$

The minimization of (4) can be solved by exploiting Lagrange theory, the corresponding Lagrange is:

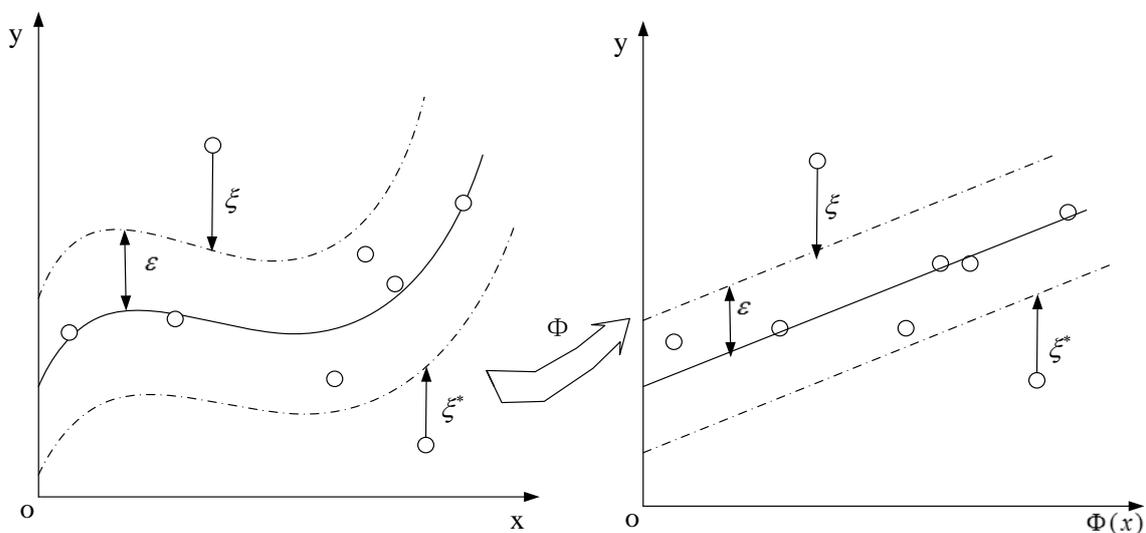


Fig. 2. Left, a nonlinear function in the original space is mapped into the feature space (right) where the function become linear,  $\varepsilon$  denotes the negligible error of SVR and data points located on or outside the tube are so-called support vectors.

$$\begin{aligned} L(w, b, \xi, \xi^*, \alpha, \alpha^*, \gamma, \gamma^*) &= \frac{1}{2} \|w\|^2 \\ &+ C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ &- \sum_{i=1}^l \alpha_i [\xi_i + \varepsilon + y_i - f(x_i)] \\ &- \sum_{i=1}^l \alpha_i^* [\xi_i^* + \varepsilon - y_i + f(x_i)] \\ &- \sum_{i=1}^l (\gamma_i \xi_i + \gamma_i^* \xi_i^*) \end{aligned} \tag{5}$$

where  $\alpha_i, \alpha_i^*, \gamma_i, \gamma_i^*$  are so-called Lagrange multipliers. This quadratic programming problem can be further transformed to an easier handled dual optimization problem, that is:

Maximize

$$\begin{aligned} w(\alpha, \alpha^*) &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \sum_{i=1}^l (\alpha_i + \alpha_i^*) \varepsilon \\ &- \frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \Phi(x_i), \Phi(x_j) \rangle \\ \text{s.t. } &\left. \begin{aligned} \sum_{i=1}^l (\alpha_i - \alpha_i^*) &= 0 \\ 0 \leq \alpha_i &\leq C \\ 0 \leq \alpha_i^* &\leq C \end{aligned} \right\} \end{aligned} \tag{6}$$

Finally, we obtain a global and unique solution, where  $w$  is the sum of product between every training data and  $\alpha_i - \alpha_i^*$ :

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(x_i) \tag{7}$$

Theoretically, these training patterns on the boundary of the  $\varepsilon$ -tube possess certain training error  $e_i = \varepsilon \cdot \text{sign}(\alpha_i - \alpha_i^*)$ , so  $b$  can be computed according to the following formula deriving from Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} b &= y_i - w \cdot \Phi(x_i) - \varepsilon \text{ for } \alpha_i \in (0, C) \\ b &= y_i - w \cdot \Phi(x_i) + \varepsilon \text{ for } \alpha_i^* \in (0, C) \end{aligned} \tag{8}$$

For the sake of stability, we take the average value of all the  $b$  computed from (8) as the eventual value of  $b$ .

Thus, the regression function given by (1) can be transformed into the following explicit form:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (9)$$

In (9),  $K(x, x_i)$  is the so-called kernel function and  $K(x, x_i) = \Phi(x_i) \cdot \Phi(x)$ . Using kernel function enables SVR to handle dot product of high-dimensional feature space in original low-dimensional space without having to know the explicit mapping function or compute the value of  $\Phi(x_i)$  directly. Any function matching Mercer's condition can be regarded as the kernel function. The commonly used kernel functions are shown in TABLE II.

