J Intell Manuf (2012) 23:1833–1847
DOI 10.1007/s10845-010-0498-9

A robust approach for root causes identification in machining processes using hybrid learning algorithm and engineering knowledge

Shichang Du · Jun Lv · Lifeng Xi

Received: 2 June 2010 / Accepted: 13 December 2010 / Published online: 28 December 2010

Abstract To improve product quality and productivity, one of the most critical factors for most manufacturers lies in quickly identifying root causes in machining process during ramp-up and production time. Though multivariate statistical process monitoring techniques using control charts have been successfully used to detect anomalies in machining processes, they cannot provide guidelines to identify and isolate root causes. One novel robust approach for root causes identification (RCI) in machining process using hybrid learning algorithm and engineering-driven rules is developed in this study. Firstly, off-line pattern match relationships between fixture fault patterns and part variation motion patterns are derived. Then, a hybrid learning algorithm is explored for identifying the part variation motion patterns. An unknown root cause is identified and isolated using the output of hybrid learning algorithm and engineering-driven rules. Finally, the data from the real-world cylinder head of engine machining processes are collected to validate the developed approach. The results indicate that the developed approach can perform effectively for identifying root causes of fixture in machining processes. All of the analysis from this study provides guidelines in developing root causes identification systems based on hybrid learning algorithm and engineering knowledge.

Keywords Root causes identification · Hybrid learning algorithm · Engineering-driven rules

S. Du (✉) · L. Xi
Department of Industrial Engineering and Logistics Engineering, School of Mechanical Engineering, Shanghai Jiaotong University, 200240 Shanghai, China
e-mail: lovbin@sjtu.edu.cn

J. Lv
School of Business, East China Normal University, 200241 Shanghai, China

Introduction

One of the most critical factors and barriers in the development and operation of modern machining processes lies in largely identification of root causes during ramp-up and production time (Du et al. 2008; Huang et al. 2008). Identifying root causes as quickly as possible can greatly improve product quality and productivity. Though multivariate statistical process control (MSPC) using control charts can effectively monitor anomalies in machining processes (Montgomery 2005; Ertuğrul and Aytaç 2009), they cannot provide guidelines to identify and isolate the root causes.

In Fig. 1, if position or diameter of fixture locator deviates from the nominal design in a machining process, then, consequently, the machined part will not be in its nominal design position. Mislocations of the locator can be manifested by mean shift or variance change in the product measurement data. For example, the variance change of the product measurement data can be caused by a variation of the location of a locator, due to locator worn-out or the excessive looseness of locator. The quality variables measurements are collected and input into multivariate control charts. If out-of-control signals are shown in control charts, the locator of fixture deviating from the nominal design, called "root cause", cannot be easily identified for most of the cases. Though multivariate control charts are widely used as effective tools to detect an unusual event, they do not directly provide the information required by a practitioner to identify the root causes.

The available root causes identification (RCI) methods in manufacturing processes can be roughly divided into three classes: (i) engineering-model-based methods, (ii) knowledge-based methods, and (iii) intelligent-learning-based methods. In engineering-model-based methods, an engineering mathematical model integrating product quality information and root causes information firstly is built, and then

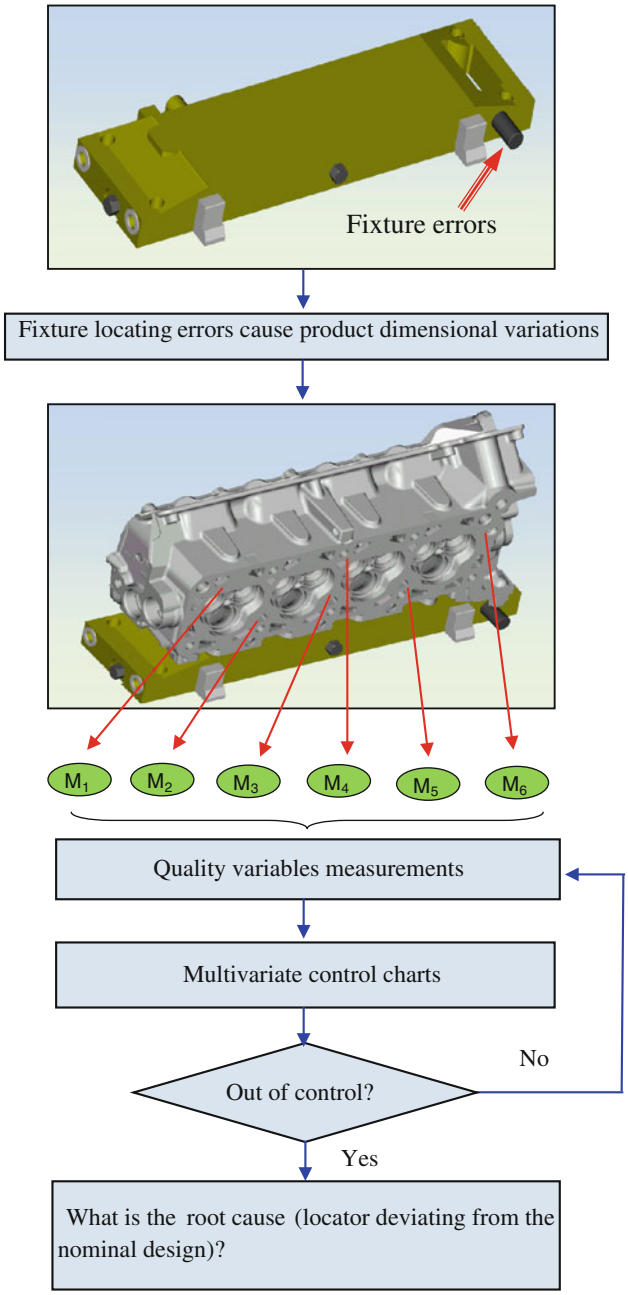


Fig. 1 Root causes in a machining process

the variances of root causes are estimated in the engineering model (McCulloch and Searle 2001; Zhou et al. 2004). However, the effectiveness of these methods relies heavily on the accuracy of the model. Moreover, the engineering-model-based methods need a thorough understanding of the physics of the process to build the process model, which is usually very difficult, if not impossible, for a complex system (Shi and Zhou 2009).

As for the knowledge-based methods, some research efforts are also reported. A systematic method of identifying failing stations and faulty parts in assembly processes is described (Ceglarek et al. 1994). A root causes diagnostic reasoning and decision approach through the combination of data mining and knowledge discovery techniques is developed (Lian et al. 2002). However, the prior knowledge for the knowledge-based methods has to be collected.

Recently, intensive research has been conducted into the use of neural networks (NNs) as an effective tool to improve product quality since neural networks (NNs) have excellent noise tolerance in real time, requiring no hypothesis on statistical distribution of monitored measurements. These important features make NNs promising and effective tools that can be used to improve data analysis in manufacturing quality control applications. Neural networks are utilized for pattern recognition in quality control charts (Bargash and Santarisi 2004; Jiang et al. 2009). After detecting such patterns, it is possible to relate these patterns to their root causes before the process may produce defective parts. Research efforts are also devoted to detect the changes in mean and/or variance of product quality using neural networks through classifying out-of-control signals in control charts (Das and Prakash 2008; Guh 2007; Wang and Chen 2002). Though these methods can detect the changes of patterns in control charts or changes in mean and/or variance of product quality, they cannot identify the root causes causing these changes.

