China Steel Technical Report, No. 30, pp.66－71, (2017)

The Development of On-Line Balancing System for Rotating Machinery

CHIOU-YI HOR*, CHUNG-YUNG WU*, HUANG-CHIN HUNG and KUANG-WEI SHIH****

**Green Energy & System Integration Research & Development Department*
***Rolling Mill Department I - Plate, Bar & Rod*
China Steel Corporation

In the steel industry, excessive vibration in the rotating machinery may produce uncomfortable noise and even cause catastrophic failure. Among the reasons that lead to this above problem, the major one is rotor imbalance. The imbalance denotes the non-uniform distribution of mass in the rotor, which may be due to corrosion or adhesion of extraneous particles. In this report, we introduce an on-line balancing system for the wire laying head. The system was developed by China Steel (CSC) and is aimed at the single-plane rotor balancing process. The balancing system is composed of an accelerometer, a probe sensor and software. In order to determine whether the accelerometer can be adopted for the subsequent balancing, we also developed a sensor checking method that is performed before the balancing process. With the assistance of the system, the mechanical personnel can proceed with the balancing process efficiently and prevent the production machines from unexpected failure.

Keywords: Vibration, Single plane balancing, Feature extraction, Decision tree

1. INTRODUCTION

The ideal rotating machine rotor should be symmetrical in geometry and uniform in mass. Consequently, it is expected that the rotating element can rotate smoothly without significant vibration. However, due to some operational requirements, like slots or keys, or other reasons, such as rustiness or adherence of material, the rotating element would become unbalanced gradually and exhibit excessive vibration. The situation may lead to machine fatigue, wear or internal friction, which eventually cause machine failures. The typical practice to perform balancing is by the graphical method, as illustrated in Fig.1. This heavily requires well-trained personnel with experience and skill. In order to avoid this situation, we developed an online balancing system in the wire laying head, so that the balance can be restored to an acceptable level without requirement of professional skills. The system is aimed at the single-plane rotor balancing process[1] and is consisted of an accelerometer (vibration sensor), a probe sensor and software.

Since an accelerometer is a key component to measure vibration, to determine whether it can function appropriately is important. The conventional way to check an accelerometer is by the BOV method[2]. Another proprietary method adopted at China Steel (CSC) is to send a excitation signal to the sensor and observe visually whether its response is conformable to specific patterns[3]. However, the process not only relies on experienced personnel but also is hardware dependent. This means that the response is different when the accelerometer and signal acquisition hardware are changed. In order to overcome the difficulty, we propose a machine-learning based method which can automatically build a prediction model according to the presented signals. The process starts by collecting signals from accelerometers of known fault states and then extracts features from these signals. The extracted fea-

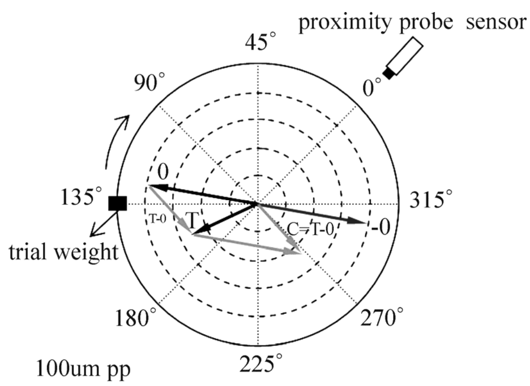


Fig.1. A graphical way for single plane balancing.

tures are then fed into a decision-tree learning machine to build a prediction model[4]. To determine whether the accelerometer of an unknown fault state is healthy or not, the features associated with its response signal are brought into the prediction model. In addition to distinguishing whether the accelerometer is healthy or not, the model also reports which kind of fault the accelerometer is, including short circuitry, open circuitry, disconnect and inverse connection. This facilitates maintenance personnel for quick trouble shooting. The overall procedure is shown in Fig.2.

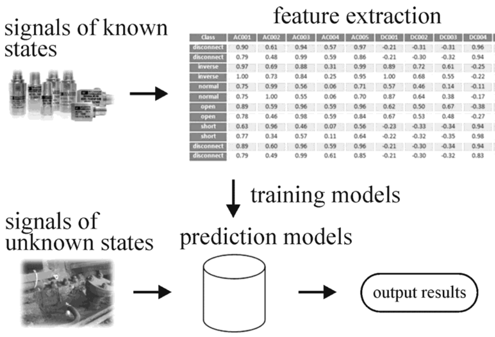


Fig.2. The sensor checking procedure.