TABLE II  
THREE COMMONLY USED KERNEL FUNCTIONS

Radial basis function (RBF)	$K(x_i, x) = \exp(-\ x_i - x\  / 2\sigma^2)$
polynomial basis function	$K(x_i, x) = (a(x_i \cdot x) + b)^d$
sigmoid function	$K(x_i, x) = \tanh(\gamma(x_i \cdot x) + \nu)$

In highly non-linear spaces, RBF kernel usually achieve more satisfactory performance compared with other mentioned kernels. Moreover, only one free parameter  $\sigma$  having to be determined by users decrease the difficulty in parameter-selection procedure and make SVRs more attractive. Thus, we employ the RBF as kernel function in this work. Note that, according to KKT condition, these data patterns lying on or outside the boundary of the  $\epsilon$ -tube will hold non-zero values of coefficients  $(\alpha_i - \alpha_i^*)$  presented in (7) and they are the so-called support vectors [20]. Obviously, it is the support vectors that give shape to the regression function as the other points keep zero value of  $(\alpha_i - \alpha_i^*)$ . Generally, increasing the value of  $\epsilon$  may lead to fewer support vectors and sparser representation of the solution. But a larger  $\epsilon$  can result in lower fitting accuracy. Therefore,  $\epsilon$  can be regarded as a trade-off between the sparseness of the representation and the fitting efficient [20].

In conclusion, three free parameters have to be tuned for SVRs with RBF kernel, namely,  $C$ ,  $\epsilon$  and  $\sigma$ . Generally speaking, the parameter selection problem devotes to choose the optimal parameter-set which can maximize the SVR generalization performance. The evaluation of the SVRs performance can be fulfilled through the computation of Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and normalized mean square error (NMSE):

$$\text{RMSE} = \sqrt{\frac{1}{l} \sum_{i=1}^l (f(x_i) - y_i)^2} \quad (10)$$

$$\text{MAPE} = \frac{1}{l} \sum_{i=1}^l \left| \frac{f(x_i) - y_i}{y_i} \right| \quad (11)$$

$$\text{NMSE} = \frac{1}{\delta^2 n} \sum_{i=1}^n (y_i - f(x_i))^2 \quad (12)$$

$$\text{where } \delta^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

### III. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) algorithm is an iterative searching method motivated by swarm intelligence [26], [27]. The optimal solution is obtained through the iterative movement of "particles" under the guidance of individual historical knowledge and social intelligence. It has gained worldwide reputation in various optimization problems owing to its excellent efficiency and easy-to-handle virtue.

Each particle is initialized with a position vector and a velocity vector. The current best position experienced by each particle is denoted as  $pbest$ , and the best global position determined by social intelligence is denoted by  $gbest$ . These two positions greatly influence the moving direction of every particle. During the searching process, every particle updates its position and velocity according to the following equations after  $pbest$  and  $gbest$  having been determined:

$$v_{id}(t) = w(t) * v_{id}(t) + c_1 r_1 (pbest_{id}(t) - x_{id}(t)) \quad (13)$$

$$+ c_2 r_2 (gbest_d(t) - x_{id}(t))$$

$$x_{id}(t) = x_{id}(t) + v_{id}(t) \quad (14)$$

where  $t$  represents the current iteration,  $x_{id}$  denotes the position of particle  $i$  on dimension  $d$ , whose value is limited in the range  $[-X_{max}, X_{max}]$ , and  $v_{id}$  is the velocity of particle  $i$  on dimension  $d$ , whose value is limited in the range  $[-V_{max}, V_{max}]$ .  $pbest_{id}(t)$  is the current best known position of particle  $i$  on dimension  $d$  at iteration  $t$  and  $gbest_d(t)$  denotes the global best known position on dimension  $d$  at iteration  $t$ .  $c_1$  and  $c_2$  are acceleration coefficients whose value usually limited in  $[0, 2]$ ,  $r_1$  and  $r_2$  are random numbers regenerated in each iteration with uniform distribute ranged in  $[0, 1]$ .  $w$ , the so-called inertia weight proposed by Shi, denotes the momentum remaining in its present position [27] and it makes a trade-off between the global exploration and local exploitation. Larger value of  $w$  usually leads to better global exploration ability, whereas smaller value of  $w$  will result in better local convergence capacity. The above-mentioned parameters are set according to experiential guidance as follows:  $c_1 = c_2 = 2$ , and the adjustment of  $w$  employs the linearly decreasing weight scheme ranging from 0.9 to 0.4:

$$w(t) = (w_{init} - w_{end}) \cdot (t_{max} - t) / t_{max} + w_{end} \quad (15)$$

In (15),  $t_{max}$  represents the max iterations,  $t$  represents the present iterations.  $w_{init}$  is the initial weight which is set to be 0.9 in this work and  $w_{end}$  is the ending weight which is set to be 0.4. Procedures of searching in the solution-space with PSO is elaborated in Algorithm 1.

---

#### Algorithm 1: Particle Swarm Optimization

---

**Input:** Amount of particle swarm  $P$ , acceleration parameters  $c_1$  and  $c_2$ , maximum iterations  $T$ , initial weight  $w_{init}$ , ending weight  $w_{end}$ , dimension of particle  $d$ .

