

# Modeling of EDM responses by support vector machine regression with parameters selected by particle swarm optimization

Ushasta Aich\*, Simul Banerjee

Mechanical Engineering Department, Jadavpur University, Kolkata 700032, India

## ARTICLE INFO

### Article history:

Received 25 October 2012

Received in revised form 3 July 2013

Accepted 11 October 2013

Available online 23 November 2013

### Keywords:

Electrical discharge machining (EDM)

Support vector machine (SVM)

Particle swarm optimization (PSO)

## ABSTRACT

Electrical discharge machining (EDM) is inherently a stochastic process. Predicting the output of such a process with reasonable accuracy is rather difficult. Modern learning based methodologies, being capable of reading the underlying unseen effect of control factors on responses, appear to be effective in this regard. In the present work, support vector machine (SVM), one of the supervised learning methods, is applied for developing the model of EDM process. Gaussian radial basis function and  $\varepsilon$ -insensitive loss function are used as kernel function and loss function respectively. Separate models of material removal rate (MRR) and average surface roughness parameter (Ra) are developed by minimizing the mean absolute percentage error (MAPE) of training data obtained for different set of SVM parameter combinations. Particle swarm optimization (PSO) is employed for the purpose of optimizing SVM parameter combinations. Models thus developed are then tested with disjoint testing data sets. Optimum parameter settings for maximum MRR and minimum Ra are further investigated applying PSO on the developed models.

Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.

## 1. Introduction

Electrical discharge machining (EDM) is a potential process of developing complex surface geometry and integral angles in mold, die, aerospace, surgical components, etc. [1]. The process is applicable to any conductive material (resistivity should not exceed 100 ohm-cm) regardless of its hardness, toughness and strength [2]. Material is eroded by series of spatially discrete and chaotic [3] high frequency electrical discharges (sparks) of high power density between tool electrode and work piece separated by a fine gap of dielectric fluid. The working zone is completely immersed into dielectric fluid medium for enhancing electron flow in the gap, cooling after each spark and easy flushing of eroded particles. Basic scheme of EDM is shown in Fig. 1.

Electrical discharge machining process can be well characterized by two responses – material removal rate (MRR) and average surface roughness (Ra). From quantitative and qualitative point of view, higher MRR and lower Ra are always preferred. Search for accurate prediction of these responses in EDM-like complex and stochastic process is still persuaded by the process engineers. Accurate predictions of MRR and Ra are prerequisite for modern precision engineering.

Several researchers proposed various methodologies for predicting the performance of EDM process [4]. Thermo-electric model of material removal was developed by Singh et al. [5]. Panda et al. [6] introduced ANN based prediction of MRR during EDM process. Surface finish modeling by multi-parameter analysis was given by Petropoulos et al. [7]. Tsai et al. developed

\* Corresponding author. Tel.: +91 9433736906; fax: +91 3324146890.

E-mail addresses: [ushasta@yahoo.co.in](mailto:ushasta@yahoo.co.in) (U. Aich), [simul\\_b@hotmail.com](mailto:simul_b@hotmail.com) (S. Banerjee).

## Nomenclature

acc	accuracy level
$b$	bias
$c_{1,initial}, c_{1,final}$	limits of cognitive acceleration coefficient
$c_{2,initial}, c_{2,final}$	limits of social acceleration coefficient
cur	current setting (A)
$d$	training input space dimension
$f(\mathbf{x})$	target function
$g_{best}$	global best position of swarm
$iter_{max}$	maximum iteration
$n$	number of particles in swarm
$p_{best}^i$	best position of $i$ th particle
rand	random number within range (0, 1)
$t_{off}$	pulse off time ( $\mu s$ )
$t_{on}$	pulse on time ( $\mu s$ )
$v_{iter}^k$	velocity of $k$ th particle in $iter$ th iteration
$\mathbf{w}$	weight vector
$\mathbf{x}$	training input vector
$x_{iter}^k$	velocity corrected position of $k$ th particle in $iter$ th iteration
$\mathbf{y}$	training output vector
$\bar{y}$	mean of training output set
$z$	number of attributes
$C$	regularization parameter
CV	coefficient of variation
$K(x_i, \mathbf{x})$	kernel function
MAPE	mean absolute percentage error
MRR	material removal rate ( $mm^3/min$ )
$N$	number of training data
Ra	average surface roughness ( $\mu m$ )
$\varepsilon$	radius of loss insensitive hyper-tube
$\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$	Lagrange multipliers
$\xi_i, \xi_i^*$	slack variables
$\sigma$	standard deviation of radial basis function (kernel function)
$\sigma_t$	standard deviation of training output set
$\Phi(\mathbf{x})$	feature space
$\Psi_{initial}, \Psi_{final}$	limits of constriction factor
$\omega_{initial}, \omega_{final}$	limits of inertia factor

semi-empirical model of surface finish [8]. Neural network based prediction of surface finish was also proposed by them [9]. Cathode erosion model based on theoretical analysis was introduced by DiBitonto et al. [10]. Perez et al. [11] suggested theoretical modeling of energy balance in electro erosion. Saha et al. [12] developed soft computing based prediction model of cutting speed and surface finish in WEDM process.

Habib [13] applied response surface methodology (RSM) to study the parametric influence on process outputs of EDM by developing individual model for several responses like MRR, Ra, gap size (GS) and electrode wear ratio (EWR) with pulse on time, peak current, average gap voltage and percentage volume fraction of SiC present in aluminum matrix as process

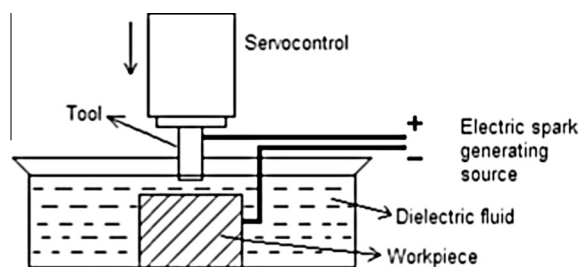


Fig. 1. Scheme of electrical discharge machining process.

variables. Optimum combination of process variables for each individual response was also determined. However, there was no clear description of optimization procedure used for finding the results.

Thus multivariable regression analysis, response surface methodology and artificial neural network (mostly back propagation neural network) are the three main data based procedures applied for modeling EDM process. Compared to artificial neural network, support vector machine (a powerful learning system), is devoid of the four problems of efficiency of training, efficiency of testing, over-fitting and algorithm parameter tuning [14]. Besides, the insensitive zone of SVM absorbs the small scale random fluctuations appeared in stochastic type responses which is beneficial for other researchers to apply these models on different products obtained in different batches.

There are a few application of SVM learning system on manufacturing processes is found. Surface roughness in CNC turning of AISI 304 austenitic stainless steel was modeled with high correlation coefficient through three SVM learning systems (LS-SVM, Spider SVM and SVM-KM) and ANN [15]. Internal parameters of SVM ( $C$  and  $\sigma$ ) were set by grid search method. Though it is reported that for model development SVM learning systems consume less time than ANN but no such clear explanation about those specific choices of searching region of SVM parameters was stated. Also, the SVM parameters' value obtained through grid search method depends on the choice of jumping interval. Ramesh et al. [16] conducted CNC end milling operation on 6061 aluminum varying feed rate, spindle speed and depth of cut. They employed SVM for modeling of surface finish in milling operation. Though their estimated model can predict with 8.34% error which is better compared to 9.71% in prediction through regression model, but the procedure of their iterative choice of internal parameters of SVM (error, global basis function width and upper bound) was not reported anywhere. Models of surface finish in face milling of steel St 52-3 were also developed using multivariable regression analysis, SVM learning system and Bayesian neural network by Lela et al. [17]. It was reported that SVM learned model estimated better than regression model. All three internal parameters of SVM were chosen by leave-one-out cross validation procedure keeping two parameters fixed at particular values and other one is searched minimizing the mean square error. A continuous optimization technique which simultaneously searches the three parameters should be used to get better result for a newly developed system. Zhang et al. [18] developed separate hybrid models of processing time and electrode wear in micro-EDM through SVM. They also employed discrete level leave-one-out cross-validation for choosing  $C$  and  $\varepsilon$ . Though they used Gaussian kernel function but no such choice of  $\sigma$  is reported. However, such predictive models of MRR and average surface roughness in EDM like complex and stochastic process in particular are not found yet in the literature. Therefore, modeling of responses through SVM and optimization of those representative models by PSO are proposed in the present work.

In this study, therefore, two conflicting type responses – MRR and  $R_a$  are chosen for modeling EDM process by support vector regression with current, pulse on time and pulse off time as control parameters. Models for MRR and  $R_a$  are fitted based on structural risk minimization principle [19]. For accurate model fitting, three internal parameters of SVM, namely regularization parameter ( $C$ ), radius of loss insensitive hyper tube ( $\varepsilon$ ) and standard deviation ( $\sigma$ ) of kernel function are to be correctly set. Particle swarm optimization is employed for the selection of optimum combination of these three internal parameters. Models thus developed, are then tested for accuracy through follow up experiments. Further, optimum process parameter (current setting, pulse on time and pulse off time) settings for maximum MRR and minimum  $R_a$  are estimated separately by applying PSO on the respective representative SVM learned models. Literature survey made so far reveals that no such work is reported till now.

## 2. Support vector machine (SVM)

Support vector machine, a supervised batch learning system, is firmly grounded in the framework of statistical learning theory. Vapnik [19] introduced structural risk minimization (SRM) principle instead of empirical risk minimization (ERM), implemented by most of the traditional artificial intelligence based modeling technologies. Neural network approaches may have suffered with generalization, producing over fitted models but SRM minimizes upper bound on the expected risk, as opposed to ERM, that minimizes error on the training data. This difference equips SVM with a greater ability to generalize [20].