2. THE METHOD OF SINGLE PLANE BALANCING

2.1 Notations for rotor balancing

In order to perform a balancing process, a proximity probe sensor and a accelerometer shall be provided. They are responsible for measuring the phase and vibration respectively. For the rotor with only one mark, the probe sensor generates a trigger signal per revolution and the rotational frequency, as known as 1X speed, can be calculated by counting the interval between two pulses. Once the rotation frequency is obtained, the signal associated with this frequency is extracted, the phase and amplitude for the 1X speed can be computed accordingly.

2.2 Two-run single plane balancing

The influence coefficient H describes how the balance changes in terms of vibration when one additional weight m_1 is added to the rotor. Assume the original imbalance (1X vibration) is O in the complex form. When one trial weight m_1 is added to the rotor, the imbalance becomes T. The influence factor is given by

$$H = \frac{C}{m_1} = \frac{T-O}{m_1} \qquad (1)$$

where C denotes the change of imbalance. Since both the weight and vibrations are complex, the influence coefficient H is also a complex number.

In order to balance the rotor, another weight, m_2, should be added, so that the resulting amount of balance change is -O. This cancels out the original imbalance. It turns out that the required weight for m_2 is as follows.

$$m_2 = \frac{-O}{H} = \frac{-O \times m_1}{T-O} \qquad (2)$$

In case m_1 is not removed, Equation (2) should be

$$m_2 = \frac{-T}{H} = \frac{-T \times m_1}{T-O} \qquad (3)$$

To assist maintenance personnel in balancing, we also developed software. The software is realized in a Windows system. It retrieves vibration and trigger signals from the accelerometer and probe sensor, which are used to calculate imbalance and rotation speed respectively. This information is then employed for suggesting how heavy and where to put the correction weight. In addition, the software also provides operation procedures and necessary plots. The self explanation user interface makes the balancing process more straightforward and easier for operation, which is illustrated in Fig.3.

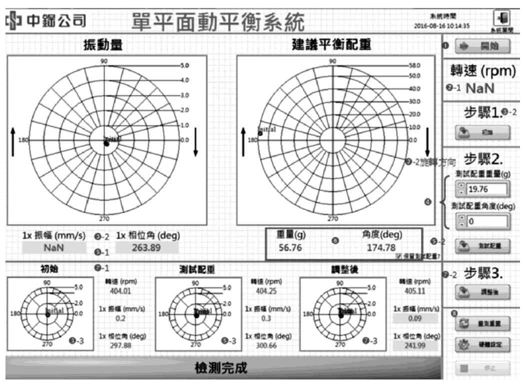
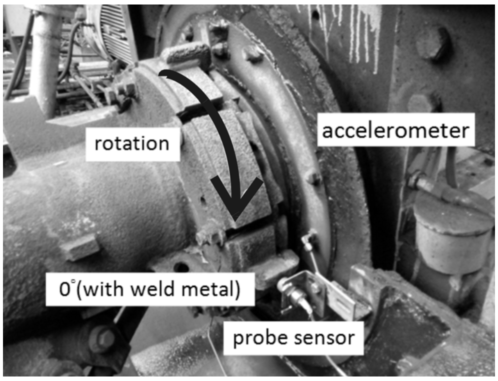
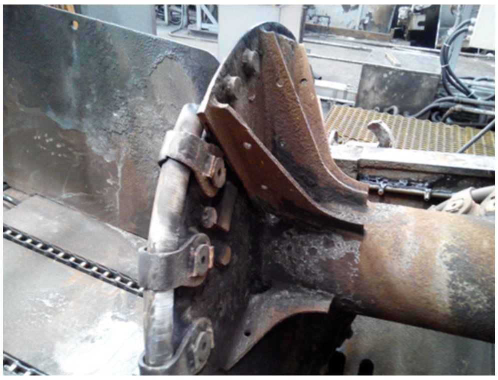


Fig.3. The user interface for balancing software.