**Output:** global best-known particle  $gbest$ .

**//step1: Initializing all particles.**

```

1. FOR( $i=1; i \leq P; i++$ )
2.   FOR( $j=1; j \leq d; j++$ )
3.     Initialize  $x_{ij}(0)$  randomly with uniform distribution
       ranged in  $[-X_{max}, X_{max}]$ ;
4.     Initialize  $v_{ij}(0)$  randomly with uniform distribution
       ranged in  $[-V_{max}, V_{max}]$ ;
5.      $pbest_{ij}(0) = x_{ij}(0)$ ;
6.   END FOR
7. END FOR
    $gbest(0) = pbest_i(0)$ ,
   where  $i = \arg \min_i Fitness(pbest_i(0))$ ;

    $w(0) = w_{init}$ ;
// step2: Searching the best particle in solution-space
8. FOR( $t=1; t \leq T; t++$ )
9. //update the  $pbest$  of every particle according to the value
   of Fitness
10.  FOR( $i=1; i \leq P; i++$ )
11.    IF( $Fitness(particle_i) < Fitness(pbest_i(t-1))$ )
12.      FOR( $j=1; j \leq d; j++$ )
13.         $pbest_{ij}(t) = x_{ij}(t)$ ;
14.      END FOR
15.    END IF
16.  END FOR
17. //update the  $gbest$  of the swarm according to the value of
   Fitness
18.  FOR( $i=1; i \leq P; i++$ )
19.    IF( $Fitness(pbest_i(t)) < Fitness(gbest(t-1))$ )
20.      FOR( $j=1; j \leq d; j++$ )
21.         $gbest_j(t) = pbest_{ij}(t)$ ;
22.      END FOR
23.    END IF
24.  END FOR
25. END FOR
26. Update  $w(t)$  according to Eq.(15);
27. //update the velocity and position of each particle
28.  FOR( $i=1; i \leq P; i++$ )
29.    FOR( $j=1; j \leq d; j++$ )
30.      Update  $v_{ij}(t)$  according to Eq. (13);
31.      Update  $x_{ij}(t)$  according to Eq. (14);
32.    END FOR
33.  END FOR
34. Output  $gbest(T)$  as the best solution.

```

#### Instructions:

- (1) The value of Fitness (particle) evaluate the performance of each particle. As it is usually measured by the error of the model using parameters extracting from corresponding particle, the smaller the value of Fitness (particle), the better the particle.
- (2) The end condition of the PSO algorithm can be matching specific accuracy requirement or given maximum iterations. This work adopts the latter.

#### IV. PARAMETERS OPTIMIZATION OF SVR WITH PSO

Choosing proper parameters of SVR is a crucial procedure as values of  $C$ ,  $\varepsilon$  and  $\sigma$  have a great impact on the generalization capacity of SVR. In this study, each particle

contains three dimensions corresponding to  $C$ ,  $\varepsilon$  and  $\sigma$ . Note that, the whole dataset should be divided into two non-overlapping and independent datasets: training set and testing set, among which the former is employed for SVR parameter optimization and model establishment procedures and the latter is utilized as testing dataset to evaluate the model prediction efficiency and robustness. The parameters optimization process with PSO can be mainly summarized into the following steps:

Step 1: Set the PSO parameters and initialize the particle swarm. Firstly, set the PSO parameters mentioned above, including the size of swarm, the dimension of solution-space, the maximum iterations, acceleration coefficients  $c_1$  and  $c_2$ , and inertia weight. Then generate a population of initial particles with random position and velocity. Here we set the penalty factor  $C \in [0.001, 100]$ , the RBF kernel parameter  $\sigma \in [0.1, 10]$ , and the  $\varepsilon$ -insensitive loss function parameter  $\varepsilon \in [0, 0.8]$ , respectively. Choose current position as initial individual  $pbest$  for each particle and set the best  $pbest$  in particle swarm as the initial  $gbest$ .

Step 2: Evaluate fitness. In order to keep a balance between computation cost and effectiveness of parameters optimization, we employed the k-fold cross validation to evaluate the fitness for each particle. In this technique, the training set is randomly divided into k non-intersecting subsets with roughly equivalent number of data patterns. For every set of SVR parameters extracting from corresponding particle, k-1 subsets are selected randomly to be the training set for establishing SVR model, and the performance of this SVR model is measured by calculating RMSE on the remaining one subset according to (10). Repeated this process for k times until each of the k subsets has been used once (only once) as testing subset in turn. Eventually, the fitness value of each particle is estimated by averaging the RMSE value over k-subset.

Step 3: Update  $pbest$  and  $gbest$ . Update the  $pbest$  and  $gbest$  according to the value of fitness. Recalculate the inertia weight according to (15).

Step 4: Calculate the velocity of each particle. Recalculate the velocity of each particle with the current  $pbest$  and  $gbest$  by (13).

Step 5: Update the position of each particle. Update position vector for each particle according to (14).

Step 6: Check stop condition. Repeat the iterative process until matching stopping criteria. Otherwise, go to Step 2.

The above mentioned steps are depicted as an important module in Fig. 3.

#### V. PSO-SVR MODEL IN FORECASTING THE VOLATILITY OF SATELLITE ATTITUDE

Procedures of forecasting the volatility of satellite attitude with our proposed PSO-SVR model are depicted in Fig. 3. The following two subsections present the dataset introduction and data preprocessing procedures.

##### A. Datasets

This study was carried out based on a space project. The experimental dataset in this work is some telemetry data

coming from the ACS of an anonymous satellite in this project. We conducted experiments on the three-axis attitude with telemetry dataset on June 10, 2011. We firstly transform the original data into standard deviation series with equal interval as standard deviation can reflect the volatility of each period. According to expert knowledge and statistical analysis on nine months of data, we consider half an hour as the proper interval. TABLE III was obtained based on statistical results and experts advise.