Ultimate goal in modeling of empirical data is to choose a model from hypothesis space, which is closest to the underlying target function. Suppose, a set of training data  $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$  is used for model developing in  $d$  dimensional input space (i.e.  $x \in R^d$ ). Key assumption in model developing is that training and testing data set are disjoint, independent and identically distributed according to the unseen but fixed underlying function [14]. The linear target function may be represented in the form [21]

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (1)$$

where  $\langle \cdot \rangle$  indicates dot product in vector space. If the input pattern does not hold any linear relation to output, (non linear SVM regression model is shown in Fig. 2) then they are mapped to feature space  $\Phi(\mathbf{x})$  from high dimensional input space via kernel functions. So, optimal choice of weight factor  $\mathbf{w}$  and threshold  $b$  (bias term) is prerequisite of accurate modeling. Flatness of the model is controlled by minimizing Euclidean norm  $\|\mathbf{w}\|$ . Besides, empirical risk of training error should also be minimized [22]. So, regularized risk minimization problem for model developing can be written as follows.

$$R_{\text{reg}}(f) = \|\mathbf{w}\|^2 / 2 + C \sum_{i=1}^N L(y_i, f(x_i)) \quad (2)$$

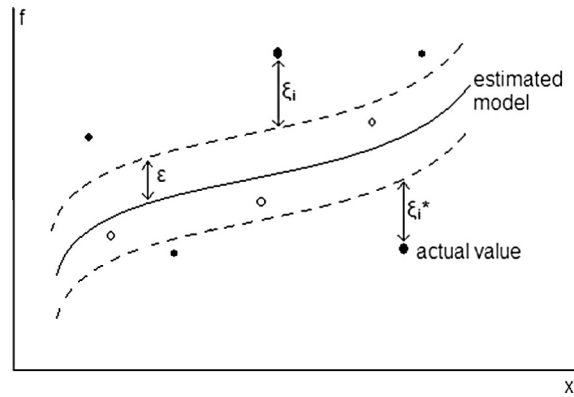
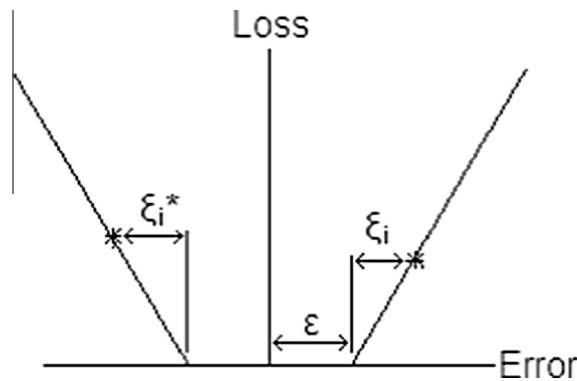


Fig. 2. Non-linear SVM regression model.

Fig. 3.  $\epsilon$ -Insensitive loss function.

Weight vector  $\mathbf{w}$  and the bias term  $b$  can be estimated by optimizing this function (Eq. (2)), which minimizes not only empirical risks but also reduces generalization error i.e. over fitting of model simultaneously. Here,  $L(\mathbf{y})$ , a loss function is introduced to penalize over fitting of model with training points. A number of loss functions are already developed for handling different types of problem [20].  $\epsilon$ -insensitive loss function (Fig. 3) is mostly used for process modeling problems. This function may be defined as

$$L(y_i, f(x_i)) = |y_{i,\text{experimental}} - f(x_i)| - \epsilon, \quad \text{if } |y_{i,\text{experimental}} - f(x_i)| \geq \epsilon \\ = 0, \quad \text{if } |y_{i,\text{experimental}} - f(x_i)| < \epsilon \quad (3)$$

Here points inside the  $\epsilon$ -tube are considered as zero loss, otherwise a penalization is calculated by introducing  $C$ , which is a trade-off between flatness and complexity of the model. Practical significance of this insensitive zone is that the points inside the hyper-tube i.e. close enough to estimated model are deemed to be well estimated and those outside the tube contribute training error loss. These outsiders of the insensitive zone belong to support vector group. So, the size of  $\epsilon$ -insensitive zone controls number of support vectors. As radius of insensitive hyper-tube increases, number of support vector reduces and flexibility of the model diminishes. This behavior may be advantageous for eliminating the effect of small random noise in output, but larger value of  $\epsilon$  will not completely extract the unseen target function. Besides, higher value of  $C$  makes the model more complex with the chance of over fitting, but too small value may increase training errors. So, optimum choice of this regularization parameter is necessary for better modeling. Two positive slack variables  $\xi_i$  and  $\xi_i^*$  are introduced [19,21] to cope with infeasible constraints of the optimization problem. Hence the constrained problem can be reformulated as

$$\begin{aligned} \text{minimize : } & \|\mathbf{w}\|^2/2 + C \sum_{i=1(1)N} (\xi_i + \xi_i^*) \\ \text{subject to : } & y_{i,\text{exp}} - \langle \mathbf{w}, \mathbf{x} \rangle_i - b \leq \epsilon + \xi_i \\ & \langle \mathbf{w}, \mathbf{x} \rangle_i + b - y_{i,\text{exp}} \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \quad i = 1(1)N \end{aligned} \quad (4)$$

This problem can be efficiently solved by standard dualization principle utilizing Lagrange multiplier. A dual set of variables are introduced for developing Lagrange function. It is found that this function has a saddle point with respect to both primal and dual variables at the solution. Lagrange function can be stated as

$$L = \|\mathbf{w}\|^2/2 + C\sum_i 1(1)N(\xi_i + \xi_i^*) - \sum_{i=1(1)N}(\eta_i \xi_i + \eta_i^* \xi_i^*) - \sum_{i=1(1)N} \alpha_i (\varepsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x} \rangle_i + b) - \sum_{i=1(1)N} \alpha_i^* (\varepsilon + \xi_i^* - y_i - \langle \mathbf{w}, \mathbf{x} \rangle_i - b) \quad (5)$$

where  $L$  is the Lagrangian and  $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$  are Lagrange multipliers satisfying

$$\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0$$

So, partial derivatives of  $L$  with respect to  $\mathbf{w}, b, \xi_i, \xi_i^*$  will give the estimates of  $\mathbf{w}$  and  $b$ . The present problem is solved by using LibSVM MATLAB Toolbox.

Support vectors can be easily identified from the value of difference between Lagrange multipliers ( $\alpha_i, \alpha_i^*$ ). Very small values (close to zero) indicate the points inside the insensitive hyper-tube but non-zero values belong to support vector group [23]. The  $\mathbf{w}$  can be calculated by [21]

$$\mathbf{w} = \sum_{i=1(1)N} (\alpha_i - \alpha_i^*) \Phi(\mathbf{x}_i) \quad (6)$$

The idea of kernel function  $K(\mathbf{x}_i, \mathbf{x})$  gives a way of addressing the curse of dimensionality [20]. It helps to enable the operations to be performed in the feature space rather than potentially high dimensional input space. A number of kernel functions satisfying Mercer's condition were suggested by researchers [23,24]. Each of these functions has its own specialized applicability. Gaussian radial basis function with  $\sigma$  standard deviation (given in Eq. (7)) is commonly used for its better potentiality to handle higher dimensional input space.

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\|\mathbf{x}_i - \mathbf{x}\|^2/2\sigma^2) \quad (7)$$

Thus the final model with optimum choice of  $C, \varepsilon$  and  $\sigma$  may be presented as [21]

$$f(\mathbf{x}) = \sum_{i=1(1)N} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b \quad \left| \begin{array}{l} C_{\text{optimum}} \\ \varepsilon_{\text{optimum}} \\ \sigma_{\text{optimum}} \end{array} \right. \quad (8)$$

### 3. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) technique is one of the most advanced evolutionary computational intelligence based optimization methodologies for optimizing real world multimodal problems. PSO mimics natural behavior found in flock of birds or school of fish seeking their best food sources [25]. In this population based swarm intelligence technique a set of randomly initialized particles (swarm) are always updated in position and velocity by gathering information from themselves. Effect of each particle as well as the whole swarm's experience modifies position of the population forwarding to optimum zone. Rate of convergence is purposefully controlled by different factors. Position of global optimum is not affected by the choice of these factors, but convergence is delayed due to improper choice or may lead to entrapping in local optima. For multi variable problem in high dimensional space, time and memory space needed for reaching optimum solution by PSO is very important.

Number of particles ( $n$ ) in swarm should be within the range (10,40) [26]. Lower choice may not gather information from whole space but higher value of  $n$  will take longer time to converge in optimum zone.

Inertia factor ( $\omega$ ) controls the effect of previous velocity of individual particle on current velocity. To modify the rate of convergence another control on simulation was done by introducing constriction factor ( $\Psi$ ) [27]. This term bounds the velocity effect of particle on their position avoiding clamping of particles to one end of search space [28]. So, higher values of inertia and constriction factor ensure wide searching which is necessary at initial stage but gradual convergence is enhanced at moderately lower value.

Another two important factors are cognitive acceleration coefficient ( $c_1$ ) and social acceleration coefficient ( $c_2$ ) which greatly control the influence of individual's and whole swarm's experience respectively on particle's new velocity. Influence of particle's individual best ( $p_{\text{best}}^i$  of  $i$ th particle) experience favors good exploration in the search space but swarm's best position ( $g_{\text{best}}$ ) always guide to converge near optimum zone. So, choice of these factors becomes important for converging to global optimum zone quickly avoiding premature entrapping in local optima.

Several researchers use different values of these control factors for their different type of problem definitions. However, in general for most of the cases nearly a same range is suggested irrespective of the nature of problem [29]. Shi and Eberhart [30] suggested linearly decreasing inertia factor from 0.9 to 0.4. Cognitive acceleration coefficient should vary linearly with iterations from 2.5 to 0.5 while the variation of social acceleration coefficient would occur just in reverse order [31]. Since constriction factor directly control the optimization time, it may be considered as linearly time varying from 0.9 to 0.4.

Further, maximum number of iterations is to be set properly. A large value is necessary for adequate convergence. In other words simulation will be terminated before reaching this limiting value.