In order to verify the functionality of the balancing system, we selected the wire laying heads, which often encounter imbalance, for this experiment. The wire laying head is located at the end of the high speed wire mill and is used to lay the hot-rolled wire into loops on a cooling conveyor. In order to ensure high productivity, the device must operate at a high speed. The typical operation speed ranges from 50 to 150 meters per second and requires high stability. Figure 4(a) shows the arrangement to balance the wire laying head. The laying head rotates at about 30 Hz in a clockwise direction. A vibration accelerometer and proximity probe sensor is installed near the bearing. The trial and correction weights are mounted at the end of the laying head, as shown in Fig.4(b). After balancing, the amount of vibration is reduced from the original 15.6 mm / s pk-pk to 0.9 mm/s. This indicates that the balancing system functions as expected.



(a) The allocation of probe sensor and accelerometer



(b) The location to place trial and correction weight

Fig.4. The arrangement to balance a wire laying head.

2.3 One-run single plane balancing

For the method presented in the above section, users have to add a trial weight to obtain influence coefficients first, and then a final correction weight is thus calculated. The method is called two-run balancing because the rotor is undertaken two stop-and-start procedures. The disadvantage of two-run balancing is that multiple overhauls are time-consuming.

In view of this drawback, we have developed the so-called one-run method. The principle is to adopt influence coefficients that are previously made in the conventional two-run practice to suggest a correction weight in an one-shot manner. Figure 5 is the software that integrates two-run and one-run balancing. Once a two-run balancing process is completed, the corresponding influence coefficient as well as date information are recorded. When a balancing process should be carried out, the software will suggest a trial weight by means of influence coefficient previously made and imbalance vibration in the meanwhile. The calculation method follows Equation (2). After adding the correction weight, the user observes whether the reduction in vibration meets the operational requirement. If so, the user can stop the experiment and operations can proceed. Otherwise, the conventional two-run balancing process is performed and a new influence coefficient is re-computed. Take Fig.5 for example, the original vibration was 5.87 mm/s. After the first trial, the vibration was reduced to 2.11 mm/s. If the balancing process is proceeded, the final vibration is further reduced to 0.37 mm/s. The method can be improved to cases of multiple influence coefficients. For instance, an influence coefficient can be built for each imbalance level. The calculated correction weight would more precisely accommodate to the current imbalance and thus a substantial reduction in vibration is more easy to obtain.

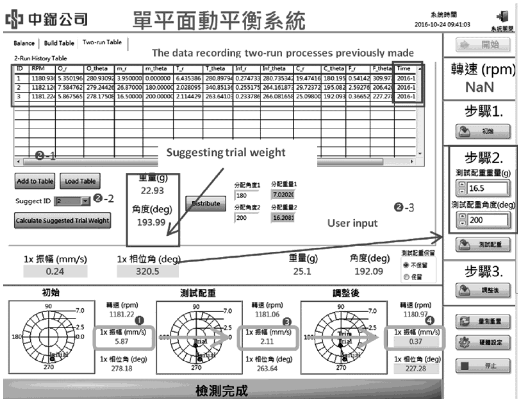


Fig.5. The software that integrates two-run and one-run balancing.

3. THE SENSOR CHECKING METHOD

The sensor checking process is divided into the following steps: data collection, model training and on-line prediction. For the first stage, it manages to collect signals from accelerometers of known fault states and extracts relevant features. Then, the collected features as well as associated fault states are combined together into a training dataset (as shown in Fig.2). The dataset is used to build a decision tree model. The model can be deployed on-line to determine the health state of an accelerometer and what kind of fault the accelerometer has, including short circuitry, open circuitry, disconnection and inverse connection. The typical AD and DC signals, which are obtained by sending an accelerometer a 4-mA current and acquiring its response in AD or DC mode, are shown in Fig.6 .