*B. Data Preprocessing*

This subsection aims to give a brief introduction to the preprocessing procedures including data cleaning, data transformation, reconstruction of standard deviation series, and normalization of reconstructed data patterns. Above mentioned procedures are presented exhaustively as follows:

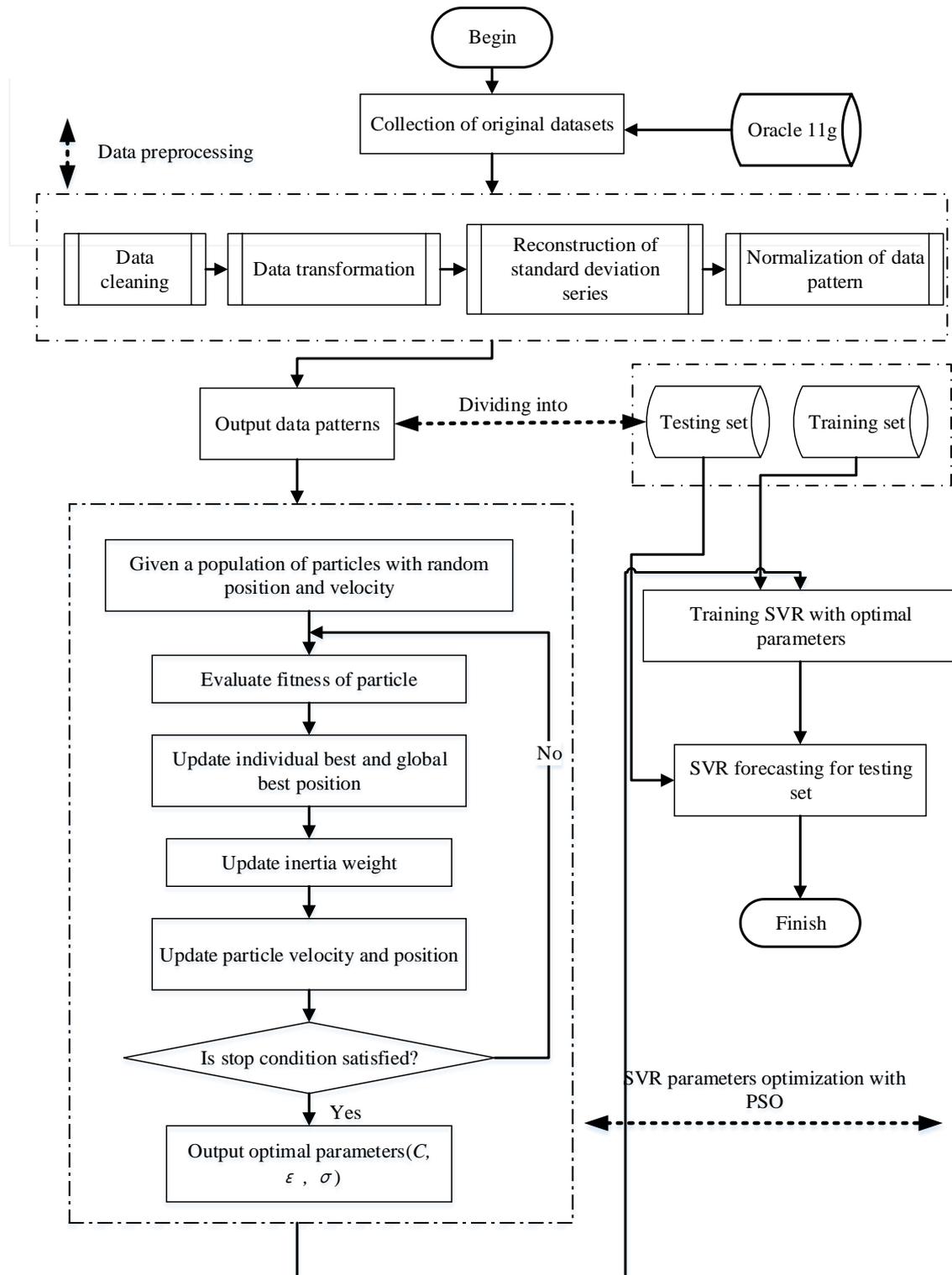


Fig. 3. Overall architecture for forecasting the volatility of satellite attitude using SVR with PSO.

TABLE III  
RELATIONSHIP BETWEEN FLUCTUATING LEVEL AND RANGES OF VOLATILITY

Case	fluctuating level	Ranges of attitudes volatility		
		PA	RA	YA
1#	Relative stable	0-0.01	0-0.005	0-80
2#	Slight unstable	0.01-0.1	0.005-0.1	80-100
3#	Serious unstable	>0.1	>0.1	>100

(1) Data cleaning. The original attitude data is probably contaminated by burst noise, which is named as outlier. Outliers are monstrous or extremely small data frame resulting from decoding errors or transmission failures. This procedure aims at eliminating these outliers as the existence of outliers may influence the volatility.

(2) Data transformation. The primary task of this procedure is to transform the original data into standard deviation series with equal time interval.