#### 4. Experiment

Experiment was conducted on an EDM machine (Tool Craft A25 EDM Machine) operating with commercially available kerosene oil as dielectric medium and an open circuit voltage of 70 volts. High speed steel of specimen C-0.80%, W-6%, Mo-5%, Cr-4%, V-2% equivalent to grade M2 was chosen as the work-piece material. The density of material is 8144 kg/m<sup>3</sup>. The tool material selected is electrolytic copper with density 8904 kg/m<sup>3</sup> and has circular cross-section of 14 mm<sup>2</sup>. Tool is connected to the positive terminal. Three most dominating control parameters of EDM process namely current setting (cur), pulse-on time ( $t_{on}$ ) and pulse-off time ( $t_{off}$ ) are considered as input parameters while material removal rate (MRR) and average surface roughness (Ra) of machined surface are taken as response parameters. Based on the availability of the machine setting, levels of the input parameters are selected and presented in Table 1.

Experiments are performed based on the procedure of full factorial design. Loss in weight of work piece material and average value of surface roughness of machined surface are measured in each experimental run. For the purpose of determining the material removal rate (MRR), the loss in weight is measured by standard measuring balance of least count 0.1 mg. It is then divided by the density of work-piece material in order to convert it into volumetric term and is further divided by the actual machining time to obtain the MRR in terms of mm<sup>3</sup>/min. The typical machining time with lowest material removal rate was observed as 45 min for a depth of 1.1 mm. The average surface roughness (Ra) of the machined work piece is evaluated by the Taylor Hobson Precision Surtronic 3+ Roughness Checker. Here a sample length of 4 mm is taken and stylus tip radius of 5  $\mu$ m is used. The value of surface roughness parameter Ra in micron for each experimental run was obtained directly from the Taly-profile software integrated with the machine. The arithmetic mean of the values of measurements taken along three mutually 120° apart directions over the area subjected to the EDM process, is taken as the representative value of Ra. Out of 64 unique combinations of the levels of different input parameters 54 sets are chosen randomly for training the learning process and the rest are left for the purpose of testing.

#### 5. Analysis and discussion

Based on training data sets, models for MRR and Ra are fitted through learning process of SVM. Fitted models are tested through another 10 sets of data. The models thus developed are then used for searching optimal process parameter (current setting, pulse on time and pulse off time) combination for maximum MRR and minimum Ra separately.

##### 5.1. Model development

Significance of parameter selection in SVM for model development and thereby for good prediction is evident. Particle swarm optimization (PSO) could be used effectively for this purpose. The most influencing parameters in SVM learning process are C,  $\varepsilon$  and  $\sigma$ . With optimum choice of C,  $\varepsilon$  and  $\sigma$ , a set of Lagrange multipliers ( $\alpha_i, \alpha_i^*$ ) could be found and the model can be estimated using Eq. (8).

At first, search spaces for each of the parameters (C,  $\varepsilon$  and  $\sigma$ ) are to be determined. A wide range of each parameter may be a good search space but this will take large computation time. Estimation of C based on experimental outputs (target values) was suggested by Cherkassky and Ma [32] and Levis and Papageorgiou [24].

$$C = \max(|y_{\text{bar}} + 3\sigma_t|, |y_{\text{bar}} - 3\sigma_t|) \quad (9)$$

where,  $y_{\text{bar}}$  and  $\sigma_t$  represent mean and standard deviation of target values. However, this exact value may lead to erroneous modeling due to presence of random noises. So, it is better to choose a range for C and searching operation is to be performed within these limits. At first, using Eq. (9) C value is calculated for both MRR and Ra as  $C_{\text{MRR}} = 18.9490$  and  $C_{\text{Ra}} = 10.6760$ . Then, considering normal distribution with mean at this calculated C value and 50% of mean value as standard deviation, two points are chosen at both sides of estimated C value. Thus range of C for each of MRR and Ra is determined (Table 2).

Based on experiments, suitable range of  $\varepsilon$  and that of  $\sigma$  are also suggested in [24,32]. Inside the bounds of  $\sigma$ , generalization performance is found as stable. Those limits can be calculated from

$$\varepsilon = [y_{\text{bar}}/30, y_{\text{bar}}/10] \quad \sigma = [0.1^{(1/z)}, 0.5^{(1/z)}] \quad (10)$$

**Table 1**  
Parameters and their levels.

	Level 1	Level 2	Level 3	Level 4
Current setting (A)	3	6	9	12
Pulse on time ( $\mu$ s)	50	100	150	200
Pulse off time ( $\mu$ s)	50	100	150	200

**Table 2**  
Searching range of SVM parameters.

	MRR	Ra
C	[6.1679, 32.2986]	[3.4702, 18.1972]
$\varepsilon$	[0.1782, 0.5345]	[0.1667, 0.5000]
$\sigma$	[0.4642, 0.7937]	[0.4642, 0.7937]

Here,  $z$  indicates number of most influencing attributes in the process. In EDM it is 3 namely current, pulse on time and pulse off time. Using Eq. (10),  $\varepsilon$  and  $\sigma$  values for each of MRR and Ra are calculated (Table 2).

For better implementation of this estimated search range it was suggested [24,32] to normalize the training inputs within the range (0,1). So, the chosen control parameters of EDM process i.e. current, pulse on time and pulse off time are normalized using the following formulae.

$$\begin{aligned} x_{1,\text{norm}} &= (\text{cur} - 3)/(12 - 3) \\ x_{2,\text{norm}} &= (t_{\text{on}} - 50)/(200 - 50) \\ x_{3,\text{norm}} &= (t_{\text{off}} - 50)/(200 - 50) \end{aligned} \quad (11)$$

As SVM reduces the chance of generalization error, in the present study, parameters of SVM are selected through minimization of mean absolute percentage error (MAPE) in estimation of training outputs. For the purpose of optimization, MAPE being a function of C,  $\varepsilon$  and  $\sigma$ , is considered as the objective function. At every step during marching procedure, a different combination of C,  $\varepsilon$  and  $\sigma$  will develop different regression models. For each of the models developed through the change of C,  $\varepsilon$  and  $\sigma$  values, MAPE (Eq. (12)) is calculated and searching of minimum MAPE is done among the models using the procedure of particle swarm optimization.

$$\text{MAPE}(\%) = (100/N) \sum_{i=1}^N (|y_{i,\text{experimental}} - y_{i,\text{estimated}}| / y_{i,\text{experimental}}) \quad (12)$$

In most of the articles, termination criterion is set by a predefined maximum number of iterations. This may not ensure global optima. Rather, a statistical measure of relative dispersion, coefficient of variation (CV) of particles' position in search space, is expected to help avoiding the premature stopping of simulation procedure. Coefficient of variation is calculated as a ratio of standard deviation to mean and expressed in percentage. A very small value of this criterion indicates a set of densely compact particles in the swarm, irrespective of their search range in different dimensions. Small coefficient of variation in conjunction with negligible change in the values of MAPE will ensure global optimum. Besides, nature of change of different components of CV values in different dimensions may give a preliminary idea about the underlying relationship among output and inputs. A simple PSO algorithm for searching optimum combination of C,  $\varepsilon$  and  $\sigma$  for minimum MAPE is stated as follows.

- Step 1: Choose number of particles in swarm ( $n$ ), maximum number of iterations  $iter_{\text{max}}$ , accuracy level (acc). Set, inertia factor range ( $\omega_{\text{initial}}$ ,  $\omega_{\text{final}}$ ), cognitive acceleration coefficient range ( $c_{1,\text{initial}}$ ,  $c_{1,\text{final}}$ ), social acceleration coefficient range ( $c_{2,\text{initial}}$ ,  $c_{2,\text{final}}$ ) and constriction factor range ( $\Psi_{\text{initial}}$ ,  $\Psi_{\text{final}}$ ).
- Step 2: Set  $iter = 1$ . Randomly initialize the position of  $n$  particles in swarm i.e.  $n$  set of initial combination of C,  $\varepsilon$  and  $\sigma$ . Also randomly set initial velocity of each particle using

$$V_{\text{iter}}^{(i)} = \text{range}^{(i)} X_{\text{rand}} \quad i = 1(1)3$$

where, “range<sup>(i)</sup>” and “rand” indicate range of C,  $\varepsilon$  and  $\sigma$  (shown in Table 2) and a random number within the range (0,1) respectively.  $i = 1(1)3$  stands for a counter which will repeat the calculation three times at that step to generate initial velocity vectors for three control parameters of this algorithm i.e. internal parameters of SVM-C,  $\varepsilon$  and  $\sigma$ . So,  $n$  set of velocity vectors with components along C,  $\varepsilon$  and  $\sigma$  coordinates are initialized.

- Step 3: Set  $t = 1$  and  $p_{\text{best}}^t$  and  $g_{\text{best}}$  as current position vector.
- Step 4: Check whether  $t = n$  or not. If yes go to step 8, otherwise set  $t = t + 1$  and go to step 5.
- Step 5: Calculate MAPE (objective function value) at  $t$ th particle position in swarm. If  $iter = 1$ , then set  $p_{\text{best}}^t$  as current position vector and go to step 7. Otherwise, go to step 6.
- Step 6: If this MAPE value is smaller (better) than MAPE at already set  $p_{\text{best}}^t$ , then update  $p_{\text{best}}^t$  vector with current position vector. Otherwise  $p_{\text{best}}^t$  is kept unaltered.
- Step 7: Check whether this current MAPE value is lesser (better) than MAPE at already set  $g_{\text{best}}$ . If yes, update  $g_{\text{best}}$  vector with current position vector and go to step 4. Otherwise  $g_{\text{best}}$  vector is kept unaltered and go to step 4.
- Step 8: Calculate coefficient of variation of all particles' position in current swarm. Each position vector has three components along C,  $\varepsilon$  and  $\sigma$ . So, three values of coefficient of variation will be found.
- Step 9: If all these three coefficient of variation values are simultaneously less than accuracy level, then terminate the simulation process and go to step 14, otherwise set  $k = 1$  and go to step 10.



