HDPS-BPSO based predictive maintenance scheduling for backlash error compensation in a machining center

Zhe Li[1], Yi Wang[2], Kesheng Wang[3], Jingyue Li[1]

Abstract. This paper presents a novel HDPS-BPSO maintenance scheduling strategy for backlash error compensation in a machining center through binary particle swarm optimization (BPSO) and data-driven regression methods. During the experiment, a hierarchical diagnosis and prognosis system (HDPS) was leveraged to predict the potential backlash error first. Then BPSO is applied to provide optimized maintenance strategies through capturing the trade-off between several factors such as maintenance cost, machining accuracy, and defective percentage. The target of proposed predictive maintenance strategy is to minimize the cost of potential failures and relevant maintenance performances. The numerical result in this case demonstrates the benefit of implementing predictive maintenance compared with preventive one.

Keywords: Maintenance scheduling, Binary particle swarm optimization, Machining center, Backlash error compensation

1 Introduction

Backlash errors that occur in machining centers may cause a series of changes in the geometry of the components and lead to breakdown with serious safety, environment, and economic impact. In our previous work [1],a hierarchical diagnosis and prognosis system (HDPS) was proposed and proved its advantages in backlash error prediction. This paper focuses on the implementation of HDPS-based predictive

[1] Zhe Li (✉), Jingyue Li

Department of Computer Science, Norwegian University of Science and Technology, Norway

e-mail: zhe.li@ntnu.no; jingyue.li@ntnu.no

[2] Yi Wang

School of Business, University of Plymouth, UK

e-mail: yi.wang@polymouth.ac.uk

[3] Kesheng Wang

Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, Norway

e-mail: kesheng.wang@ntnu.no

maintenance. In order to capture the trade-off between several factors such as maintenance cost, machining accuracy, and defective percentage, a novel HDPS-BPSO maintenance implementation strategy driven by binary particle swarm optimization (BPSO) is proposed in this paper. After discovering fault information of the equipment from HDPS, the next step is to implement predictive maintenance according to the prediction of potential failures or degradation, which is usually a nondeterministic polynomial time problem. Here, the implementation strategy can be regarded as a maintenance scheduling optimization problem. Inspired by particle swarm optimization's (PSO) advantages [2], a novel HDPS-BPSO maintenance implementation strategy is proposed to find the optimum solution for predictive maintenance implementation. Since PSO is easier to implement with a few parameters to tune and is computationally inexpensive [3], it may be a perfect solution in this case.

The remaining part of this paper is organized as follows. Section 2 introduces the basis of PSO. Section 3 details the principle of BPSO. Section 4 proposes a novel HDPS-BPSO based maintenance scheduling along with the numerical results. Conclusion and future work are summarized in the last section of this paper.

2 Basis of PSO

As a kind of computational method, the target of particle swarm optimization is to solve the optimization problems, by iteratively trying to improve candidate solutions with communication within the swarm and randomly search, which is inspired from movement of organisms in a bird flock [4]. The current solution $\overrightarrow{x_i}$ is considered as coordinates which could describe a position in space. If the new solution could be better than others, the new one would be stored as the personal best solution $\overrightarrow{p_i}$. And the best result so far will be saved as the global best solution $\overrightarrow{p_g}$. The target is to keep finding better solutions and updating $\overrightarrow{p_i}$ and $\overrightarrow{p_g}$.

In optimization process, candidate solutions are produced in the form of particles. These particles move around in the solution space of the problem according to some simple mathematical formulae over the particle's positon and velocity. The movement of each particle is influenced by the best known personal position and also the best known global position in the searching space, which is updated as the best solution found so far by the swarm. This update makes the swarm move toward the best solutions [5]. During optimization process, each particle would remember own previous best ever value together with their neighbourhood best. The general process of implementing PSO were shown as follows:

1.Parameters Initialization, such as iterations, population, velocities v_i, and positions x_i.

2. Loop

3. Check the target optimization fitness according to each particle's position $\overrightarrow{x_i}$

4. Update the optimized personal best solution $\overrightarrow{p_i}$until now.

5. Update the optimized global best solution $\overrightarrow{p_g}$ so far.

6. Change the velocity of each particle at t^{th} iteration to $(t+1)^{th}$ iteration according to:

$$\vec{v_i}(t+1) = \omega \cdot \vec{v_i}(t) + c_1 \cdot r_1\left(\vec{p_i} - \vec{x_i}(t)\right) + c_2 \cdot r_2\left(\vec{p_g} - \vec{x_i}(t)\right) \tag{1}$$

In which, w is the inertia weight, $c_1$and c_2 are coefficients for acceleration and $r_1$and r_2 are distribution in [0, 1].