(3) Reconstruction of standard deviation series. Time series prediction method based on SVR needs to find a regression function fitting the historical input vector and future output value. The original time series should be transformed into data patterns  $T = \{(X_1, Y_1), \dots, (X_i, Y_i), \dots, (X_{n-m+1}, Y_{n-m+1})\} \in (X \times Y)^{n-m+1}$  firstly, where

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_m \\ x_2 & x_3 & \dots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-m+1} & x_{n-m+2} & \dots & x_n \end{bmatrix}, Y = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{n+1} \end{bmatrix}$$

Each row of the matrix  $X$  represents an input vector, and the similar row in matrix  $Y$  is the corresponding output value.  $m$  is the embedded dimension. Then, the  $n+1$ -th output value can be predicted ahead by regression function described as follows:

$$x_{n+1} = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(X_i, X_{n-m+1}) + b \quad (15)$$

where  $X_i$  refers to the  $i$ th row of matrix  $X$ ,  $X_{n-m+1}$  is the last testing pattern and  $l$  denotes the number of training patterns. In the traditional context of time series prediction, there is no mature theoretical guidance for choosing proper embedded dimension. This work determines the optimal  $m$  according to the MAPE (calculate from (11)) measured on testing set based on the Final Error Minimization Principle. We conducted experiments on PA, RA, YA with the value of  $m$  ranging from 1 to 10, respectively. Fig. 4 shows that the value of  $m$  has an impact on the forecasting performance of SVRs. The three free parameters of SVR are fixed in order to get rid of their influence on the final result. We take the  $m$  which minimized the MAPE on the testing set as the optimal embedded dimension. So the optimal dimensions for PA, RA, and YA are 3, 4, and 7 respectively.

(4) Normalization of reconstructed data patterns. Before the establishment of the SVR models, in order to facilitating the training procedure and improve the predicting performance, the experimental data, including the training patterns and testing pattern generated from above procedure should be scaled to range in  $[0,1]$  based on the following formula:

$$x' = \frac{(x - x_{\min})}{x_{\max} - x_{\min}} \quad (16)$$

where  $x'$  is the normalized value,  $x$  is the original value, and  $x_{\min}$  ( $x_{\max}$ ) is the minimal (maximum) value of corresponding dimension, respectively. Note that the predicted outputs will be remapped to their original value space by the inverse mapping function of (16) before calculating any performance criterion.

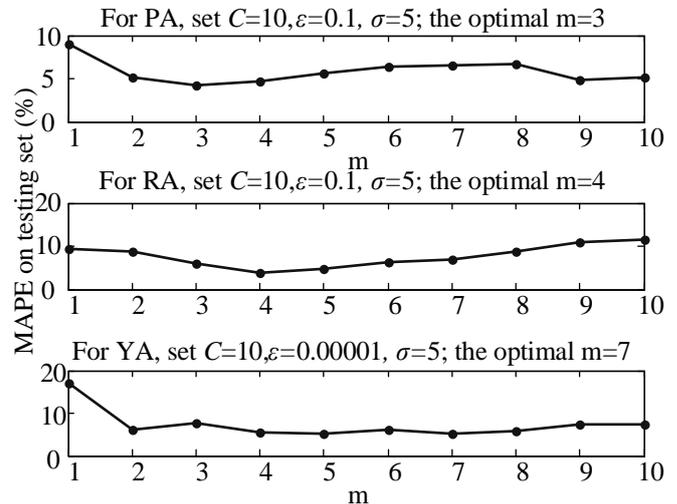


Fig. 4. Effect of embedded dimension  $m$  on forecasting accuracy, SVR parameters was fixed to reduce their influence.

### C. Experimental Result and Discussion

All experiments were conducted on a PC equipped with Intel(R) Core i5-3470M CPU and 4G RAM running on 64-bit Windows 7 Professional Edition. All above mentioned algorithms was implemented in JAVA and the experimental data was stored in Oracle 11g. We recoded the source code of LIBSVM toolbox programmed in JAVA, which was proposed by Chang and Lin and can be downed from: <http://www.csie.ntu.edu.tw/~cjlin/> [28]. As previously discussed, we proposed a PSO-SVR model to forecast the volatility of satellite three-axis attitude. We firstly partitioned the preprocessed data patterns into two disjointing parts with the ratio 90% and 10%, wherein 90% of the older data patterns were used as training set for parameters optimization and predicting model establishment, and the most recent 10% data patterns were used as testing set to evaluate the fitting effectiveness and forecasting capacity. Based on that, we can obtain one-step ahead future prediction and several step-ahead prediction if we establish the sliding window mechanism.

In the training stage, we firstly conducted the optimization process, which aims at obtain optimal parameters for SVR. The fundamental theory and the process of SVR parameters optimization using PSO is elaborated exhaustively in Section 3 and Section 4. We set the size of particles to be 30, the maximum iterations to be 50 and 5-fold cross validation was adopted to evaluating the fitness value of each particle. The obtained optimal parameters of PSO-SVR model and corresponding forecasting accuracy for different attitude

angles are illustrated in TABLE IV. Satisfactory forecasting accuracy can be observed in TABLE IV, especially for YA testing set.