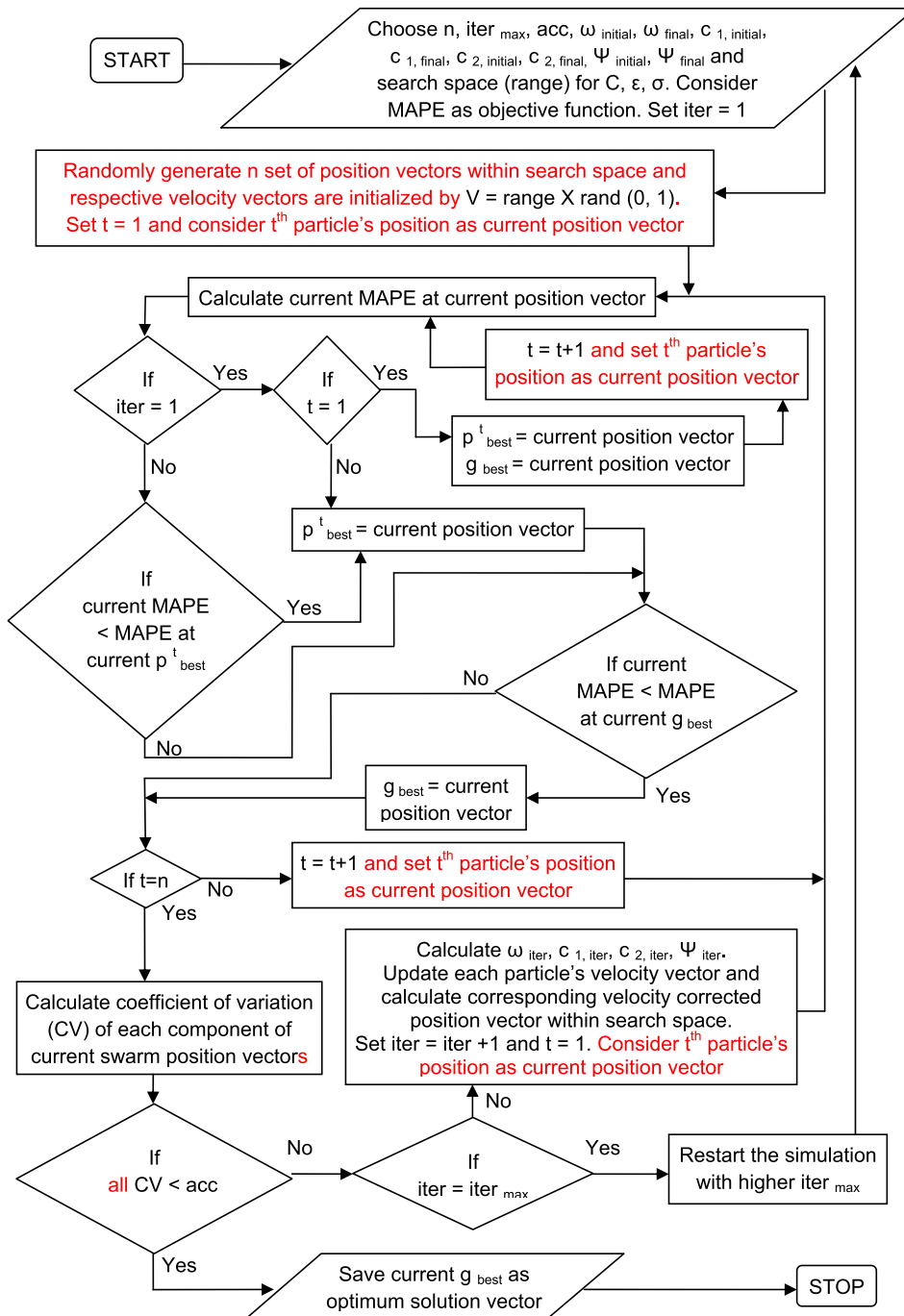


Fig. 4. Flow chart for searching optimum combination of  $C$ ,  $\varepsilon$  and  $\sigma$  through PSO by minimizing MAPE of training data.

Step 10: Considering linear gradient of  $\omega$ ,  $c_1$ ,  $c_2$  and  $\Psi$ , set the value of these internal control factors for this iteration ( $iter$ ) using these relations:

$$\omega_{iter} = \omega_{initial} + (\omega_{final} - \omega_{initial})(iter - 1)/(iter_{max} - 1)$$

$$c_{1,iter} = c_{1,initial} + (c_{1,final} - c_{1,initial})(iter - 1)/(iter_{max} - 1)$$



$$c_{2,iter} = c_{2,initial} + (c_{2,final} - c_{2,initial})(iter - 1)/(iter_{max} - 1)$$

$$\psi_{iter} = \psi_{initial} + (\psi_{final} - \psi_{initial})(iter - 1)/(iter_{max} - 1)$$

Step 11: Update  $k$ th particle's velocity vector and its velocity corrected position vector using the following formulae.

$$V_{(m)iter+1}^k = \omega_{iter}(v_{(m)iter}^k) + c_{1,iter}(\text{rand})(p_{(m)best}^k - x_{(m)iter}^k) + c_{2,iter}(\text{rand})(g_{(m)best} - x_{(m)iter}^k)$$

$$x_{(m)iter+1}^k = x_{(m)iter}^k + \psi(v_{(m)iter+1}^k) \quad m = 1(1)3$$

**Table A1**

Initial position and velocities of particles for modeling of MRR.

Particle no.	Initial position vector (C, $\varepsilon$ , $\sigma$ )	Initial velocity vector (C, $\varepsilon$ , $\sigma$ )
1	(26.5520, 0.3975, 0.6308)	(10.4560, 0.2607, 0.1159)
2	(11.3061, 0.4345, 0.6482)	(7.6523, 0.2807, 0.1979)
3	(30.2338, 0.2719, 0.7053)	(17.7463, 0.0440, 0.1434)
4	(20.4459, 0.4641, 0.5876)	(11.8924, 0.3109, 0.2240)
5	(18.7940, 0.2769, 0.6385)	(7.0299, 0.3484, 0.1760)
6	(9.2729, 0.2674, 0.4771)	(12.5413, 0.1209, 0.2960)
7	(24.7718, 0.5111, 0.6108)	(18.1684, 0.2927, 0.1716)
8	(17.4105, 0.4780, 0.5343)	(19.2243, 0.3129, 0.1180)
9	(26.2052, 0.1990, 0.7913)	(22.3141, 0.2996, 0.3251)
10	(14.2206, 0.3710, 0.5771)	(11.6949, 0.2823, 0.2889)
11	(27.7687, 0.4845, 0.5796)	(14.2339, 0.3124, 0.0870)
12	(22.0594, 0.3478, 0.6225)	(12.3910, 0.2842, 0.2600)
13	(30.0507, 0.4632, 0.5451)	(5.6264, 0.2415, 0.0753)
14	(25.0192, 0.2802, 0.4671)	(4.8870, 0.3083, 0.2764)
15	(21.3330, 0.4329, 0.4992)	(15.7863, 0.2070, 0.0083)
16	(26.1167, 0.4849, 0.6007)	(12.7524, 0.3248, 0.2038)
17	(15.5942, 0.2519, 0.6216)	(2.0804, 0.2079, 0.1145)
18	(9.1076, 0.4269, 0.7843)	(22.1432, 0.2344, 0.1683)
19	(9.8836, 0.3565, 0.6236)	(8.3997, 0.2215, 0.0623)
20	(14.3355, 0.2717, 0.7415)	(13.0249, 0.1841, 0.0653)

**Table A2**

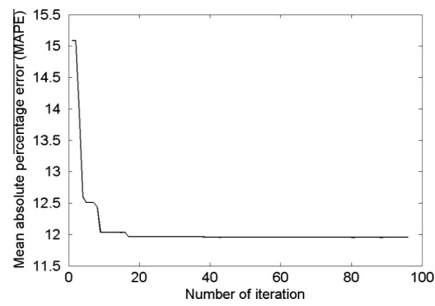
Initial position and velocities of particles for modeling of Ra.

Particle no.	Initial position vector (C, $\varepsilon$ , $\sigma$ )	Initial velocity vector (C, $\varepsilon$ , $\sigma$ )
1	(12.8771, 0.2107, 0.6416)	(12.7077, 0.1085, 0.2065)
2	(10.1715, 0.4953, 0.6825)	(14.5676, 0.3285, 0.2813)
3	(3.7682, 0.3277, 0.6990)	(5.1092, 0.0221, 0.1257)
4	(16.1091, 0.3597, 0.6867)	(10.1375, 0.3204, 0.2446)
5	(12.0130, 0.1796, 0.5856)	(0.1570, 0.0025, 0.2478)
6	(7.3998, 0.1823, 0.6312)	(8.6184, 0.1745, 0.1942)
7	(11.4877, 0.4076, 0.7137)	(4.0899, 0.1468, 0.3208)
8	(7.8380, 0.2506, 0.4792)	(11.6317, 0.0735, 0.0727)
9	(13.5376, 0.3246, 0.6025)	(5.2951, 0.0289, 0.0534)
10	(7.6890, 0.3996, 0.7165)	(6.8345, 0.2397, 0.1693)
11	(4.9043, 0.4160, 0.5482)	(11.3837, 0.2437, 0.1824)
12	(13.3836, 0.4329, 0.7757)	(4.9409, 0.0172, 0.0347)
13	(4.8434, 0.2424, 0.6933)	(12.7079, 0.0517, 0.3212)
14	(10.2481, 0.2625, 0.6693)	(3.1943, 0.0289, 0.0352)
15	(8.9305, 0.4027, 0.5961)	(7.7303, 0.0137, 0.2874)
16	(14.3399, 0.3753, 0.4842)	(0.4980, 0.0273, 0.3139)
17	(14.7289, 0.2964, 0.5273)	(4.6034, 0.1780, 0.0035)
18	(5.2923, 0.4699, 0.5416)	(8.0744, 0.2215, 0.0751)
19	(14.4048, 0.2803, 0.7503)	(6.2102, 0.1450, 0.2158)
20	(11.8346, 0.2630, 0.6681)	(4.8394, 0.2524, 0.1764)

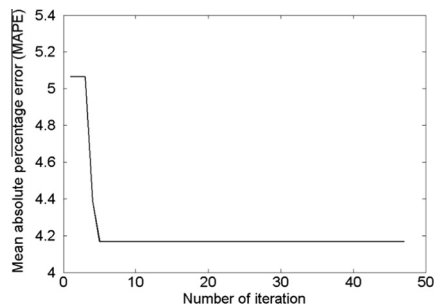
**Table 3**

Optimum internal parameters of SVM and training error measurement.