Therefore, in this study a novel robust approach for RCI in machining processes is developed, which integrates product and fixture design, on-line dimensional measurements, neural networks ensemble technique as well as engineering-driven rules. The developed method is presented in three parts. Firstly, off-line patterns relationships between the fixture fault patterns and part variation motion patterns are derived based on engineering knowledge including fixture layout, measurement information, and product design. Once the part variation motion patterns using control chart are recognized, the unknown root causes in fixture can be identified and isolated quickly through the off-line patterns relationships. However, recognition of the part variation motion patterns is not an easy problem to solve. Due to high recognition power of NN ensembles (Breiman 1996; Hansen and Salamon 1990; Zhou et al. 2002), in the second part, an improved Particle swarm optimization with Simulated annealing-based selective Neural network ensemble (PSN) algorithm is developed for classifying the part variation motion pattern triggering the out-of-control signals of control chart. Utilization of selective NN ensemble based on particle swarm optimization with simulated annealing aims to improve the generalization performance of neural-system in comparison with using single NN recognizers, and to improve the ability to escape from a local optimum. Finally, an unknown root cause in fixture is identified and isolated using the output of PSN algorithm and engineering-driven rules.

The rest of this paper is organized as follows. Section "Overview" presents the overview of the methodology. The

patterns relationships between fixture fault patterns and part variation motion patterns are derived in sect. "Patterns relationships". PSN algorithm is explored in sect. "PSN algorithm". The identification procedure using output of PSN algorithm and engineering-driven rules is developed in sect. "Identification procedure" and an illustrated example from real-world is illustrated and the performance of the explored approach is analyzed and evaluated in sect. "Case study". Finally, the conclusions are given in sect. "Conclusions and discussions".

Overview

This section provides a practical overview with respect to identifying root causes in fixture in machining processes. This study is concerned with product variability increase with assumption that the product mean is not changed. Assume there are q part dimensional measurement points for a workpiece $X_1, X_2, \ldots X_q$ need to be monitored, which is expressed as one vector $\mathbf{X} = (X_1, X_2, \ldots X_q)$. When increases δ occur at time t_p, the observations of measurement points vector $\mathbf{X}$ can be expressed as follows:

$$\mathbf{X}(t) = \boldsymbol{\mu}_0 + \mathbf{K}(t) + \boldsymbol{\delta}(t, t_p) \tag{1}$$

where

- $\mathbf{X}(t)$: part dimensional measurement points vector observed at time t;
- $\boldsymbol{\mu}_0$: the mean of measurement points vector, $\boldsymbol{\mu}_0 = (\mu_1, \mu_2, \ldots \mu_q)$ when the process is under control, which is assumed as one constant in this study;
- $\mathbf{K}(t)$: $N(0, \boldsymbol{\Sigma}_0)$, multivariate normal distribution;
- $\boldsymbol{\Sigma}_0$: the covariance of measurement points vector when the process is under control;
- $\boldsymbol{\delta}(t, t_p)$: $(\delta_{X_1} + \sigma_{X_1}, \delta_{X_2} + \sigma_{X_2}, \ldots\ldots \delta_{X_q} + \sigma_{X_q})$, δ_{X_j} is the magnitude of increases in terms of σ_{X_j}, which is the standard deviation of jth measurement point.

For example, in a bivariate process, $\mathbf{X}(t)$ follows $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ when the process is under control. The covariance matrix $\boldsymbol{\Sigma}_0$ can be expressed:

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1X_2} \\ \sigma_{X_1X_2} & \sigma_{X_2}^2 \end{bmatrix} \tag{2}$$

There are three abnormal classes of patterns for a bivariate process: (1) the variance of first measurement point is increased $(\delta_{X_1} + \sigma_{X_1}, \sigma_{X2})$; (2) the variance of second measurement point is increased $(\sigma_{X_1}, \delta_{X2} + \sigma_{X2})$; (3) the variance of both two measurement points is increased $(\delta_{X_1} + \sigma_{X_1}, \delta_{X2} + \sigma_{X2})$.

The architecture of the robust RCI approach in machining process is shown in Fig . 2, in which four modules are in series: Module I, Module II, Module III, and Module IV. Module design divides the complex original problem into more manageable sub-problems, which are solved effectively by using different sub-systems.

Module I builds the off-line patterns relationships between the fixture fault patterns and the part variation motion patterns. When an unknown fixture fault occurs in a machining process, it usually causes one or some part variation motion patterns. In order to identify the root cause in fixture, it firstly needs to on-line monitor the part dimensional measurements and identify the part variation motion patterns. Therefore, Module II is used for monitoring the part dimensional measurement points and judging whether the process is under control. Montgomery (2005) evaluated several statistics for controlling the variances in a process, and took $|S|$ control method as the one of most excellent control chart for process variability. Therefore, $|S|$ control chart is used in this study for monitoring the variance of part dimensional measurement points.

Module III is responsible for identifying the part variation motion patterns. When an out-of-control signal is detected by $|S|$ control chart, PSN will wait for more observations (i.e. half of the size of the moving window) from the process and then identify the part variation motion patterns triggering the out-of-control signals. The purpose of this procedure is to include more abnormal points in the recognition window, so that more abnormal features can be exposed to the recognizer. The size of an analytical window determines the effect of neural networks on monitoring the abnormal variations. The more abnormal points are included in the recognition window, the more accurate the recognition will be. Ideally, waiting for the points of the whole size of the window will guarantee that a full abnormal pattern is presented to PSN. However, for monitoring and identifying out-of-control signals, a very important aim is to identify abnormal signals as soon as possible. Thus, waiting for several observations (i.e., half of the size of the moving window) is suitable for this aim. Module IV builds the identification procedure, which identify root causes using output of PSN and engineering-driven rules.

Patterns relationships

Fixture fault patterns

Fixtures are used to locate and hold workpiece in machining processes, thus fixture failure can directly affect part location and the final product dimensional quality. In general, fixture elements can be classified by their functionality into locators and clamps. Locators establish the datum

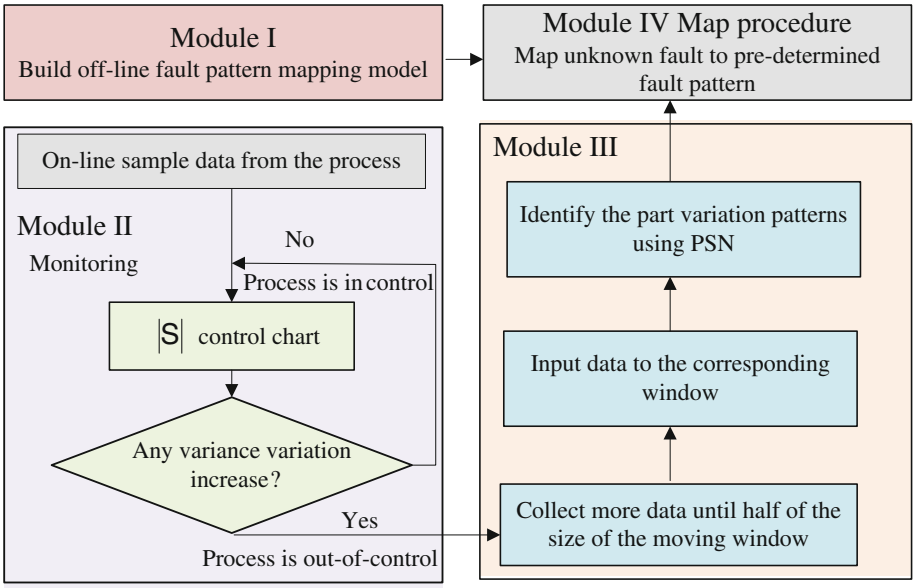


Fig. 2 RCI based on engineering knowledge and PSN

reference frame and provide kinematic restraint. Clamps provide additional restraint by holding the part in position under the application of external forces during the machining process.