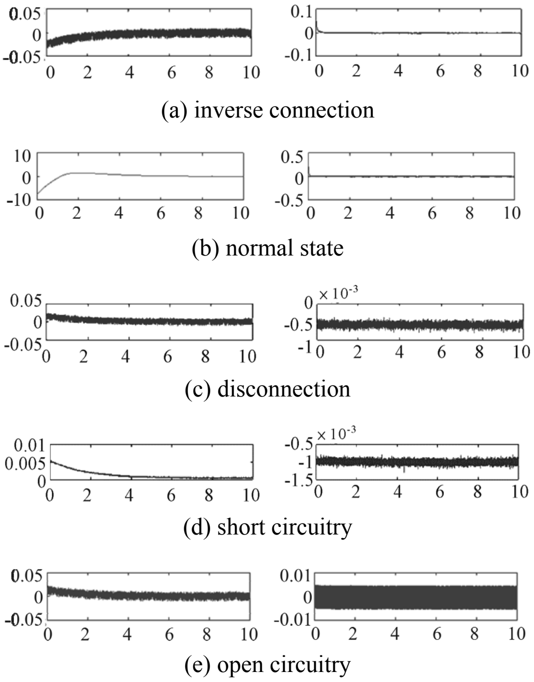


Fig.6. Typical AC/DC signals for accelerometers of various states.

3.1 Data collection

In this report, we adopted two categories of features to encode signals. The first one was histogram-based feature, and the other one time-domain feature.

A histogram, as shown in Fig.7 (a), records the number of occurrences, also known as frequency, for each kind of event and arranges them in a bar chart. Since the event, usually located in the horizontal axis, is shown in a discrete manner, continuous variables should be converted into discrete events or bins. The way to partition continuous variables into discrete events is called binning.

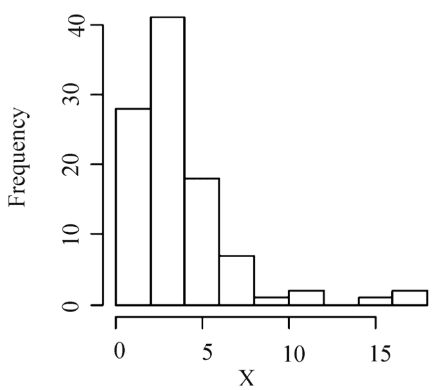


Fig.7. Histogram examples.

In this report, the original excited signals, including AC and DC ones, are first normalized as follows.

$$\mu = \frac{1}{N}\Sigma_i \ x_i, \sigma = \frac{1}{N}\Sigma_i (x_i - \mu)^2 \quad \text{.................................(4)}$$

$$x_i = (x_i - \mu)/\sigma, i = 1 \sim N$$

In the above equation, x_i, where i =1~N, denotes the original excited signals, and μ and σ are the mean and standard deviation, respectively. Each normalized signal is then subjected to an outlier removal and histogram processing. The outlier removal threshold is set to be $\mu \pm 3\sigma$ and the bin number is fixed to 50.

We first select five template signals, which are associated with normal, short circuitry, open circuitry, disconnection and inverse connection, and calculate their normalized histograms. For a given signal, a correlation coefficient is made between its normalized histogram and each normalized one of the above template signals. Since there are totally five AD and DC templates, ten histogram-based features are extracted.

For the time-domain features, a time series is partitioned into several regions. In the example shown in Fig.8, the number of regions is seven. Data in each region is averaged and a mean value is obtained. Besides, differences of mean values between adjacent regions are calculated and then compared with a threshold, such as 10% standard deviation of the entire signal. If a difference is greater than one plus or minus threshold, an +1 or -1 is designated. Otherwise, a 0 is output. For a AC signal of seven regions, there are 7 + 6 = 13 features. Combining the DC signal, a total of 26 features are extracted. All the above features are named in a prefix AC or DC plus serial numbers, like AC001, AC002, AC003 etc.

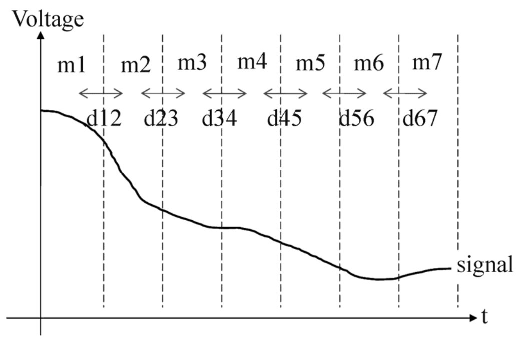


Fig.8. Time-domain features.

3.2 Model training and validation

Decision trees (DTs) belong to one of the non-parametric supervised learning methods and are widely used for both classification and regression tasks. The learning goal is to construct a model that predicts the value of a target variable by deriving simple decision rules inferred from the training data. A decision tree is composed of nodes, each of which is represented by a "IF…Then...Else…" form. The main advantage for this type of model is ease of interpretation and can be visualized, as illustrated in Fig.9.