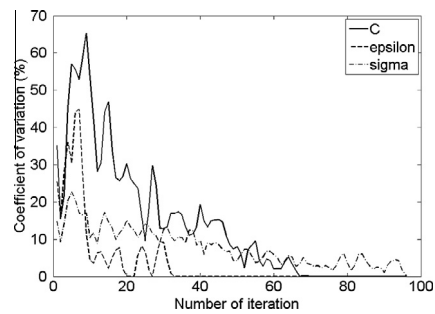
	Simulation time (s)	C	$\varepsilon$	$\sigma$	MAPE	$r^2$
MRR	6200	32.2986	0.1782	0.6335	11.9544	0.9789
Ra	3048	17.0212	0.1667	0.4642	4.1671	0.9793



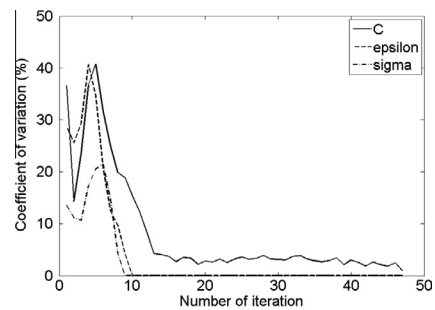
**Fig. 5.** Marching steps for optimization of MAPE for modeling MRR.



**Fig. 6.** Marching steps for optimization of MAPE for modeling Ra.



**Fig. 7.** Change of CV along different dimensions with iteration for optimization of MAPE for MRR.



**Fig. 8.** Change of CV along different dimensions with iteration for optimization of MAPE for Ra.

$m = 1(1)3$  stands for a counter which will repeat the calculation three times at that step to evaluate velocity and velocity corrected position vectors for three control parameters of this algorithm i.e. internal parameters of SVM -  $C$ ,  $\varepsilon$  and  $\sigma$ .

Step 12: If new positions of particles ( $C$ ,  $\varepsilon$  and  $\sigma$ ) go outside the specified search range (shown in Table 2), then reset its current position to the violating limit. If  $k = n$ , then go to step 13. Otherwise set  $k = k+1$  and go to step 11.

Step 13: If  $iter = iter_{max}$ , then go to step 1 and restart the simulation with larger  $iter_{max}$ , such that termination criterion will be satisfied before reaching that new limit. Otherwise set  $iter = iter + 1$  and  $t = 1$ , go to step 5.

Step 14: Set current  $g_{best}$  of the swarm as optimum point, i.e. components of latest  $g_{best}$  vector are become the optimum combination of  $C$ ,  $\varepsilon$  and  $\sigma$  for minimum MAPE in fitting the SVM learned regression model on training data.

**Table A3**

Difference of Lagrange multipliers ( $\alpha_i$ ,  $\alpha_i^*$ ) for MRR and Ra model (# and ## indicate support vectors for MRR and Ra model respectively).

Sl no	Training input vector ( $cur$ , $t_{on}$ , $t_{off}$ )	Difference of Lagrange multipliers for MRR model ( $b = 0$ ) ( $C = 32.2986$ , $\varepsilon = 0.1782$ , $\sigma = 0.6335$ )	Difference of Lagrange multipliers for Ra model ( $b = 0$ ) ( $C = 17.0212$ , $\varepsilon = 0.1667$ , $\sigma = 0.4642$ )
1	(3,50,50)	-15.345666149315205#	-2.264845323911442##
2	(3,50,100)	26.507158461315061#	0.000000011813290
3	(3,50,150)	0.000000040193789	1.026952556018298##
4	(3,50,200)	-0.000000142314069	0.000000006473360
5	(3,100,50)	-19.872689054470300#	-0.000000013596888
6	(3,100,100)	-13.402323784852832#	0.000000055512598
7	(3,100,200)	-11.879095178170926#	-2.486718086738095##
8	(3,150,50)	0.000000216396248	2.674254916782730##
9	(3,150,100)	5.594184239597962#	0.000000172734159
10	(3,150,150)	0.000000114025404	-0.000000055916479
11	(3,150,200)	0.000000028767992	2.190396504210260##
12	(3,200,50)	-0.000000094320172	2.158909751985969##
13	(3,200,100)	-4.706469840570585#	-8.441227167740673##
14	(3,200,150)	0.000000010573454	1.426969738212617##
15	(3,200,200)	-0.000000027704970	-7.60888739335333##
16	(6,50,50)	23.132040361262796#	3.750631579517261##
17	(6,50,100)	-32.298599643643684#	2.933436104996499##
18	(6,50,200)	7.008854031990202#	1.119227109544963##
19	(6,100,50)	32.294582095275317#	5.600416392176629##
20	(6,100,100)	32.298599935395558#	-5.126535621250921##
21	(6,100,200)	0.313818907868921	3.531298268909546##
22	(6,150,50)	-0.000000044518715	-8.604990297390348##
23	(6,150,150)	-32.298599963825701#	-0.000000038854848
24	(6,150,200)	32.298599017004229#	0.000000020207895
25	(6,200,50)	-9.975078655681205#	-0.000000016726673
26	(6,200,100)	32.298599914000832#	17.021199964019374##
27	(6,200,150)	0.000000036654417	-3.030093015770768##
28	(6,200,200)	0.000000188188123	17.021199976607981##
29	(9,50,50)	-32.298599911582038#	-7.012770145575573##
30	(9,50,100)	-25.456218553630912#	0.000000012047670
31	(9,50,150)	32.298599859007638#	0.000000178980284
32	(9,50,200)	-32.298599600195914#	-0.230783123216074##
33	(9,100,50)	32.298599960835936#	10.825847607118046##
34	(9,100,100)	-15.098104432736248#	-9.142806804491199##
35	(9,100,200)	25.858796128450866#	3.406058473239213##
36	(9,150,50)	-32.298599914127628#	-0.000000243752026
37	(9,150,100)	-32.298599937977286#	-0.000000048275993
38	(9,150,150)	32.298599957720391#	17.021199984790353##
39	(9,150,200)	-32.298599906057696#	-17.021199968287913##
40	(9,200,100)	0.000000020248129	-8.096656866905709##
41	(9,200,150)	-20.586503842472563#	-13.817004398149844##
42	(9,200,200)	-11.264009054310538#	-5.064745196096352##
43	(12,50,50)	32.298599968237660#	0.000000003897541
44	(12,50,100)	-32.298599959930343#	11.455665956866689##
45	(12,50,150)	32.298599948842764#	-11.501354573286539##
46	(12,50,200)	-7.806425327883569#	10.020409122174373##
47	(12,100,50)	-13.956539491230133#	-5.53568533255612##
48	(12,100,100)	23.594143869393140#	9.994838079535516##
49	(12,100,200)	-0.000000285099019	-5.507976169177828##
50	(12,150,50)	32.298599987456114#	6.478832355090940##
51	(12,150,100)	-32.298599975039302#	-17.021199993098836##
52	(12,150,200)	-13.357954398775604#	9.469377165782142##
53	(12,200,100)	29.208451563381381#	17.020860227936449##
54	(12,200,200)	32.298599654382265#	9.509967268663713##

Therefore, final  $g_{best}$  vector satisfying the termination criterion becomes global optimum in that specified search space. The flow chart for searching optimum internal parameter combination ( $C$ ,  $\varepsilon$  and  $\sigma$ ) of SVM through PSO by minimizing MAPE of training data is also shown in Fig. 4.

In the present work, different control factors of PSO are set as, particle swarm  $n = 20$ ,  $\omega_{initial} = 0.9$ ,  $\omega_{final} = 0.4$ ,  $c_{1,initial} = 2.5$ ,  $c_{1,final} = 0.5$ ,  $c_{2,initial} = 0.5$ ,  $c_{2,final} = 2.5$ ,  $\Psi_{initial} = 0.9$ ,  $\Psi_{final} = 0.4$ ,  $iter_{max} = 250$ , accuracy level ( $acc$ ) = 1.0 (i.e. simulation will stop when each component of coefficient of variance of particles' position goes down below 1%).

As PSO is a population based searching technique, a proper choice of initial position vectors of the particles within specified search space and their corresponding velocities are to be set randomly. The wide spreading of initial position vectors must be assured by the operator for better searching. These set of values should be memorized by the computing machine for exact repetitiveness of the simulation process. Initial position and velocity vectors of the particles used in this study for modeling of MRR and Ra are listed in Tables A1 and A2 respectively.

Different combinations of  $C$ ,  $\varepsilon$  and  $\sigma$  change the shapes of model and best fitted model will have minimum fitting error i.e. minimum MAPE. As such the MAPE of the fitted models should be minimized. In this study, best fitted model with minimum MAPE is constructed on optimum combination of  $C$ ,  $\varepsilon$  and  $\sigma$  i.e. best position of the particles in swarm.

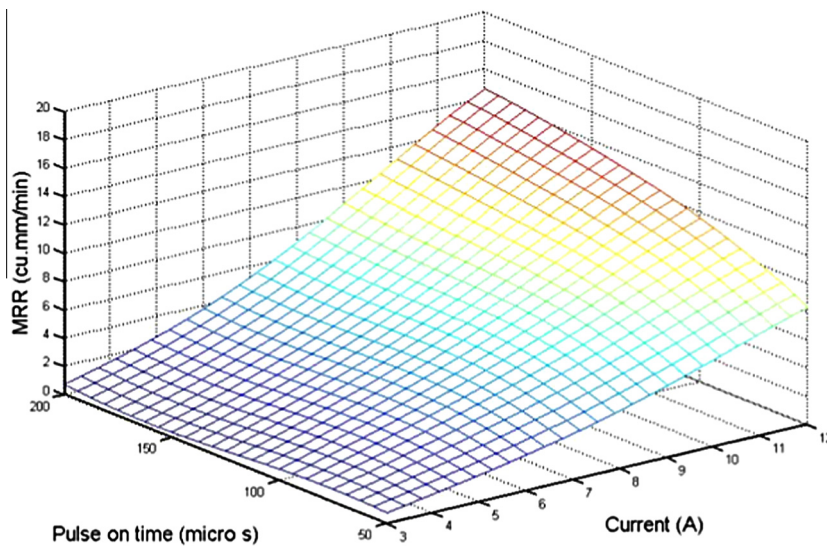


Fig. 9. Effect of current & pulse on time on MRR at pulse off time 125  $\mu$ s.