For a rigid part, the most common fixture layout method is 3-2-1 principle, which locates a part by three groups of locators laid out in two orthogonal planes. One 3-2-1 fixture and measurement layout is shown in Fig. 3. A four-way pin $P_1(x_{P_1}, y_{P_1}, z_{P_1})$ is used to precisely position the part in two directions (X and Y direction), and a two-way pin $P_2(x_{P_2}, y_{P_2}, z_{P_2})$ is used to locate the part in one direction (Y direction) in the first plane, respectively. All remaining NC blocks (C_1, C_2 and C_3) are used to locate the part in the second plane (Z direction). Assume all measurement gages use the same Cartesian coordinate system, which makes it easy to use and compare data from different gages. The manifestation of the pins P_1, P_2 and/or NC blocks failures is represented by part dimensional measurement points $M_1(x_{M_1}, y_{M_1}, z_{M_1})$, $M_2(x_{M_2}, y_{M_2}, z_{M_2})$, $M_3(x_{M_3}, y_{M_3}, z_{M_3})$ and their standard deviations $\sigma_{M_1}, \sigma_{M_2}$and σ_{M_3}. For example, pin P_2 controls part motion in the Y direction. Failure at P_2 can be represented as a rotation of the part around $P_1(P_2'(x_{P_2'}, y_{P_2'}, z_{P_2'})$ in Fig. 3), thus M_1, M_2 and M_3 will also rotate around Z axis and have standard deviations $\sigma_{M_1}, \sigma_{M_2}$ and σ_{M_3}.

Table 1 summarizes the six hypothetical fixture fault patterns, in which the arrows represent the direction of part mislocation due to the fault of P_1, P_2 and NC blocks. There are two kinds of motion for failing locators: translation and rotation. The failing pin P_1 in X axis makes the fixture translate in X axis, failing pin P_1 or P_2 in Y axis makes the fixture rotate, and the failing NC block in Z axis makes the fixture rotate around the line between the remaining two NC blocks, respectively.

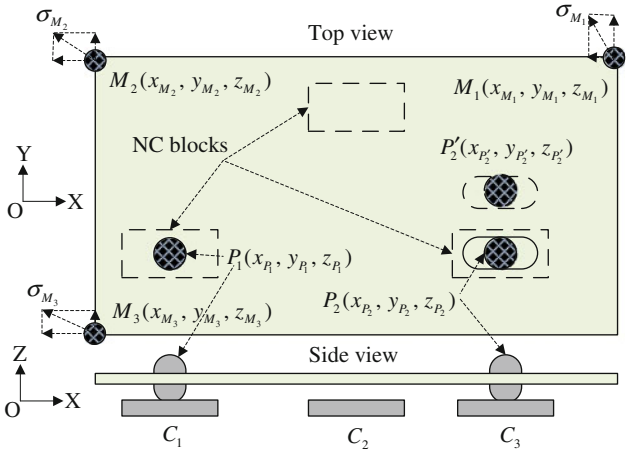


Fig. 3 A 3-2-1 fixture and measurement layout

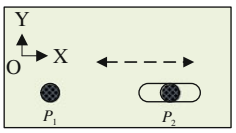
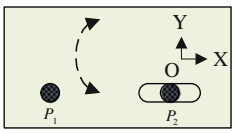
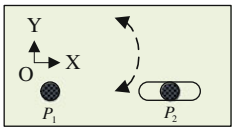
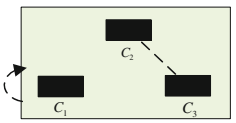
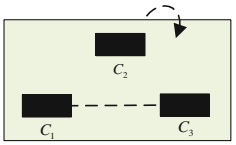
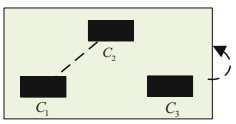
Part motion variation patterns

In a general case, all rigid body part variation motion can be described through six patterns (as shown in Fig. 4), i.e., translations along X, Y, and Z axis (T_x, T_y, and T_z), and rotations around X, Y, and Z axis (R_x, R_y, and R_z), respectively. These six patterns are generic and complete, namely, all rigid part variation motions can be described through these six patterns or their combinations, regardless of the measurements layout. More important, these patterns are mutually orthogonal, which make it possible to uniquely identify them from one another even if they coexist.

Patterns map process

If significant variation motion patterns of T_z and R_y with a strong negative correlation between their components,

Table 1 Relationships between fixture fault patterns and variation manifestation

Fault patterns	Failing locator	Variation manifestation
Type-1 F_1	Failing P_1 in X axis	
Type-2 F_2	Failing P_1 in X axis	
Type-3 F_3	Failing P_2 in Y axis	
Type-4 F_4	Failing C_1 in Z axis	
Type-5 F_5	Failing C_2 in Z axis	
Type-6 F_6	Failing C_3 in Z axis	

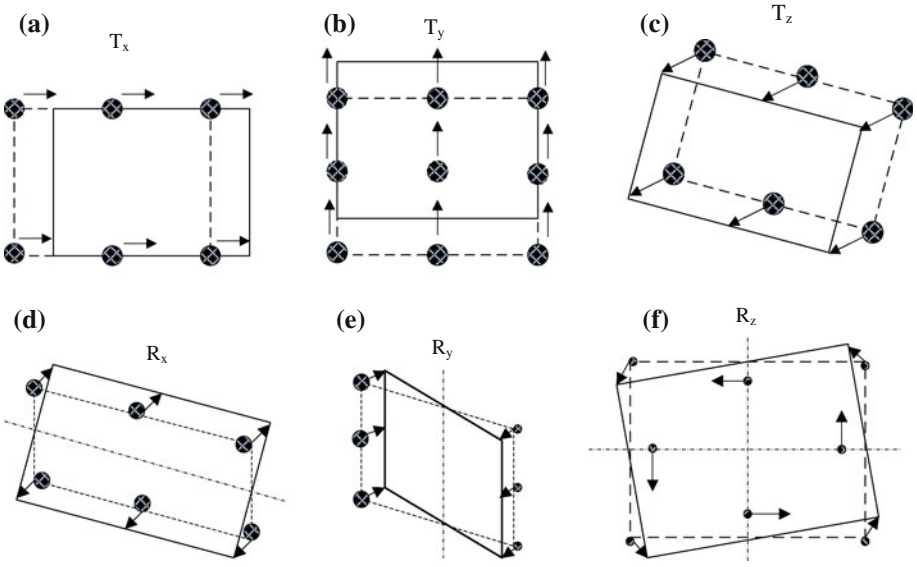


Fig. 4 Variation motion patterns for rigid body part

geometrically it means that whenever the part has a translation in the Z direction, and it rotates around the Y axis counterclockwise. These two simultaneous motions combined are actually equivalent to varation pattern F_4 (refer to Table 1, Figs. 3, 4). As studied by Liu and Hu (2005), this pattern map process can be described as follows:

$$T_z \oplus R_y \oplus Cor\left(T_z, R_y^{-1}\right) \Rightarrow F_4 \tag{4}$$

where $\oplus$ represents logic operator "AND", $Cor\left(T_z, R_y^{-1}\right)$ denotes a strong correlation between the patterns T_z and inverse R_y, and R_y^{-1} is inverse of R_y, respectively. With similar inference, all pattern map relationships between rigid body part variation motion patterns and fixture fault patterns can be found as follows:

$$T_x \Rightarrow F_1 \tag{5}$$

$$R_z \oplus T_x \oplus T_y \oplus Cor\left(T_x, T_y\right) \oplus Cor\left(T_x, R_z\right) \oplus Cor\left(T_y, R_z\right) \Rightarrow F_2 \tag{6}$$

$$R_z \oplus T_x \oplus T_y \oplus Cor(T_x, T_y) \oplus Cor\left(T_x, R_z^{-1}\right) \oplus Cor\left(T_y, R_z^{-1}\right) \Rightarrow F_3 \tag{7}$$

$$R_x \oplus T_z \oplus Cor\left(T_z, R_x^{-1}\right) \Rightarrow F_5 \tag{8}$$

$$R_x \oplus R_y \oplus Cor\left(R_x, R_y\right) \Rightarrow F_6 \tag{9}$$

The map model builds the exact relationships between part variation motion patterns and fixture fault patterns. Thus, in order to identify the root cause in fixture, one needs firstly to analyze the part variation patterns through on-line measurements of parts.