TABLE IV  
FORECASTING ACCURACY AND OPTIMAL PARAMETERS FOR PAO-SVR MODEL

type	m	SVR parameters			Training MAPE /%	Testing MAPE /%
		C	$\epsilon$	$\sigma$		
PA	3	98.8219	1.323E-5	9.2693	1.8607	2.2343
RA	4	31.7201	0.0845	2.2547	1.5201	3.8978
YA	7	89.8469	5.556E-4	1.5052	2.1333	0.791

With the purpose of exhibiting the superiority of PSO-SVR, contrast experiments was conducted on the same dataset using three existing prediction methods: backward propagation neural network (BPNN), GM (1, 1) and Residual GM (1, 1). With regards to BPNN, we taken the standard three-layer BPNN as the benchmark. The number of input layer nodes  $I$  is equal to the dimension of input vector, and the output layer nodes is equal to 1 corresponding to the output vector. Besides, the number of hidden layer nodes is set to be  $2 * I + 1$  according to the Kolmogorov Theorem. So that we established 3-7-1 BP structure for PA, 4-9-1 BP structure for RA, and 7-15-1 BP structure for YA. The learning rate and momentum is set to be 0.01 and 0.9, respectively, as a BP network constructed with these learning parameters may achieve desirable prediction accuracy with relative few epochs [15]. In addition, the number of epochs in this work is set to be 1000 and the sigmoid function is used as transfer function. As for GM, we used the GM (1, 1) and Residual GM (1, 1) as comparison methods, in which the first '1' means only for one dimension series and the second one means one-step ahead prediction. Residual GM (1, 1) was improved based on GM (1, 1). Note that, different from PSO-SVR and BPNN, the GM (1, 1) and Residual GM (1, 1) was executed on the whole dataset without reconstruction and normalization-process.

Comparison of performance between the four above-mentioned methods is shown in Table V and Table VI. The obtained evaluations values of training set in Table V reflect the ability of learning the structure of data patterns. Smaller these values, the better the fitting effect on training set. While the evaluations measured on the testing set Table VI indicates the generalization potential and forecasting accuracy extending the established model to unused testing set. It indicates that the proposed PSO-SVR could achieve desirable fitting effect on the training set, but also superb generalization capacity on the testing set. Whereas the performance of all the other three model is not so satisfactory compared with PSO-SVR. Besides, the performance of BPNN is much better than that of linear methods GM (1, 1) and Residual GM (1, 1) as BPNN can also cope with nonlinear regression problem. Residual GM (1, 1) is slightly better than GM (1, 1).

In order to present a visualized performance comparison, Fig. 5~Fig. 7 depict the real observations and predicted values of PA, RA, and YA volatility, respectively. Obviously, the forecasting results of GM (1, 1) merely take on a

TABLE V  
COMPARISON OF THE FORECASTING RESULTS FOR TRAINING SET AMONG PSO-SVR, BPNN, GM (1, 1) AND RESIDUAL GM (1, 1).

Attitude	Prediction Models	MAPE (%)	RMSE	NMSE
PA	PSO-SVR	1.86	1.57e-4	0.0864
	BPNN	3.10	1.81e-4	0.1158
	GM(1,1)	10.56	8.93e-4	0.9244
RA	Residual GM(1,1)	7.43	7.90e-4	0.7228
	PSO-SVR	1.52	2.12e-4	0.1562
	BPNN	3.17	3.36e-4	0.1767
YA	GM(1,1)	10.07	5.24e-4	0.9680
	Residual GM(1,1)	5.92	3.06e-4	0.3313
	PSO-SVR	2.13	0.0049	2.185e-5
	BPNN	38.64	0.0806	0.0059
	GM(1,1)	320.55	1.12	0.9484
	Residual GM(1,1)	93.61	0.6388	0.3068

Note: evaluation criteria of GM (1, 1) and Residual GM (1, 1) was calculated on the whole dataset.

TABLE VI  
COMPARISON OF THE FORECASTING RESULTS FOR TESTING SET AMONG PSO-SVR, BPNN, GM (1, 1) AND RESIDUAL GM (1, 1).

Attitude	Prediction Models	MAPE (%)	RMSE	NMSE
PA	PSO-SVR	2.23	9.75e-5	0.4559
	BPNN	5.40	2.52e-4	0.6521
	GM(1,1)	10.56	8.93e-4	0.9244
RA	Residual GM(1,1)	7.43	7.90e-4	0.7228
	PSO-SVR	3.91	1.73e-4	0.1805
	BPNN	7.17	3.06e-4	0.2864
YA	GM(1,1)	10.07	5.24e-4	0.9680
	Residual GM(1,1)	5.92	3.06e-4	0.3313
	PSO-SVR	0.7912	0.0227	0.0012
	BPNN	4.66	0.1492	0.0523
	GM(1,1)	320.55	1.12	0.9484
	Residual GM(1,1)	93.61	0.6388	0.3068

Note: evaluation criteria of GM (1, 1) and Residual GM (1, 1) was calculated on the whole dataset.

monotonously decreasing or increasing trend, while the actual data of volatility possesses complex non-linearity, accompanying certain fluctuation. The residual GM (1, 1) can merely capture the rough changing trend of dataset. Generally speaking, the more non-linear the objective data, the smaller the forecasting accuracy of GM (1, 1) and residual GM (1, 1). The proposed PSO-SVR model exhibited excellent fitting and forecasting performance even though the change of data presents great fluctuation and complex non-linearity. As shown in Fig. 5~Fig. 7, many predicted values using PSO-SVR are overlapped with their actual observations and most of the turning points can be well captured by PSO-SVR. As BPNNs require more training data, under-fitting phenomenon is always happening. Biggish error can be distinctly observed at the wave crest and wave hollow in Fig.