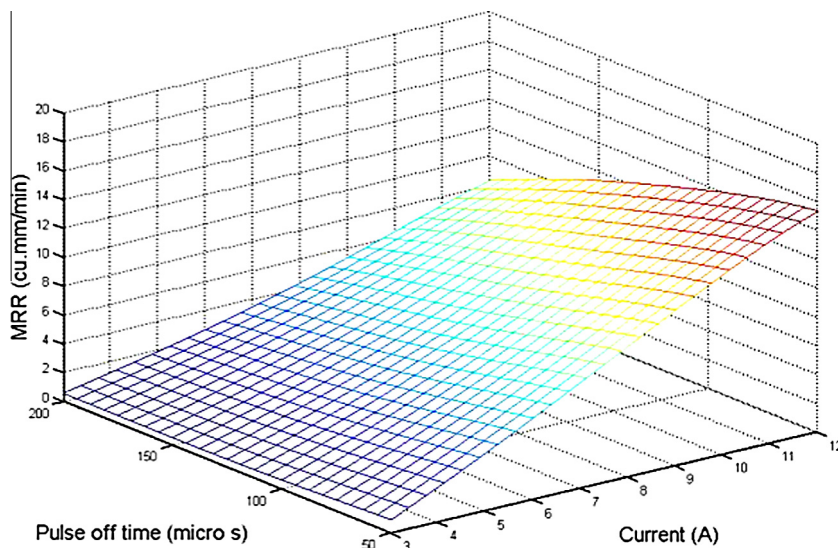


Fig. 10. Effect of current & pulse off time on MRR at pulse on time 125  $\mu$ s.

Thus optimizing MAPE by PSO with the initial position and velocity vectors, a set of  $C$ ,  $\varepsilon$  and  $\sigma$  for each of MRR and  $R_a$  is found (Table 3). Marching steps of the simulation process for minimizing MAPE are shown in Figs. 5 and 6. Though MAPE values for each of MRR and  $R_a$  remain unchanged after some iteration, yet the simulation is continued further. In this work, termination criterion is set by CV of position vectors' components along each dimension ( $C$ ,  $\varepsilon$  and  $\sigma$ ), as this will ensure proper convergence of the searching operation to optimum zone. So, iteration process is continued until coefficient of variation of position vector components in each dimension ( $C$ ,  $\varepsilon$  and  $\sigma$ ) go down below 1% (refer Figs. 7 and 8) in spite of MAPE remains unchanged. Coefficient of variation (expressed in%) of position vectors in different dimensions are calculated by taking the ratio of standard deviation to mean of position vector component along respective dimensions. Another statistical measurement, correlation coefficient of fitted model (based on training data) is also calculated. High value of  $r^2$  (Table 3) indicates adequacy of the fitted model for each of MRR and  $R_a$ .

Fig. 6 indicates that relative to  $C$  and  $\varepsilon$  the effect of  $\sigma$  is lower. Again  $\varepsilon$  is more influencing than  $C$ , as CV for  $\varepsilon$  decreases at a faster rate than that for  $C$ . This indicates that the model of MRR is complex and could efficiently capture small random noises. However, fluctuation outside the insensitive tube is very small as CV for  $\sigma$  reduces gradually. In case of  $R_a$  all three param-

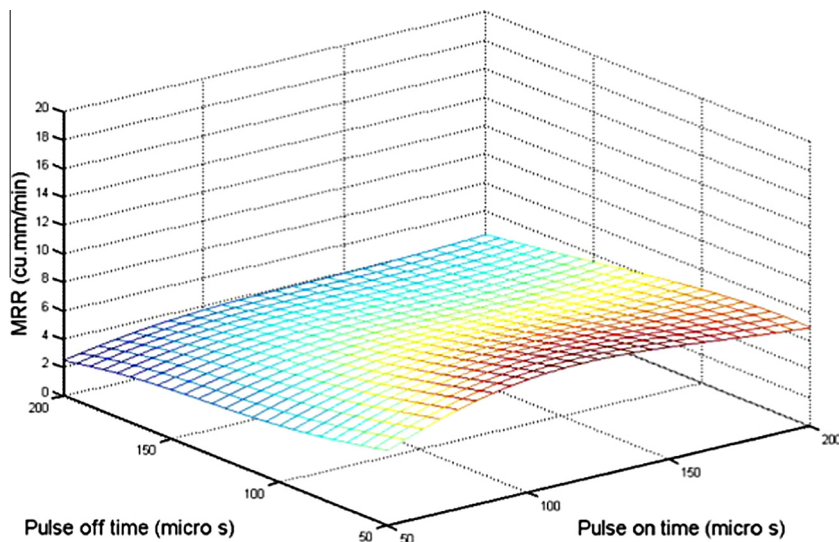


Fig. 11. Effect of pulse on time & pulse off time on MRR at current 7.5 A.

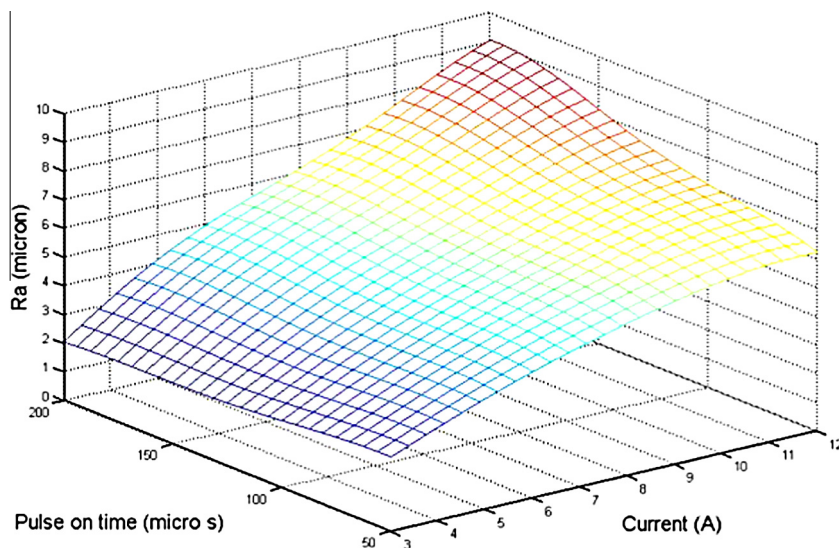


Fig. 12. Effect of current & pulse on time on  $R_a$  at pulse off time 125  $\mu$ s.

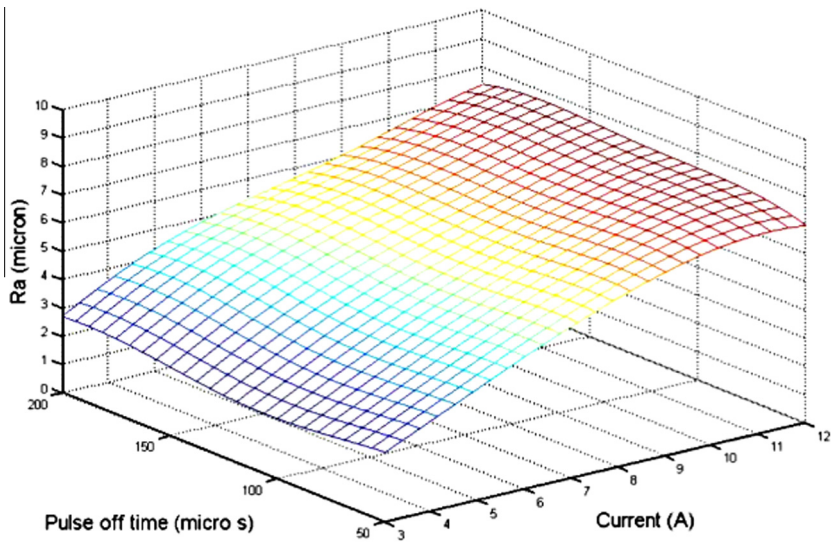


Fig. 13. Effect of current & pulse off time on Ra at pulse on time 125  $\mu$ s.

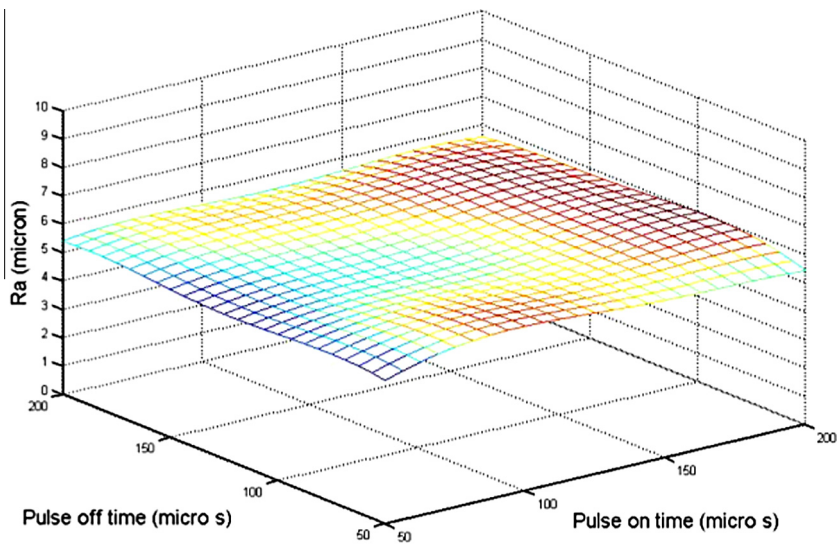


Fig. 14. Effect of pulse on time & pulse off time on Ra at current 7.5 A.