PSN algorithm

NN ensemble framework

NN ensemble originating from Hansen and Salamon's work (Hansen and Salamon 1990) is a learning paradigm where a collection of several NNs is trained for the same task. It shows significantly improved generalization performance, which outperforms those of single NNs (Breiman 1996; Zhou et al. 2002). Since this technology behaves remarkable well, NN ensemble has been widely studied and already been successfully applied to various areas such as medical diagnosis (Zhou et al. 2002), weather prediction (Maqsood et al. 2004), and face recognition (Gutta and Wechsler 1996), etc. However, no researches have been conducted to apply NN ensemble techniques into RCI in fixture in machining processes.

Figure 5 illustrates the basic framework of an NN ensemble used in this study. Instead of attempting to design an ensemble of independent NNs directly, several "candidate" NNs are initially created. Given such some networks, through

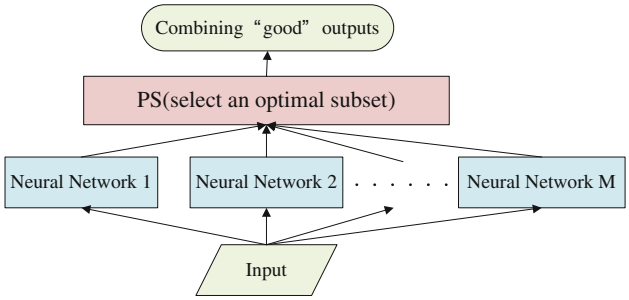


Fig. 5 Basic framework of NN ensemble

using an improved Particle swarm optimization algorithm with Simulated annealing (PS) technique, PSN aims to select the optimal subset formed by accurate and diverse networks and obtain a better ability to escape from the local optimum. Then, the predicted outputs of each of these NNs are combined to produce the output of the ensemble.

PSN design

The detailed design algorithm of PSN consists of three major steps: creation of "candidate" networks by using Bagging method, selection of subset from the "candidate" networks by using PS algorithm, and combining predictions of component networks in the ensemble. PSN can be described in Fig. 6, and the details of *PS technique* is explained further in the following sub-sections.

"Candidate" NNs creation

Creation stands for the process of creating a number of different individual networks, which then constitutes the pool of available networks. For ensemble to be effective, the component NNs in the ensemble should be as accurate and diverse as possible. There are a number of different ways to achieve this, for example, training networks by varying different initial conditions, varying the topology or algorithm involved. The most prevailing approaches are Bagging and Boosting. Bagging is proposed based on bootstrap sampling (Breiman 1996). It generates several training sets from the original training set and then trains a component neural network from each of those training sets. Boosting is proposed in (Schapire 1990). It generates a series of component neural networks, in which training sets are determined by the performance of former ones. Training instances that are wrongly predicted by former networks will play more important roles in the training of later networks. In this study, Bagging method is used for creating "candidate" networks.

Fig. 6 Design algorithm of PSN

Input: Training data set T, validation data set V, learner f, trials N, parameters of PS.
Procedure:
For i=1 to N {
T_i=Pre-stage's T (i>1) ∪ {examples bootstrap sample from T}
f_i=f(T_i) is trained and then is involved
T=Some examples of T_i, which are predicted wrongly by f_i
}
Implement *PS algorithm* to obtain the optimal particle vector v on the validation data set V.
where the fitness function is *fitness*.
Output: ensemble f^{opt}

$$f^{opt}(x) = \arg\max \frac{1}{M} \sum_{v_i=1, f_i(x)=y} 1(y \in Y) \ (M \text{ NN are selected for ensemble})$$

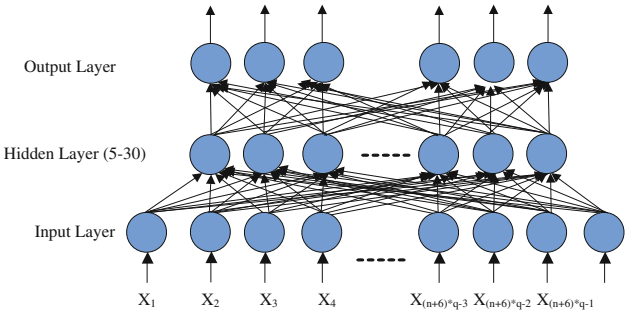


Fig. 7 The architecture of component NNs in the PSN

Component NNs design

Figure 7 shows the architecture of a component back-propagation neural network (BPN) in PSN. The BPN, which has been successfully used in many applications in the domain of NNs for MSPC (Guh 2007; Smith 1994), is used as the component NNs of PSN. There are three layers in the every component NNs: input layer, hidden layer, and output layer. An input to the neural system should consist of a time-series window vector. A window with more than one size can potentially provide the required data for identification of abnormal patterns.

Once an out-of-control signal is detected by $|S|$ statistic with type I error being equal to 0.5%, more abnormal data points will be waited for (i.e., half of the size of the moving window in this study) from the process to generate a moving window vector. These vectors consisting of raw observations, and six statistical feature elements are used as the training data sets of PSN (input data feature will be discussed in section "Input data features"). For each measurement point, the size of the moving window is n (i.e., n observations are sampled overtime series for each measurement point), and the number of statistical features is 6, thus, the number of input neurons in PSN is $(n+6) \times q$, where q is the number of part dimensional measurement points.

The number of neurons in the hidden layer varies randomly from 5 to 30 when creating each "candidate" NN in this study. The outputs of component NNs represent the different part variation motion patterns. According to sect. "Part motion variation patterns", there are six output neurons, which represent T_x, T_y, T_z, R_x, R_y, and R_z, respectively. There are totally 64 (2^6) possible output results. The desired output values in the training sample are represented by 1 and 0, which indicate that the pre-determined part variation motion patterns appear or do not appear, respectively. However, note that there are some cases, where the output values of trained NN are close to either zero or one, and the real value varies between 1 and 0, but is not exactly equal to 0 or 1. Therefore, a threshold, which is usually called an activation cutoff, γ_0, is used to determine whether the actual output value is sufficiently close to the target value. If the output value is equal to or larger than γ_0, the output is 1. On the other hand, an output value that is smaller than γ_0 indicates that the output is 0.

The choice of cutoff value usually is based on the considerations of Type I and Type II errors. In statistical hypothesis testing, there are two types of incorrect conclusions or errors that can be drawn. If a null hypothesis is incorrectly rejected, when it should in fact be accepted, it is called a Type I error (also known as a false positive). A Type II error (also known as a false negative), occurs when a null hypothesis is incorrectly accepted when it should in fact be rejected. As the cutoff value is closer to 1, the network will result in a smaller Type I error and a larger Type II error. On the other hand, smaller cutoff values result in a larger Type I error and smaller Type II error. It is possible to use different cutoff values for upper and lower one-sided tested, however, in order to reduce the complexity of the determination of process status, the same cutoff value for upper and lower tests usually is used. There are no absolute rules for setting the cutoff value (Cheng and Cheng 2001; Shanker and Hu 1996). If the diagnosis scheme is very effective in reducing Type I error, the cutoff value can be set closer to 1 and vice versa. In this study, the value of 0.5 is used as the cutoff due to two reasons. First

one is that the cutoff value 0.5 resulted in the good percentage of correct identification for PSN algorithm during the training process, and the second one is that 0.5 is usually the appropriate cutoff value for a practitioner with a relatively big dataset (Shanker and Hu 1996). A total of 5,016 samples were collected in this study.

Selection of "Good" NNs using PS technique

Once a number of trained NNs are obtained, PSN will select several NNs from the "candidate" NNs to construct the NN ensemble. Both theoretical and empirical studies demonstrate that ensemble several "good" NNs with accuracy and diversity could be better than ensemble all NNs for both regression and classification problems (Zhou et al. 2002). Furthermore, the size of the ensemble is reduced without worsening the performance. However, excluding those "bad candidate" and selecting the optimal subset from "candidate" NNs is not an easy task as we may have imagined since the space of possible subsets is very large $(2^z - 1)$ for a population of size z (z is the number of "candidate" NNs). Thus, it is impractical to use exhaustive search to find an optimal subset. There is still no agreement about which method is the most appropriate for selection of NNs. One possible method is to order the population of trained NNs in terms of increasing mean squared error, and to create an ensemble by including those with the lowest error (Perrone and Cooper 1993).