7. Update the position of each particle according to the following Equation (2):

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t+1) \tag{2}$$

8. If the optimised solution could meet certain criterion, then loop will end. The criterion is usually set to be the maximum iterations, the number of iterations in which the objective has not been improved, or the fitness is sufficiently good.

The role of weight w in Equation 1 is considered critical for the convergence behaviour of PSO. Some researchers have reported that it is usually better to set the inertia to a large value [6], which may promote global exploration, and reduce it to get more refined positions according to their experiments [7,8]. The parameters $c_1$and c_2 in Equation 1 are not critical for the convergence of PSO. Some experiment results indicate that $c_1 = c_2$ =1.49 might provide even better results. According to Equation 1, it is better for local exploration when $c_1 > c_2$, while global exploration would do better when $c_1 < c_2$ [9].

3 BPSO

PSO was originally developed for continuous valued spaces, however, some practical problems require discrete solutions, which shall be defined in discrete spaces. Kennedy and Eberhart proposed binary particle swarm optimization (BPSO) to solve this issue in 1997 [10]. In their model, each particle could present the solution through a binary value.

In BPSO, the personal best and global best solution are also updated in a continuous version. The difference lies on the improvement of velocity during optimization, which is defined as changes of probabilities, which could make the change in one state or the other. Therefore, velocity would be limited into [0,1] through defining a logistic transformation S, usually a sigmoid function as Equation (3).

$$S\left(v_{ij}(t)\right) = \frac{1}{1+e^{-v_{ij}(t)}} \tag{3}$$

Where $v_{ij}(t)$ means the j^{th} component of vector $\vec{v_i}(t)$. Then the new solution of the particle could be updated through Equation (4).

$$if\ rand_{ij} < S\left(v_{ij}(t)\right)\ then\ x_{ij}(t+1) = 1 \quad (4)$$

$$otherwise\ x_{ij}(t+1) = 0$$

Where $rand_{ij}$ is a random number selected from a uniform distribution in [0, 1], $x_{ij}(t+1)$ represents the j^{th} component of vector $\vec{x_i}(t+1)$.

However, raising the positive direction in the BPSO will cause larger probability for the particle solution, while raise in the negative direction results in probability of zero. When the optimization process has nearly reached to the optimum solution, the probability of changing the position of the particle must be near to zero, while at this point using sigmoid function, the position will change by taking the value of 1 or 0 with the probability of 0.5, which would cause the algorithm not to converge well. To avoid this situation, hyperbolic tangent (Tanh) function, as shown in Equation (5), is leveraged as the transformation function

$$S\left(v_{ij}(t)\right) = \left|tanh\left(\alpha v_{ij}(t)\right)\right| = \frac{e^{\alpha v_{ij}(t)} - e^{-\alpha v_{ij}(t)}}{e^{\alpha v_{ij}(t)} + e^{-\alpha v_{ij}(t)}} \quad (5)$$

Where α is the weight vector of the transportation.

4 HDPS-BPSO based maintenance scheduling

As introduced above, backlash error that will occur in the equipment at all positions and directions could be predicted through proposed HDPS. During the scheduling, our target is to minimize the total cost raised by backlash error, including the maintenance cost, machining accuracy, and defective percentage in the latest 25 weeks.

The main cost function in this case study includes degradation cost C_G, maintenance cost C_M, and inspection cost C_I. Here, the assumptions and definitions in the mathematical model are given:

Assumption 1: In practical industrial applications, the relationship between production and maintenance is usually considered as a conflict in management decision. Here, we assume the maintenance scheduling compromises the production scheduling, which means the workload of the equipment will not change with maintenance decisions.

Assumption 2: The degradation in specific direction and position completely follows the mapping provided by HDPS.

Assumption 3: Once a maintenance has been performed, the degradations in all directions and positions are supposed to return back to the initial values (Week 1). Subsequent degradations keep following HDPS according to the distance from the last maintenance performance.

Assumption 4: If maintenance has been scheduled, it is supposed to be performed at the beginning of that week.