**Table 4**

Testing results for MRR.

Sl. No.	Current (A)	Pulse on time ( $\mu$ s)	Pulse off time ( $\mu$ s)	MRR ( $\text{mm}^3/\text{min}$ ) (Experimental)	MRR ( $\text{mm}^3/\text{min}$ ) (Estimated)	Absolute % error
1	3	100	150	0.7061	0.7667	8.5823
2	6	50	150	2.6178	2.4288	7.2198
3	6	100	150	3.5916	3.3688	6.2034
4	6	150	100	3.7811	4.0978	8.3759
5	9	100	150	7.5987	6.6644	12.2955
6	9	200	50	9.1232	9.7135	6.4703
7	12	100	150	10.8249	10.3329	4.5451
8	12	150	150	12.3972	12.1168	2.2618
9	12	200	50	17.2259	14.9725	13.0815
10	12	200	150	15.3902	13.5628	11.8738
Mean Absolute Percentage Error (MAPE) (Testing)						8.0909



**Table 5**  
Testing results for Ra.

Sl. No.	Current (A)	Pulse on time ( $\mu$ s)	Pulse off time ( $\mu$ s)	Ra ( $\mu$ m) (Experimental)	Ra ( $\mu$ m) (Predicted)	Absolute% error
1	3	100	150	2.48	2.55	2.82
2	6	50	150	4.59	4.56	0.65
3	6	100	150	4.81	4.79	0.41
4	6	150	100	4.77	4.65	2.52
5	9	100	150	6.24	6.18	0.96
6	9	200	50	5.34	6.38	19.48
7	12	100	150	7.28	6.96	4.40
8	12	150	150	7.18	8.32	15.88
9	12	200	50	8.35	7.73	7.42
10	12	200	150	7.79	9.06	16.30
Mean Absolute Percentage Error (MAPE) (Testing)						7.08

**Table A4**  
Initial position and velocity vectors for maximization of MRR.

Particle No.	Initial position vector (cur, $t_{on}$ , $t_{off}$ )	Initial velocity vector (cur, $t_{on}$ , $t_{off}$ )
1	(6.3502, 145.9835, 171.2663)	(4.5791, 137.0234, 33.5541)
2	(9.3106, 185.2821, 121.4341)	(7.7946, 33.8536, 147.5145)
3	(4.4774, 51.9548, 58.2878)	(7.1082, 138.0891, 41.2125)
4	(9.9477, 84.7907, 85.2986)	(0.0036, 35.0496, 128.0681)
5	(8.7414, 101.1892, 104.2100)	(4.5313, 18.8761, 13.6016)
6	(8.5953, 86.6074, 162.3100)	(5.5466, 138.8904, 90.4722)
7	(7.8880, 177.6428, 103.9217)	(4.0672, 128.2751, 21.6121)
8	(8.3930, 112.9951, 53.3238)	(2.2994, 139.0586, 146.5996)
9	(8.0157, 76.6659, 65.8305)	(4.1145, 144.1409, 69.0305)
10	(5.8737, 137.4766, 187.1970)	(8.9990, 125.3434, 16.8256)
11	(5.4772, 96.7670, 95.7951)	(0.4009, 37.2945, 29.7563)
12	(9.5189, 113.6890, 166.2264)	(6.0660, 127.1887, 39.7017)
13	(7.9797, 129.2688, 88.0619)	(4.9017, 87.5313, 15.0361)
14	(10.3557, 104.3300, 69.0317)	(0.1838, 129.8109, 65.2117)
15	(3.7653, 118.2075, 184.2592)	(6.4726, 55.4268, 30.4438)
16	(3.9005, 92.4084, 101.5861)	(3.6115, 88.6854, 6.9332)
17	(8.8224, 103.5622, 56.2338)	(3.1224, 44.7472, 131.4852)
18	(7.2804, 105.2064, 54.7977)	(0.9108, 101.7234, 88.4260)
19	(9.5905, 80.1031, 151.7571)	(4.6840, 5.1967, 114.9855)
20	(5.0791, 115.5322, 67.6604)	(4.1846, 33.0060, 60.2151)

**Table A5**  
Initial position and velocity vectors for minimization of Ra.

Particle No.	Initial position vector (cur, $t_{on}$ , $t_{off}$ )	Initial velocity vector (cur, $t_{on}$ , $t_{off}$ )
1	(7.7489, 163.4916, 196.8020)	(5.5590, 45.6519, 104.5374)
2	(7.1410, 104.6063, 194.4287)	(8.2420, 81.1565, 124.4670)
3	(4.2273, 158.7125, 144.4384)	(7.0003, 16.4060, 138.4989)
4	(5.1157, 193.9144, 69.2488)	(5.6365, 87.8402, 44.1032)
5	(7.2666, 182.9408, 66.7646)	(6.0385, 48.7033, 112.3988)
6	(5.0795, 184.0457, 150.4867)	(7.9285, 14.3711, 26.9754)
7	(3.5620, 196.8563, 179.6135)	(3.0543, 79.6444, 120.0053)
8	(3.1415, 56.9995, 175.1477)	(0.8026, 42.9514, 1.4097)
9	(5.9780, 175.7115, 164.2868)	(2.6199, 46.6177, 52.4645)
10	(10.5052, 187.3639, 117.3346)	(0.9002, 34.5273, 54.8004)
11	(3.9813, 172.0814, 188.8559)	(4.7930, 48.7678, 97.9479)
12	(10.1528, 81.1377, 110.3155)	(6.6592, 63.6175, 108.6162)
13	(3.0818, 125.3124, 199.3927)	(1.4635, 46.9031, 46.4831)
14	(10.2990, 126.3300, 149.3520)	(2.8364, 71.1978, 19.4216)
15	(3.6128, 138.2435, 132.0246)	(1.5889, 141.4391, 7.1967)
16	(7.3268, 92.6774, 102.2425)	(8.9296, 109.5821, 81.0179)
17	(4.4867, 98.0606, 170.9564)	(7.1420, 143.1209, 73.3404)
18	(6.4906, 168.6958, 163.3033)	(5.1396, 62.3674, 60.7684)
19	(7.1391, 171.8654, 144.1674)	(5.4354, 146.7743, 26.2091)
20	(10.7423, 150.5421, 164.7894)	(8.7100, 105.2290, 125.4884)



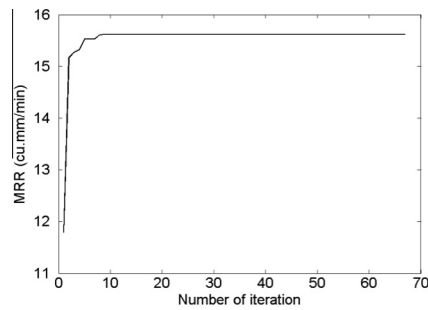


Fig. 15. Marching steps for maximization of MRR.

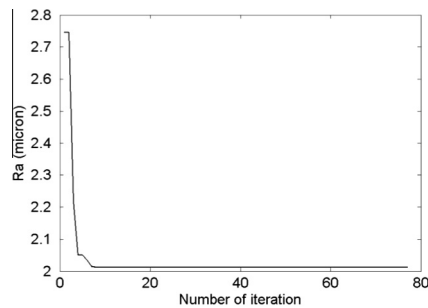


Fig. 16. Marching steps for minimization of Ra.

**Table 6**

Results of optimization.

	Simulation time (s)	Current (A)	Pulse on time ( $\mu$ s)	Pulse off time ( $\mu$ s)	Output
MRR ( $\text{mm}^3/\text{min}$ )	4024	12.0	153.9865	50.0000	15.6191
Ra ( $\mu\text{m}$ )	4644	3.0	200.0000	126.8332	2.0121

eters influence the model to almost the same extent i.e. complexity and generalization of the model are well-balanced (Fig. 7).

Optimum values of  $C$  and  $\varepsilon$  are found near the upper and lower bound of search space respectively. Higher value of  $C$  (with respect to specified search space) signifies that the models of MRR and Ra are highly complex in nature with large random noises and gives a good indication in favor of stochastic nature of EDM process. Again, random noises in the output of stochastic process are fairly controlled in developing the models by SVM with proper choice of  $\varepsilon$ . Process outputs within  $\varepsilon$ -tube do not greatly affect the complexity of model. Thus, lower values of  $\varepsilon$  for both MRR and Ra indicate that the models could capture the complexity of the EDM process adequately. Higher  $\sigma$  captures the oscillatory nature in training data which are not penalized by  $\varepsilon$ -insensitive zone. In case of MRR optimum  $\sigma$  shifts towards upper end, i.e. there is an oscillatory pattern outside the insensitive tube. But for Ra fluctuations in output is almost entrapped inside the insensitive zone.

With the estimated optimum combination of  $C$ ,  $\varepsilon$  and  $\sigma$  for each of MRR and Ra, Lagrange multipliers ( $\alpha_i, \alpha_i^*$ ) are calculated (Table A3) and the models for MRR and Ra thus developed are presented as

$$\text{MRR} : f(\mathbf{x}) = \sum_{i=1(1)N} (\alpha_i - \alpha_i^*) K(x_i, \mathbf{x}) + b \quad \left| \begin{array}{l} C = 32.2986 \\ \varepsilon = 0.1782 \\ \sigma = 0.6335 \end{array} \right. \quad (13)$$

$$\text{with } K(x_i, \mathbf{x}) = \exp(-||x_i - \mathbf{x}||^2 / 2\sigma^2) \quad \left| \begin{array}{l} \sigma = 0.6335 \end{array} \right.$$

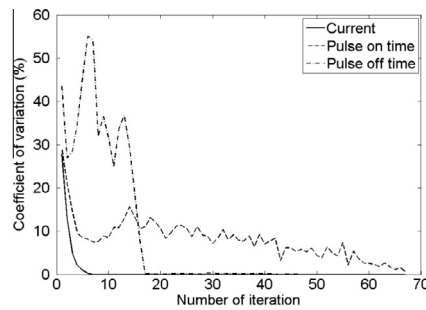


Fig. 17. Change of CV along different dimensions with iteration for maximization of MRR.