Some intelligent evolution algorithms show distinguished performance for solving these problems. Particle swarm optimization (PSO) is a population based search algorithm, which is inspired by the social behavior of bird flocks as originally developed by Eberhart and Kennedy (1995). It is widely reported that the PSO algorithm is very easy to implement and has fewer parameters to adjust when compared to other evolutionary algorithms. Some researchers use PSO to train neural networks and find that PSO-based ANN has a better training performance, faster convergence rate, as well as a better predicting ability than BP-based ANN (Lu et al. 2003; Zhou et al. 2002).

In this study, for PSN, one improved PS technique is used to obtain an optimal subset from "candidate" NNs. Each particle is "flown" through the multidimensional search space, adjusting its position in search space according to its own experience and that of neighboring particles. The particle therefore makes use of the best position encountered by itself and that of its neighbors to position itself toward an optimal solution. The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimization problem.

It is an iterative process in which the change in weights of a particle at the beginning of an iteration are calculated using Eq. (14) and new position of every particle is found using Eq. (15).

$$v_i(t+1) = wv_i(t) + c_1r_1(p_{id} - x_i(t)) + c_2r_2(p_{gd} - x_i(t)) \tag{14}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{15}$$

where t is the current step number, w is the inertia weight, c_1 and c_2 are the acceleration constants, r_1 and r_2 are two random numbers in the range [0, 1], $x_i(t)$ is the current position of the particle, p_{id} is the best one of the solutions this particle has reached, p_{gd} is the best one of the solutions all the particles have reached.

For PSN algorithm, simulated annealing (SA) is used to improve the ability to escape from a local optimum. SA is normally introduced as a heuristic approach to solve numerous combinatorial optimization problems to replace those schemes where the solution could get stuck on local optimum. In the search process, the SA accepts not only better but also worse neighboring solutions with a certain probability. Such mechanism can be regarded as a trial to explore new space for new solutions, either better or worse. The relative success of SA algorithm has been tapped by many researchers (Pandey et al. 2006; Yen et al. 2004).

In this study, the SA technique is used to deal with every particle according to the following two steps.

(1) If $p_{id} > p_{gd}$, accept $p_{gd} = p_{id}$ with the probability 1
(2) If $p_{id} < p_{gd}$, accept $p_{gd} = p_{id}$ with the probability *prob* defined as:

$$prob = 1 - \exp\left(-\frac{p_{gd} - p_{id}}{temp}\right) \tag{16}$$

$$temp = initemp \times h \tag{17}$$

where *prob* is the probability function, *temp* is the current temperature, *initemp* is a constant selected as initial temperature, p_{id} is the best one of the solutions this particle has reached, p_{gd} is the best one of the solutions all the particles have reached, and h is the current step number.

In the optimization process, for each of particle, the probability of making the transition from the current state S to a candidate new state S' is specified by an acceptance probability function $prob(e, e', T)$, that depends on the energies $e = E(s)$ and $e' = E(s')$ of the two states, and on a global time-varying parameter T called the temperature (*temp* is the current temperature and *initemp* is the initial temperature). One essential requirement for the probability function $prob$ is that it must be nonzero when $e' > e$ (i.e., $p_{id} > p_{gd}$), meaning that the system may move to the new state even when it is worse (has a higher energy) than the current one. This feature means that the SA technique makes it possible to jump out of a local optimum to search for the global optimum. On the other hand, when temperature T goes to zero,

the probability $prob(e, e', T)$ must tend to zero if $e' > e$ (i.e., $p_{id} > p_{gd}$), and to a positive value if $e' < e$ (i.e., $p_{id} < p_{gd}$). The probability $prob(e, e', T)$ is defined as 1 when $e' < e$ (i.e., $p_{id} < p_{gd}$), i.e., the procedure always moved downhill when it found a way to do so, irrespective of the temperature T. That way, for sufficiently small values of temperature T, the swarm will increasingly favor moves that go to lower energy values. Given these properties, the evolution of the state S depends crucially on the temperature T. The probability of accepting a worse solution is larger at higher initial temperature. As the temperature decreases, the probability of accepting worse solutions gradually approaches zero.

The above treatment can increase the diversity in the particles and enable PSO to accept a bad solution with probability, which will gradually decrease to zero as the temperature T increases. In PSN algorithm, the particles are encoded as binary strings with length N where in each elements of each particle indicates the presence (1) or absence (0) of NNs in the ensemble. Through continuous evolution of these particles to minimize the generalization error, the most optimal global solution (i.e., the most optimal particle vector) can be obtained.

Combination of outputs of component NNs

As for combining the predictions of component neural networks, the most prevailing approaches are plurality voting or majority voting for classification tasks (Hansen and Salamon 1990), and simple averaging or weighted averaging for regression tasks (Zhou et al. 2002). In this study, the majority voting method is used to RCI in fixture.

Input data features

The selection of the input features in training dataset significantly affects the performance of an NN identifier. The input feature vector must be able to strengthen the pattern feature of the dataset. Hassan et al. (2003) conducted an experimental study for identifying six types of basic statistical patterns, where the performances of two BPN recognizers using statistical features and raw data as input features were compared, respectively. The results indicated that the BPN using statistical features as input vectors showed the better performance than those of the other BPN using raw data as input vectors. In the experiment, they selected six statistical features from the ten statistical features (omitted remaining four features) as an input vector into BPNs (shown in Table 2).

The majority of the above features are commonly used in statistical applications. The skewness is a measure of the asymmetry of the data around the sample mean. The mean square value is the "average power" of the input vector. Autocorrelation measures the dependence of data at one instant in time with other data at another instant in time. The Cusum statistics involves the calculation of a cumulative sum and incorporates all information from sample values over time series by accumulating the sums of deviations of sample values from a target value. For more details of these statistical features, readers are referred to the literature (Hassan et al. 2003). The six statistical features extracted from raw data can represent the original data very effectively, and thus are used as one part of input features of NN recognizers in this study. Thus, the input feature vector for PSN comprises raw observations and the corresponding six statistical feature values (R&F-based input).

Identification procedure

Based on the developed pattern relationships, PSN algorithm and $|S|$ control chart, the root cause in fixture identification procedure is proposed as follows:

Step 1: Build the off-line relationships between fixture fault patterns F_1, F_2, F_3, F_4, F_5, F_6 and part variation motion patterns T_x, T_y, T_z, R_x, R_y, R_z according to section "Patterns relationships".

Step 2: Collect data sets by recording on-line time series measurements for each measurement point that need to be monitored and judge whether the process is in normal or abnormal statistical condition using $|S|$ control chart.

Step 3: Identify which ones in the part variation motion patterns T_x, T_y, T_z, R_x, R_y, R_z occur in machining process using PSN algorithm designed in section "PSN algorithm", if the process is in abnormal statistical condition.

Step 4: Identify and isolate the root cause in fixture based on the following rules (also shown in Table 3).

Rule 1: If the output result of PSN shows that T_x variation pattern appears, then the variation type F_1 is identified.

Rule 2: If the output result of PSN shows that T_x, T_y and R_z variation pattern appear,

Table 2 List of the selected and omitted statistical features (Hassan et al. 2003)

Selected features	Omitted features
Mean	Median
Standard deviation (SD)	Range
Skewness (sk)	Kurtosis
Mean-square value (ms)	Slope
Autocorrelation (ac)	
Cusum (cu)	

and $Cor(T_x, T_y) \oplus Cor(T_x, R_z) \oplus Cor(T_y, R_z)$ is true, then the variation type F_2 is identified.

Rule 3: If the output result of PSN shows that T_x, T_y and R_z variation pattern appear, and $Cor(T_x, T_y) \oplus Cor\left(T_x, R_z^{-1}\right) \oplus Cor\left(T_y, R_z^{-1}\right)$ is true, then the variation type F_3 is identified.