7. Besides, the training process with BPNN is more time-consuming compared with SVR. Therefore, it can be concluded that the studied PSO-SVR is more effective for forecasting the volatility of satellite attitude.

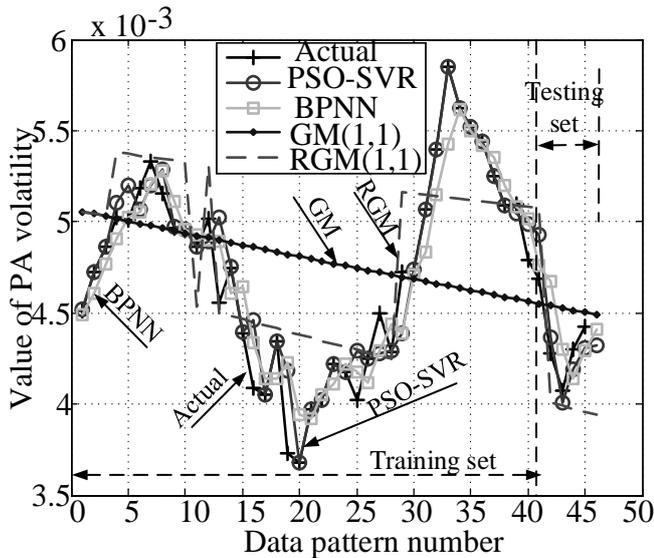


Fig. 5. Comparison between forecasting value with different prediction model and actual observations of PA volatility.

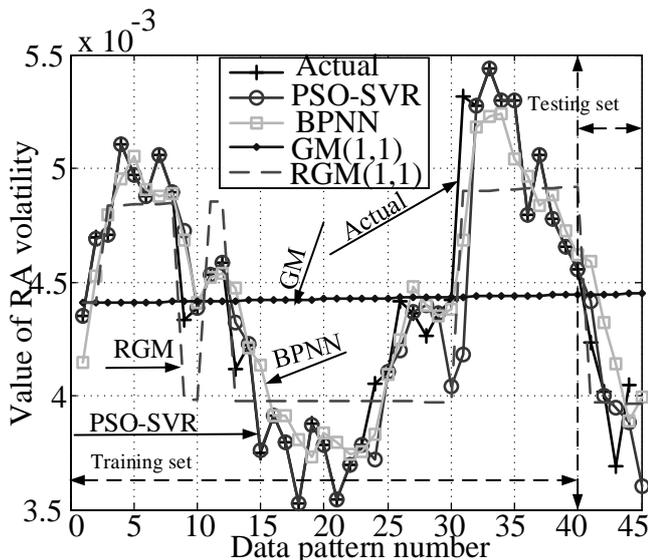


Fig. 6. Comparison between forecasting value with different prediction model and actual observations of RA volatility.

## VI. CONCLUSIONS

This paper proposes a hybrid PSO-SVR forecasting model to predict the volatility of satellite attitude. The volatility is an important indicator reflecting the running state of satellites according to experts' knowledge and statistical analysis of the telemetry data. In the PSO-SVR approach, PSO is employed to determine suitable SVR parameters since improper parameters always lead to awful performance. Experiments conducted on the real telemetry data aim to testify its feasibility in forecasting the volatility of satellite attitude. The experimental results show that PSO-SVR can obtain better performance compared with the existing prediction methods, such as neural network BPNN and grey model GM (1, 1), residual GM (1, 1). It exhibits great potential in capturing complex relationship between input and output and can avoid trapping in the local minimal that Neutral networks usually

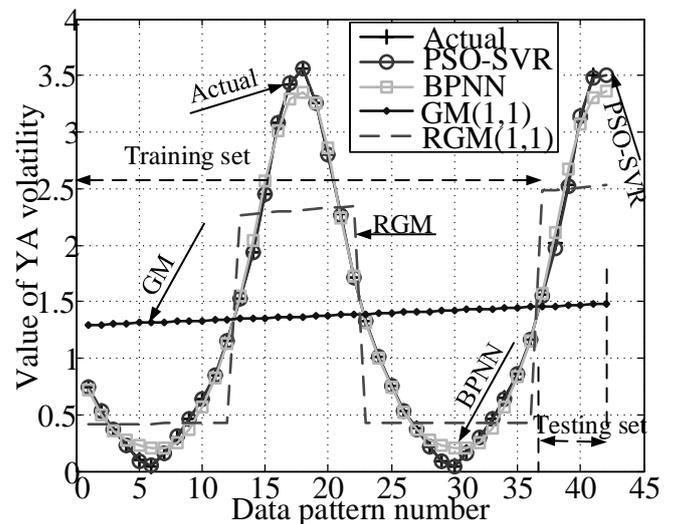


Fig. 7. Comparison between forecasting value with different prediction model and actual observations of YA volatility.

encounter. However, this forecasting method involves priori knowledge about specific satellite, such as the time interval which is relative to the usual abnormality duration, and relationship between fluctuating level and volatility ranges.

The PSO-SVR model can be used as real-time forecasting model to detect latent attitude problem in advance if the up-to-date telemetry data is used as experimental dataset. In this sense, the time taken by the whole procedure should be as least as possible so that there is enough time for regulating attitude. Furthermore, efforts will be made towards combining long-term prediction method with SVR in order to give a long-term and accurate forecast of the satellite attitude volatility. More importantly, the proposed method could be further applied in predicting other crucial parameters of satellite.