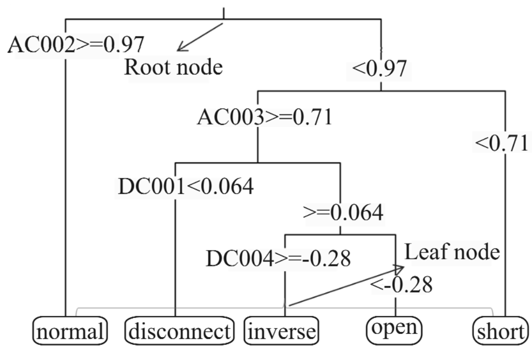


Fig.9. A decision tree of sensor checking model.

After a decision tree is trained, it can be used for on-line prediction. That is, the fault state of any incoming signal can be determined by the tree. The prediction process starts by extracting features from the signal and searching the tree from its root node to leaf nodes. The search process continues until a consistent path, from the root node to leaf node, is found. A path is consistent if all conditions along the path are met. Take Fig.9 for example, the first condition to check is whether the AC002 feature is greater than 0.97. If the condition is met, the search path will follow the "yes" branch. Otherwise, the "no" branch will be adopted and the next condition "AC003>=0.71" is checked again. Each leaf node represents the final decision that the tree mode reports.

In this report, we adopted a cross validation to examine the performance of the built-in prediction model. In the case of a 10-fold cross-validation, the original data is divided into ten equal parts, among which nine of them are used for training and the rest of the data is used for validation. This process is repeated ten times until all validation data has been tested and the average performance is calculated. In Table 1, for example, a total of 820 pieces of data are used for the experiments. There are a total of five categories of data and the number of each piece of data for each category ranges between 163 to 165. The results show that except for five among the 165 open pieces of data are misclassified as inverse, the remaining ones are correctly classified. The average error rate is 1.22%. The experiments are also made on different brands of signal acquisition hardware, as illustrated in Table 2, and similar results are obtained. This means that the proposed

Table 1 A confusion table for the Brand A signal acquisition hardware

10-fold CV error rate: 1.22%						
	disconnection	inverse	normal	open	short	Sum
disconnection	163	0	0	0	0	163
inverse	0	160	0	5	0	165
normal	0	0	164	0	0	164
open	0	**5**	0	160	0	165
short	0	0	0	0	163	163
Sum	163	165	164	165	163	820

Table 2 A confusion table for the Brand B signal acquisition hardware

10-fold CV error rate: 1.34%						
	disconnection	inverse	normal	open	short	Sum
disconnection	163	0	0	0	0	163
inverse	0	160	0	5	0	165
normal	0	0	164	0	0	164
open	0	**4**	0	161	0	165
short	0	0	0	0	163	163
Sum	163	164	164	166	163	820

features are sufficient and decision trees are capable of extracting the knowledge underlying the data.

4. CONCLUSION

In this report, we introduce an on-line balancing system for a rotary machine. The system was aimed at the single-plane rotor balancing process and has been validated on the wire laying head. We also developed software, which can help maintenance personnel to perform balancing tasks efficiently. Beside the balancing system, we proposed an accelerometer checking methodology, which can quickly help maintenance personnel identify whether accelerometers can function appropriately. To make the checking procedure less hardware dependent, we propose to extract relevant features that are incorporated with decision trees to make the process more universal. Experimental results reveal that our methods can be applied to different hardware and the performance is sufficient enough for practical applications. With the assistance of the system, the mechanical personnel can proceed with the balancing process efficiently and prevent the production machines from unexpected failure.

REFERENCES

1. Victor Wowk: "Machinery Vibration: Balancing", McGraw-Hill Education, 1998, pp. 91-116.
2. Richard M. Barrett: "Troubleshooting accelerometer installations", Wilcoxon Research, Maryland, 2004.
3. Chih-Chung Wang, Yi-An Wu and Tai-Yuan Chiu: "The method to detect malfunction of accelerometer", Taiwan Patent I335425, 2011.
4. Johannes Ledolter: "Data Mining and Business Analytics with R", John Wiley & Sons, 2013, pp. 161-184.
5. Donald E Bently: "Fundamentals of Rotating Machinery Diagnostics", ASME Press, 2003, pp. 21-32.