Rule 4: If the output result of PSN shows that T_z and R_y variation pattern appears, and $Cor(T_z, R_y^{-1})$ is true, then the variation type F_4 is identified.

Rule 5: If the output result of PSN shows that T_z and R_x variation pattern appear, and $Cor\left(T_z, R_x^{-1}\right)$ is true, then the variation type F_5 is identified.

Rule 6: If the output result of PSN shows that R_x and R_y variation pattern appear, and $Cor(R_x, R_y)$ is true, then the variation type F_6 is identified.

Rule 7: If the output result of PSN is not anyone of the results in Rule 1–6, or the output result of PSN is anyone of the results in Rule 1–6 but the corresponding result of correlation analysis is not true, then wrong identification is made.

Correct identification percentage (CIP) is used to evaluate the ability for the developed approach to identify the root cause correctly in a machining process. Higher correct identification percentage one identification approach has, better identification approach is. It can be obtained through Eq. (18),

$$CIP = \frac{CIN}{CIN + WIN} \tag{18}$$

where *WIN* represents wrong identification number, *CIN* represents correct identification number.

Case study

To validate the usefulness and effectiveness of the developed methodology, its real application in a cylinder head of engine machining process for RCI is illustrated in this section. Cylinder head is one of these most important components of an engine and has very strict dimensional specification. Its quality can directly affect the reliability and performance of the engine. However, the complexity of cylinder head machining processes make it very difficult to efficiently identify the root cause in fixture depend solely on the operator's experience. The immediate location of root cause in fixture in cylinder head machining processes can facilitate rapid analysis and corrective action by quality engineers to greatly improve dimensional quality and reduce machining cost.

Machining process

Figures 8 and 9 graphically show the operational sequence of cylinder head of engine, and fixture locating scheme of cylinder head of engine, respectively.

There are totally twenty-five measurement points to be on-line observed and monitored through the developed $|S|$

Table 3 RCI rules

Output of PSN						Result of correlation analysis	Root cause
T_x	T_y	T_z	R_x	R_y	R_z		
1	0	0	0	0	0		F_1
0	1	0	0	0	0		
⋮	⋮	⋮	⋮	⋮	⋮		
0	0	1	1	0	0	$Cor\left(T_z, R_x^{-1}\right)$	F_5
⋮	⋮	⋮	⋮	⋮	⋮		
0	0	1	0	1	0	$Cor\left(T_z, R_y^{-1}\right)$	F_4
⋮	⋮	⋮	⋮	⋮	⋮		
0	0	0	1	1	0	$Cor\left(R_x, R_y\right)$	F_6
⋮	⋮	⋮	⋮	⋮	⋮		
1	1	0	0	0	1	$Cor\left(T_x, T_y\right) \oplus Cor\left(T_x, R_z\right) \oplus Cor\left(T_y, R_z\right)$	F_2
						$Cor\left(T_x, T_y\right) \oplus Cor\left(T_x, R_z^{-1}\right) \oplus Cor\left(T_y, R_z^{-1}\right)$	F_3
⋮	⋮	⋮	⋮	⋮	⋮		

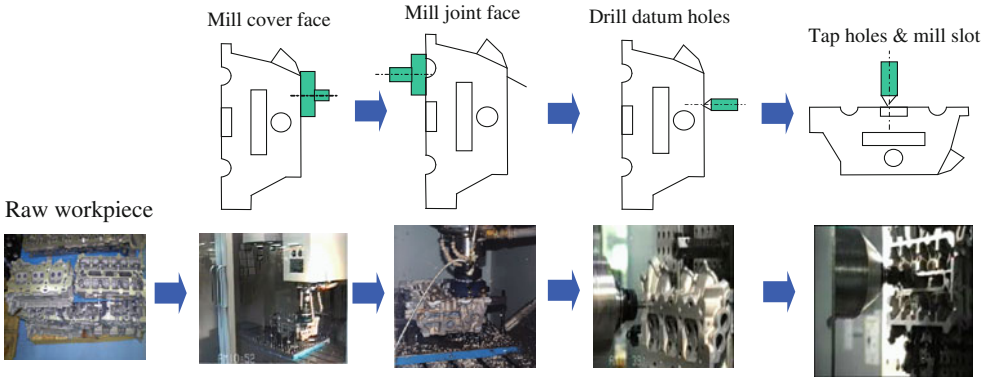
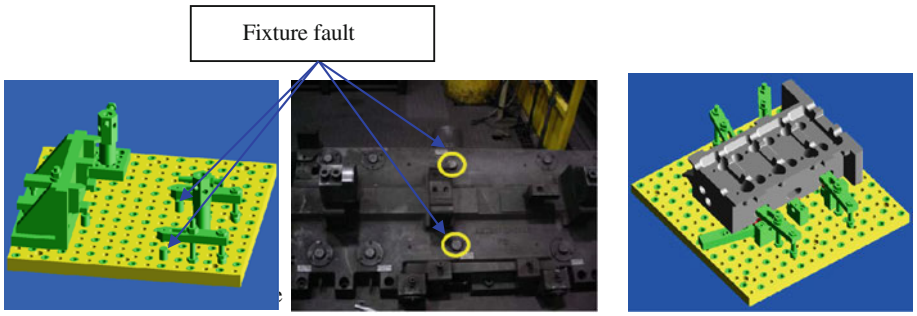


Fig. 8 Machining process of cylinder head of engine

Fig. 9 Fixture layout and locating scheme of cylinder head of engine



control chart and PSN, which are on the different product characteristics (as shown in Fig. 10):

Cover face A: $H09, H10, H11, H12, H13, H14, H15, H16$
Plane M: $X01$, $X02$, $Y01$, $Y02$, $Z01$
Slot S: $S01$, $S02$, $S03$, $S04$
Joint face B: $H01, H02, H03, H04, H05, H06, H07, H08$

The final parts are measured using coordinate measurement machine (CMM) at measurement station (shown in Fig. 11).

Structure and relative parameters

The performance of learning in PSN algorithm is influenced by its structure and relative parameters. There are no absolute rules for tuning these key factors, which are dependent on the characteristics of the real-world problems. A large number of factors discussed possibly affect the performance of PSN. These factors include different input features, number of elements in time-series window vector, sample size, number of "candidate" NNs, number of particles, and initial temperature, etc.

For practical application of the developed monitoring and identifying system, user-friendly computer programs using MATLAB toolbox® (The MathWorks Company, 2004) is developed, so potential users do not need to have a background in multivariate control and neural network. Fig. 12 shows one interface of the program.

Component NNs

(1) Input layer: There are $(n + 6) \times q$ neurons including n raw observations and six statistical features for each measurement point, where q is the number of measurement point in the assemble process. The window size 2, 4, 8, 10, 14, and 30 are considered in this study.
(2) Output layer: The output layer consists of six neurons (i.e., T_x, T_y, T_z, R_x, R_y, and R_z) representing six part variation motion pattern.
(3) Hidden layer: There is one hidden layer in the component network. The number of neurons in the hidden layer is randomly obtained from 5 to 30 when creating each "candidate" NN, which can improve the diversity of component networks and avoid complicated searching procedure to find suitable number of neurons in the hidden layer when constructing a component network.
(4) Activation function: The hyper tangent (*tansig*) and sigmoid (*logsig*) functions are used for hidden and output layer activation function, respectively.
(5) Error function: The standard error function is used to evaluate the error between the expected and actual output values of the NNs. The mean square error (MSE) is used in this study.