## REFERENCES

- [1] Hu Q, "Sliding mode maneuvering control and active vibration damping of three-axis stabilized flexible spacecraft with actuator dynamics," *Nonlinear Dynamics*, vol. 52, no. 3, pp. 227-248, May. 2008.
- [2] Kristiansen R, Nicklasson P J, Gravdahl J T, "Satellite attitude control by quaternion-based backstepping," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 1, pp. 227-232, Jan.2009.
- [3] Cheng C H, Shu S L, Cheng P J, "Attitude control of a satellite using fuzzy controllers," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6613-6620, Apr. 2009.
- [4] Abdelrahman M, Chang I, Park S Y, "Magnetic torque attitude control of a satellite using the state-dependent Riccati equation technique" *International Journal of Non-Linear Mechanics*, vol.46, no. 5, pp. 758-771, Jun. 2011.
- [5] Lee S H, Ahn H S, Yong K L, "Three-axis attitude determination using incomplete vector observations," *Acta Astronautica*, vol. 65, no. 7, pp. 1089-1093, Oct.-Nov. 2009.
- [6] Liu C, Zhou Z, Fu X, "Attitude determination for MAVs using a Kalman filter," *Tsinghua Science & Technology*, vol. 13, no. 5, pp. 593-597, Oct. 2008.
- [7] Wu Q, Saif M, "Neural adaptive observer based fault detection and identification for satellite attitude control systems," in *Proc. American Control Conference, IEEE*, pp. 1054-1059, Jun. 2005.

- [8] Wu Q, Saif M, "Robust fault diagnosis for a satellite large angle attitude system using an iterative neuron PID (INPID) observer," *Proc. American Control Conference, IEEE*, pp. 5710-5715, Jun. 2006.
- [9] Gao C, Zhao Q, Duan G, "Robust actuator fault diagnosis scheme for satellite attitude control systems," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2560-2580, Nov. 2013.
- [10] Thissen U, Van Brakel R, De Weijer A P, W. J Melssen, L.M.C Buydens, "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35-49, Nov. 2003.
- [11] Liu H C, Hung J C, "Forecasting S&P-100 stock index volatility: The role of volatility asymmetry and distributional assumption in GARCH models," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4928-4934, July 2010.
- [12] Lee Y S, Tong L I, "Forecasting energy consumption using a grey model improved by incorporating genetic programming," *Energy conversion and Management*, vol. 52, no. 1, pp. 147-152, Jan. 2011.
- [13] Dai C, Pi D, Fang Z, Peng H, "Wavelet Transform-based Residual Modified GM (1, 1) for Hemispherical Resonator Gyroscope Lifetime Prediction," *IAENG International Journal of Computer Science*, vol.40, no. 4, pp. 250-256, Nov. 2013.
- [14] Deh Kiani M K, Ghobadian B, Tavakoli T, A.M. Nikbakht, G. Nagahi, "Application of artificial neural networks for the prediction of performance and exhaust emissions in SI engine using ethanol-gasoline blends," *Energy*, vol. 35, no. 1, pp. 65-69, Jan. 2010.
- [15] Haykin S, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1999, ch4.
- [16] Cortes C, Vapnik V, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, Sep. 1995.
- [17] Vapnik V, *The nature of statistical learning theory*, springer, 2000.
- [18] Yang J M, Liu Z Y, Qu Z Y, "Clustering of Words Based on Relative Contribution for Text Categorization," *IAENG International Journal of Computer Science*, vol.40, no. 3, pp. 207-219, Aug. 2013.
- [19] Ingrid Nurtanio, Eha Renwi Astuti, I Ketut Eddy Purnama, Mochamad Hariadi, "Classifying Cyst and Tumor Lesion Using Support Vector Machine Based on Dental Panoramic Images Texture Features," *IAENG International Journal of Computer Science*, vol. 40, no. 1, pp. 29-37, Mar. 2013.
- [20] Drucker H, Burges C J C, Kaufman L, Smola A, Vapnik V, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp.155-161, 1997.
- [21] Smola A J, Schölkopf B, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199-222, Aug. 2004.
- [22] Yeh C Y, Huang C W, Lee S J, "A multiple-kernel support vector regression approach for stock market price forecasting," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2177-2186, Mar. 2011.
- [23] Moura M C, Zio E, Lins I D, Droguett E, "Failure and reliability prediction by support vector machines regression of time series data," *Reliability Engineering & System Safety*, vol. 96, no. 11, pp. 1527-1534, Nov. 2011.
- [24] Fei S, Wang M J, Miao Y, Tu J, Liu C, "Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil," *Energy Conversion and Management*, vol. 50, no. 6, pp. 1604-1609, Jun. 2009.
- [25] Lins I D, Araujo M, Moura M C, Silva M A, Droguett E L, "Prediction of sea surface temperature in the tropical Atlantic by support vector machines," *Computational Statistics & Data Analysis*, vol. 61, pp. 187-198, May 2013.
- [26] Kennedy J, Eberhart R, "Particle swarm optimization" in 1995 *Proc. proceedings of IEEE International conference on neural networks*, pp. 1942-1948.
- [27] Shi Y, Eberhart R. A, "modified particle swarm optimizer" in 1998 *Proc. Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73.
- [28] Chang C, Lin C J, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1-27, Apr. 2011.