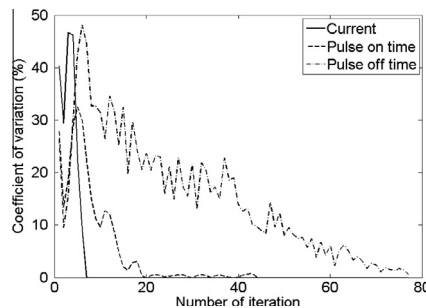


Fig. 18. Change of CV along different dimensions with iteration for minimization of Ra.

$$Ra : f(\mathbf{x}) = \sum_{i=1(1)N} (\alpha_i - \alpha_i^*) K(x_i, \mathbf{x}) + b \quad \left| \begin{array}{l} C = 17.0212 \\ \varepsilon = 0.1667 \\ \sigma = 0.4642 \end{array} \right. \quad (14)$$

$$\text{with } K(x_i, \mathbf{x}) = \exp(-||x_i - \mathbf{x}||^2 / 2\sigma^2) \quad \left| \begin{array}{l} \sigma = 0.4642 \end{array} \right.$$

To depict the effects of different control parameters (current, pulse on time and pulse off time) on responses, surface plots for MRR and Ra are generated using the above estimated models. The representative surface plots are shown in Figs. 9 through 14. Current, pulse on time and their interaction are observed as the common significant contributive factors to both the material removal rate and average surface roughness.

### 5.2. Testing of fitted model

Testing of the fitted models is performed with 10 disjoint data sets obtained from ten separate follow up experimental runs. The results of testing are shown in Tables 4 and 5. The values of mean absolute percentage error (8.0909 for MRR and 7.08 for Ra) suggest the adequacy of the models for application in practical field of work.

### 5.3. Optimization of MRR and Ra

Higher value of MRR increases production rate and lower value of Ra indicates good quality of the product. Hence for determining the optimum parameter combination of process parameters for maximum MRR and minimum Ra the respective models developed by support vector regression analysis are further searched by PSO. Particle swarm optimization is applied with the same control factor ( $n$ ,  $\omega$ ,  $c_1$ ,  $c_2$ ,  $\Psi$ ,  $iter_{max}$ , acc) settings. Initial position vectors and corresponding velocity vectors are randomly chosen within the specified search range (Table 1). They are listed in Tables A4 and A5 in the appendix.

Marching steps of the simulation process for maximization of MRR and minimization of Ra are shown in Figs. 15 and 16. The results of optimization are enumerated in Table 6.

Coefficient of variation specifically signifies the relative measurement of spreading of the particles in swarm. Though CV value is dependent on random numbers but the nature of change of this value reflects the physical relationship among parameters and responses to some extent. Fig. 17 indicates that the position vector component for current dies down rapidly (i.e. CV decreases at a faster rate) while there is a decay of the components for pulse on time and pulse off time over a wider range. As current increases, energy per spark and thereby crater size becomes greater. Thus small change in this control parameter leads to a substantial change in MRR value. As a result faster convergence of randomly chosen particles takes place. However, MRR is not affected much with variation of pulse on time and pulse off time. The relationships being less sensitive, lead to die down of scattering of particle positions (Fig. 17) at a larger value of number of iterations.

On the other hand, surface generated by EDM process becomes rough when energy applied to the process is high. Larger current and pulse on time cause high energy per spark. High energy in a spark applied over a longer time develops deeper crater and consequently higher surface roughness. Hence, components of particle's position vector along the control parameters current and pulse on time quickly jump to optimum zone but component of less influencing parameter pulse off time slowly moves to optimum region. Fig. 18 explains this gradual die down nature.

## 6. Conclusion

Material removal rate (MRR) and average surface roughness (Ra) respectively control the quantitative and qualitative aspect of the obtained product in any manufacturing process. Predictive models of those attributes are therefore, prerequisite during the product development stage and freezing the design before final production. Due to stochastic nature of EDM small scale random variation in responses is inevitable. SVM learned model developed with optimum parameter settings ( $C$ ,  $\varepsilon$  and  $\sigma$ ) can efficiently capture these small scale random fluctuations and can predict the responses - MRR and Ra in a robust way. Testing result of estimated models favors the practical use of these models in the chosen range. Optimum settings of control parameters of EDM for maximum MRR and minimum Ra are further found separately by applying PSO technique on the developed models. Also a new termination criterion, coefficient of variation (CV) of particles' position, to ensure global optimum in PSO is suggested. Thus, a procedure is proposed that may be used effectively to develop models for this stochastic type process and determine optimal setting of parameters.

## References

- [1] K.H. Ho, S.T. Newman, State of the art electrical discharge machining (EDM), in: *Int. J. Mach. Tools Manuf.* 43 (2003) 1287–1300.
- [2] W. König, D.F. Dauw, G. Levy, U. Panten, EDM future steps towards the machining of ceramics, *Ann. CIRP* 37 (2) (1988) 623–631.
- [3] M. Kunieda, B. Lauwers, K.P. Rajurkar, B.M. Schumacher, Advancing EDM through fundamental insight of the process, *Ann. CIRP* 54 (2) (2005) 599–622.
- [4] N.M. Abbas, D.G. Solomon, M.F. Bahari, A review on current research trends in electrical discharge machining, *Int. J. Mach. Tools Manuf.* 47 (7–8) (2007) 1214–1228.
- [5] A. Singh, A. Ghosh, A thermo-electric model of material removal during electric discharge machining, *J. Mater. Process. Technol.* 39 (1999) 669–682.
- [6] D.K. Panda, R.K. Bhoi, Artificial neural network prediction of material removal rate in electro discharge machining, *Mater. Manuf. Process* 20 (2005) 645–672.
- [7] G. Petropoulos, N.M. Vaxevanidis, C. Pandazaras, Modeling of surface finish in electro-discharge machining based upon statistical multi-parameter analysis, in: *J. Mater. Process. Technol.* 155–156 (2004) 1247–1251.
- [8] K.M. Tsai, P.J. Wang, Semi-empirical model of surface finish on electrical discharge machining, *Int. J. Mach. Tools Manuf.* 41 (2001) 1455–1477.
- [9] K.M. Tsai, P.J. Wang, Prediction on surface finish in electrical discharge machining based upon neural network models, *Int. J. Mach. Tools Manuf.* 41 (2001) 1385–1403.
- [10] D.D. DiBitonto, P.T. Eubank, M.R. Patel, M.A. Barrufet, Theoretical models of electrical discharge machining process. I. A simple cathode erosion model, *J. Appl. Phys.* 66 (1989) 4095–4103.
- [11] R. Perez, H. Rojas, G. Walder, R. Flukigar, Theoretical modeling of energy balance in electro erosion, *J. Mater. Process. Technol.* 149 (2004) 198–203.
- [12] P. Saha, A. Singha, S.K. Pal, P. Saha, Soft computing models based prediction of cutting speed and surface roughness in wire electro-discharge machining of tungsten carbide cobalt composite, *Int. J. Adv. Manuf. Technol.* 39 (1–2) (2008) 74–84.
- [13] S.S. Habib, Study of the parameters in electrical discharge machining through response surface methodology approach, *Appl. Math. Model.* 33 (2009) 4397–4407.
- [14] N. Cristianini, J.S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based-Learning Methods*, Cambridge University Press, 2000.
- [15] U. Çaydaş, S. Ekici, Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel, *J. Intell. Manuf.* 23 (3) (2012) 639–650.
- [16] R. Ramesh, K.S.R. Kumar, G. Anil, Automated intelligent manufacturing system for surface finish control in CNC milling using support vector machines, *Int. J. Adv. Manuf. Technol.* 42 (2009) 1103–1117.
- [17] B. Lela, D. Bajić, S. Jozić, Regression analysis, support vector machines, and Bayesian neural network approaches to modeling surface roughness in face milling, *Int. J. Adv. Manuf. Technol.* 42 (2009) 1082–1088.
- [18] L. Zhang, Z. Jia, F. Wang, W. Liu, A hybrid model using supporting vector machine and multi-objective genetic algorithm for processing parameters optimization in micro-EDM, *Int. J. Adv. Manuf. Technol.* 51 (2010) 575–586.
- [19] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
- [20] S.R. Gunn, *Support Vector Machines for Classification and Regression*, Technical report, University of Southampton, May 10, 1998.
- [21] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, *Stat. Comput.* 14 (3) (2004) 199–222.
- [22] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, in: *IEEE Comput. Intell. Mag.* 4 (2) (2009) 24–38.
- [23] P.S. Yu, S.T. Chen, I.F. Chang, Support vector regression for real-time flood stage forecasting, *J. Hydrol.* 328 (2006) 704–716.
- [24] A.A. Levis, L.G. Papageorgiou, Customer demand forecasting via support vector regression analysis, *Chemical Engineering research and Design* 83 (A8) (2005) 1009–1018.
- [25] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, vol. IV, Piscataway, NJ, 1995, pp. 1942–1948.
- [26] X. Hu, R. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, 2002, pp. 1677–1681.

- [27] M. Clerc, J. Kennedy, The particle swarm: explosion stability and convergence in a multi-dimensional complex space, *IEEE Trans. Evol. Comput.* 6 (2002) 58–73.
- [28] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Inf. Sci.* 177 (2007) 5033–5049.
- [29] Y. Tang, Z. Wang, J. Fang, Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm, *Expert Syst. Appl.* 28 (2011) 2523–2535.
- [30] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, proceedings of 7th international conference on evolutionary programming VII, LNCS, 1447, Springer-Verlag, New York, 1998, pp. 591–600.
- [31] A. Ratnaweera, S.K. Halgamure, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Trans. Evol. Comput.* 8 (2004) 240–255.
- [32] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, *Neural Networks* 17 (2004) 113–126.