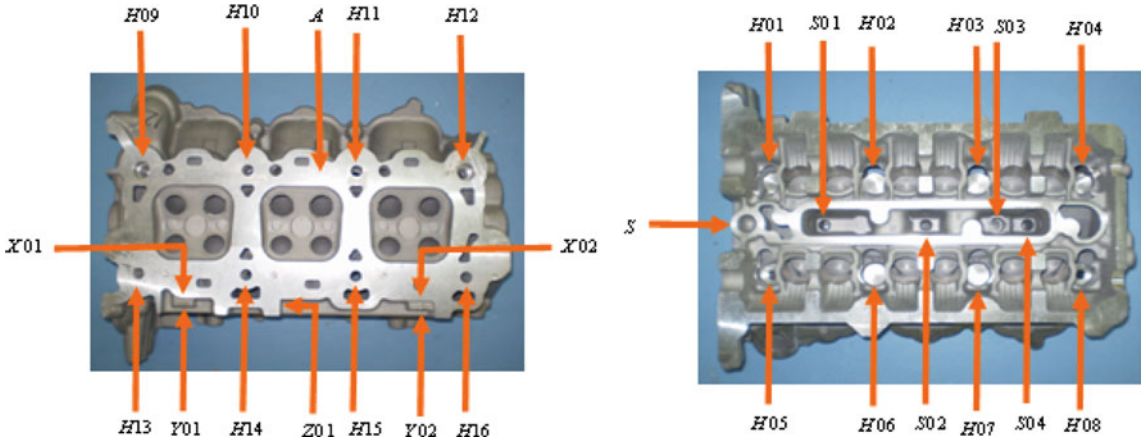


Fig. 10 Key measurement points of cylinder head of engine

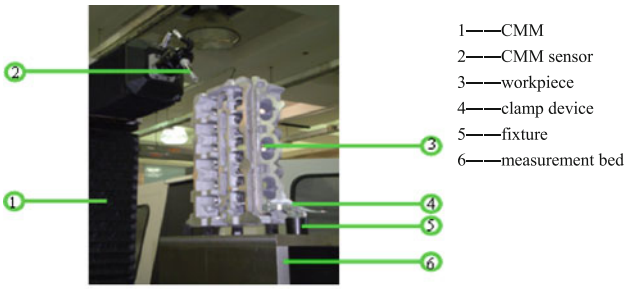


Fig. 11 Measurement station of cylinder head of engine

(6) The initial connective weights are randomly generated within a small range, i.e., [−0.2, 0.2].
(7) The learning number: the NNs stop learning when they reach a pre-determined learning number. In this study, the learning number is 1,500.
(8) The tries of "candidate" NNs: in the first step of constructing PSN, the number of "candidate" NNs, 5, 10, 15, 20, 25, 30, 35 and 40 are considered.
(9) Fitness function: fitness function plays a key role in selecting the optimal subset from "candidate" NNs. In this study, the generalization error of NN ensemble is used (i.e., fitness $= \hat{E}$).
(10) The proportion of the training data set and the test data set: in this study, 75% of examples are used as the training dataset.

PS

(1) The number of particles in *PS*: when using PS technique to select the subset from "candidate" NNs, the number of particles 5, 10, 15, 20, 25, 30, 35 and 40 are considered.
(2) Iteration times: epoch equals 150.
(3) The inertia weights: $w_{\min}$ equals 0.2 and $w_{\max}$ equals 0.8.
(4) The speed of particles: $v_{\max}$ equals 4 and $v_{\min}$ equals −4.
(5) The initial velocities: the initial velocities of the initial particles were generated at random in the range [−4, 4].
(6) The study factors: c_1 equals 2 and c_2 equals 2.
(7) The initial temperature *initemp* is 0.55.

Results and analysis

The developed RCI approach based on PSN algorithm and engineering knowledge can serve as an effective tool to monitor and identify the root cause in fixture and facilitate the formulation of better process control performance. This section reports the initial results of applying this system to a real machining process cylinder head of engine.

There are thirteen measurement points ($H09$, $H10$, $H11$, $H12$, $H13$, $H14$, $H15$, $H16$, $X01$, $X02$, $Y01$, $Y02$, and $Z01$) on cover face A (and plane M), and twelve measurement points ($H01$, $H02$, $H03$, $H04$, $H05$, $H06$, $H07$, $H08$, $S01$, $S02$, $S03$, $S04$) on joint face B (and slot S), which were measured independently. A total of 5,016 samples of measurement data for cover face A (and plane M) and joint face B (and slot S) were collected and used as training (learning) data set and test (applying) data set. The training data set and test data set consists of 3,762 samples (75%) and 1,254 samples (25%), respectively. Tables 4 and 5 present the test results of the CIP for cover face A (and plane M) and joint face B (and slot S) using R&F-based PSN and engineering rules developed in this study. Columns 1–4 in Tables 4 and 5 show output mode, actual number of times the machining process reaches the fault type, the numbers of correct identification by PSN, and CIP, respectively. F_o in the Tables 4 and 5 means that no failure occurs in machining processes. The overall total percentage of correct identification for cover face A and joint face B is 86.09 and 88.50%,

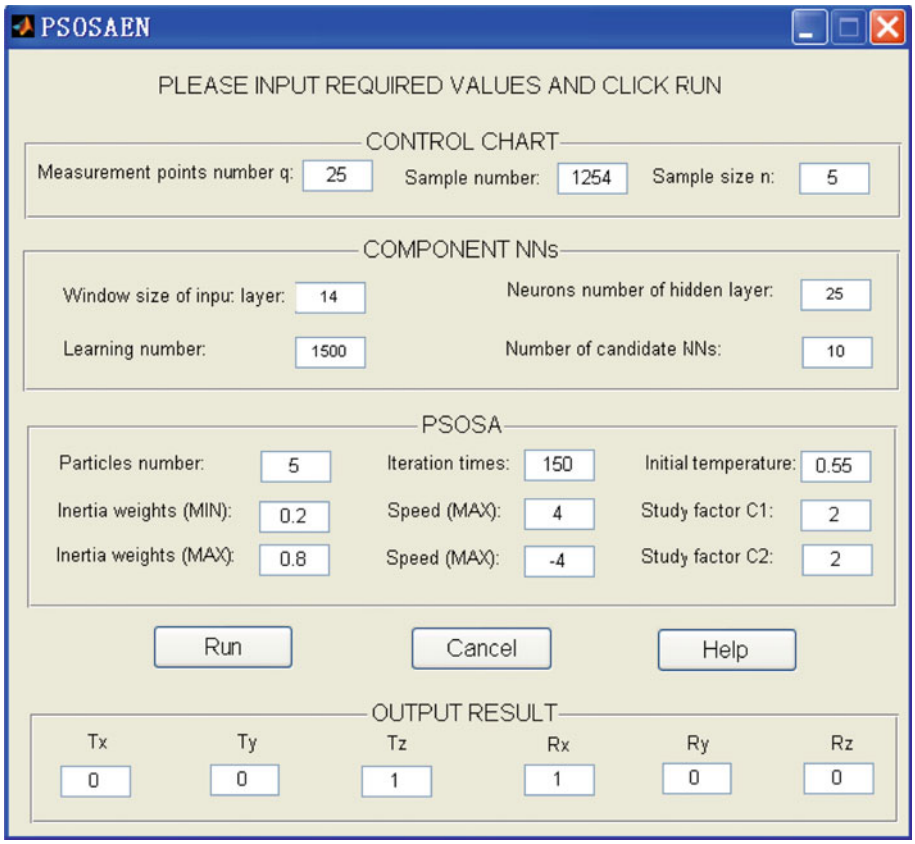


Fig. 12 Practical application of the developed monitoring and identifying system

Table 4 CIP for cover face *A* and plane *M*

Type	Actual number	The number of right classification	CIP (%)
F_o	1,148	1,127	98.17
F_1	25	24	96.00
F_2	13	11	84.62
F_3	19	16	84.21
F_4	22	17	77.27
F_5	17	14	82.35
F_6	10	8	80.00
Average			86.09

Table 5 CIP for joint face *B* (and slot *S*)

Output mode	Actual number	The numbers of right classification	CIP (%)
F_o	1,154	1,136	98.44
F_1	21	21	100
F_2	17	14	82.35
F_3	11	10	90.91
F_4	12	10	83.33
F_5	19	17	89.47
F_6	20	15	75.00
Average			88.50

indicating that the developed method exhibits a good ability to on-line identify the root cause in fixture.

From Tables 4 and 5, it can be observed that the developed identification approach has different identification ability for different fault type. For fault type F_1, the identification method has the best identification ability (CIP is 96.00 and 100%, respectively), which means that whenever the fixture fault type F_1 occurs, the approach would identify it with highest probability. Moreover, identifying fault type F_1 does not have correlation analysis. It also can be found that it is easier for the developed approach to identify fault type F_1, F_2, F_3(the average CIP is 88.28 and 85.53%, respectively) than to identify abnormal mode F_4, F_5, F_6 (the average CCP is 79.88 and 82.60, respectively). This means that it is easier to identify the translation of fixture than to identify the rotation fault of fixture. All these results demonstrate that the developed method in this study can perform effectively for RCI in a machining process.

Conclusions and discussions

For complex machining processes, solely relying on measurement data or pure artificial intelligence methods, with-

out integration with knowledge about the product/process, it is insufficient for the purpose of localizing root cause. In these cases, it becomes very necessary to integrate artificial intelligence methods as well as engineering knowledge.

One novel robust RCI approach in machining process based on hybrid learning algorithm and engineering knowledge is developed, which has the following merits: (1) this approach has excellent noise tolerance in real time, and requires no hypothesis on statistical distribution of measurements since it uses neural network ensemble; (2) the approach has explicit engineering interpretation since it integrates engineering knowledge about product, process as well as measurement information; (3) this approach can be operated without lots of intervention of operators to on-line automatically monitoring product quality based on multivariate control chart. Moreover, some user-friendly computer programs using MATLAB toolbox are developed, so potential users do not need to have a background in multivariate control and neural network. The data from the real-world cylinder head of engine machining processes are collected to validate the effectiveness of the developed approach. The analysis results indicate that the developed approach can perform effectively for identifying the root cause in fixture in the machining process.

Although this work considers mainly application in machining processes of cylinder head of engine, the proposed approach can be applied to many machining processes if these processes only need to satisfy two conditions. First one is that the parts in machining processes are rigid. The number of flexible parts variation motion patterns is possibly more than six, whereas all rigid body part variation motion can be described only through six patterns, i.e., translations along X, Y, and Z axis and rotations around X, Y, and Z axis. Since 3-2-1 fixture locating scheme is generic, the patterns map process developed in sect. "Patterns map process" can be applied to the machining processes of all rigid body parts. Another condition is that enough samples of measurements needs to be obtained in machining processes. This is not a big problem in modern machining processes. With the advancements in sensing and computational technologies, an enormous amount of process/product-related information is available in real time.

Acknowledgments The authors are thankful to the anonymous referees and the editor for their valuable comments and suggestions. The authors also thank financial support from the Natural Science Foundation of China (Grant No. 50905114), 863 High-Tech Project of China (Grant No. 2009AA043000, 2009AA043001), the Research Fund of State Key Lab of MSV of China (Grant No. MSV-2010-13), and special grade of the financial support from China Postdoctoral Science Foundation (Grant No. 201003270).

References

Bargash, M., & Santarisi, N. (2004). Pattern recognition of control charts using artificial neural networks-analyzing the effect of the training parameters. *Journal of Intelligent Manufacturing, 15*, 635–644.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Ceglarek, D., Shi, J., & Wu, S. M. (1994). A knowledge-based diagnosis approach for the launch of the auto-body assembly process. *ASME Transactions. Journal of Engineering for Industry, 116*(4), 491–499.

Cheng, C. S., & Cheng, S. S. (2001). A neural network-based procedure for the monitoring of exponential mean. *Computers & Industrial Engineering, 40*(4), 309–321.

Das, N., & Prakash, V. (2008). Interpreting the out-of-control signal in multivariate control chart—a comparative study. *International Journal of Advanced Manufacturing Technology, 37*, 966–979.

Du, S., Xi, L., Ni, J., Pan, E., & Liu, R. (2008). Product lifecycle-oriented quality and productivity improvement based on stream of variation methodology. *Computers in Industry, 59*(2–3), 180–192.

Eberhart, C. R., & Kennedy. J. (1995). Particle swarm optimization. In *Proceedings IEEE international conference on neural networks* (pp. 1942–1948). Piscataway, NJ.

Ertuğrul, İ., & Aytaç, E. (2009). Construction of quality control charts by using probability and fuzzy approaches and an application in a textile company. *Journal of Intelligent Manufacturing, 20*, 139–149.

Guh, R. S. (2007). On-line identification and quantification of mean shifts in bivariate processes using a neural network-based approach. *Quality and Reliability Engineering International, 23*, 367–385.

Gutta, S., & Wechsler, H. (1996). Face recognition using hybrid classifier systems. In: *Proceedings of the ICNN-96* (pp. 1017–1022). Washington, DC: IEEE Computer Society Press, Los Alamitos, CA.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*, 993–1001.

Hassan, A., Shariff, N. B. M., Shaharoun, A. M., & Jamaluddin, H. (2003). Improved SPC chart pattern recognition using statistical features. *International Journal of Production Research, 41*(7), 1587–1603.

Huang, Y., McMurran, R., Dhadyalla, G., & Jones, R. P. (2008). Probability based vehicle fault diagnosis: Bayesian network method. *Journal of Intelligent Manufacturing, 19*(3), 301–311.

Jiang, P. Y., Liu, D. Y., & Zeng, Z. J. (2009). Recognizing control chart patterns with neural network and numerical fitting. *Journal of Intelligent Manufacturing, 20*, 625–635.

Lian, J., Lai, X., Lin, Z., & Yao, F. S. (2002). Application of data mining and process knowledge discovery in sheet metal assembly dimensional variation diagnosis. *Journal of Materials Processing Technology, 129*, 315–320.

Liu, G., & Hu, S. J. (2005). Assembly fixture diagnosis using designated component analysis. *Journal of Manufacturing Science and Engineering, 127*, 358–368.

Lu, W. Z., Fan, H. Y., & Lo, S. M. (2003). Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong. *Neurocomputing, 51*, 387–400.

Maqsood, I., Khan, M. R., & Abraham, A. (2004). An ensemble of neural networks for weather forecasting. *Neural Computing and Application, 13*, 112–122.

McCulloch, C., & Searle, S. R. (2001). *Generalized, linear, and mixed models*. New York, NY: Wiley.

Montgomery, D. C. (2005). *Introduction to statistical quality control* (5th ed.). New York, NY: Wiley.

Pandey, V., Tiwari, M. K., & Kumar, S. (2006). An interactive approach to solve the operation sequencing problem using simulated annealing. *International Journal of Advanced Manufacturing Technology, 29*, 1212–1231.

Perrone, M. P, & Cooper, L. (1993). When networks disagree: Ensemble method for neural networks. In R. J. Mammone (Ed.), *Artificial neural networks for speed and vision* (pp. 126–142). New York: Chapman & Hill.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*, 197–227.

Shanker, M., & Hu, M. (1996). Cutoff values for two-group classification using neural networks. *Industrial Mathematics, 46*(1), 33–45.

Shi, J., & Zhou, S. (2009). Quality control and improvement for multistage systems: A survey. *IIE Transactions, 41*, 9,744–9,753.

Smith, A. E. (1994). X-Bar and R control chart interpretation using neural computing. *International Journal of Production Research, 32*, 309–320.

Wang, T. Y., & Chen, L. H. (2002). Mean shifts detection and classification in multivariate process: A neural-fuzzy approach. *Journal of Intelligent Manufacturing, 13*(3), 211–221.

Yen, C. H., Wong, D. S. H., & Jang, S. S. (2004). Solution of trim-loss problem by an integrated simulated annealing and ordinal optimization approach. *Journal of Intelligent Manufacturing, 15*, 701–709.

Zhou, Z. H., Wu, J. X., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137*, 239–263.

Zhou, S., Chen, Y., & Shi, J. (2004). Root cause estimation and statistical testing for quality improvement of multistage manufacturing processes. *IEEE Transactions on Automation Science and Engineering, 1*(1), 73–83.