Assumption 5: Holidays have been excluded from the mathematical model.
C_G: Degradation cost.
C_M: Maintenance cost.
C_I: Inspection cost.
W: Number of weeks to be scheduled.
A: Number of axes inspected.
P: Number of axial positions inspected.
Pr: Production profit in unit time.
D_P: Maximum permissible degradation.
D_N: Criterion of normal product.
M: Cost of maintenance performance.
$Load_i$: Working load in week i.
H: Maximum working hours per week.
h: Time of single maintenance performance.
D_{ijk}: Degradation in week i along j axis at position k predicated from HDPS.
D'_{ijk}: Degradation in week i along j axis at position k after maintenance scheduling.
α: Weighting factor for degradation cost.
β: Weighting factor for maintenance cost.
d_i: Distance from the last maintenance in week i.
x_i: Decision variable.
Decision variable x_i during optimization is defined as:

$$if\ maintenance\ performanced\ in\ week\ i\quad then\ \ x_i = 1 \qquad otherwise\ x_i = 0 \tag{6}$$

The degradation cost here is caused by the geometrical error from backlash directly. It could be estimated as following:

$$C_G = \sum_{i\in W}\sum_{j\in D}\sum_{k\in P} Load_i * H * \varphi(D'_{ijk}) \tag{7}$$

$$D'_{ijk} = D_{d_i jk}$$
$$if\quad x_i = 1 \qquad then\ d_i = 1 \qquad otherwise\ d_i = d_{i-1} + 1$$

Where $\varphi(\ )$ denote the production cost caused by degradation. It can be calculated according to Equation 8:

$$\varphi(D'_{ijk}) = \begin{cases} 0 & if\ \ D'_{ijk} \le D_N \\ \frac{D'_{ijk} - D_N}{D_P} * Pr & if\ \ D_N < D'_{ijk} \le D_P \\ Pr & if\ \ D'_{ijk} > D_P \end{cases} \tag{8}$$

Here, we consider when the degradation is between the normal and maximum permissible degradation, the manufacturing profit decreases with a linear manner with degradation. The maintenance cost here is evaluated according to the number of maintenance performance.

$$C_M = M * \sum_{i \in W} x_i \quad (9)$$

Then, the total cost C_{tot} can be obtained as:

$$C_{tot} = \alpha * C_G + \beta * C_M + C_I \quad (10)$$

With constraint $\forall x_i \in W\colon\ x_i * h + Load_i * H \leq H$

Because the equipment is inspected in a continuous manner in this model, the value of C_I is fixed. Since some issues such as incidental damage or cost caused by maintenance, and the loss in reputation of producing imperfect products. α and β can be leveraged to weight the effect of degradation and maintenance here, respectively.

The parameters of HDPS-BPSO are set according to the case study as: number of population size is 100, maximum iteration is 500, weighting coefficients α and β are both set as 1, W is 25 weeks, A is 2 axes, P is 25 positions, Pr is 2,000 Norwegian Krone (NOK) /hrs, M is 15,000 NOK, D_P is 16 µm, D_N is 12.5 µm, H is 45 hours, h is 2 hours. During the test, we leveraged hyperbolic tangent function as logistic transformation for optimization. The numerical result of HDPS-BPSO is as following. The convergence starts around 200^{th} iteration. According the numerical result, the best predictive maintenance solution in this case is to perform maintenance in week 9 and week 18, in which the total cost including the loss from degradation and maintenance cost is 33,303 NOK. According to the previous preventive maintenance strategy, the maintenance is supposed to be performed every 6 weeks. The cost is also calculated based on the preventive maintenance strategy. When maintenance executed in week 7, 13 and 19. The total cost is 47,881 NOK. Therefore, through predictive maintenance, the maintenance cost of single machine center can be reduced by 14,578 NOK in this case.

5 Conclusion and future work

In this paper, a novel maintenance implementation strategy HDPS-BPSO is proposed to illustrate the implementation of predictive maintenance in practical application. A maintenance model for backlash error compensation in machining centers is also established. With the help of BPSO, we can find the optimized maintenance strategy for the machining center to achieve zero-defect production and leverage the remaining useful life as long as possible. The numerical result shows the benefit of implementing the strategy of predictive maintenance compared with that of preventive maintenance. In this research, we assume the maintenance scheduling compromises the production scheduling, which removes the issues about joint production from this research since maintenance scheduling for a single machine will always compromise its planned work. This assumption could separate machining centers from each other in maintenance scheduling to fit this case. Future work may focus on the application of predictive maintenance in group maintenance scheduling.

Acknowledgement

This work is supported by the CIRCIT (Circular Economy Integration in the Nordic Industry for Enhanced Sustainability and Competitiveness) project, which is financed by Nordic Green Growth Research and Innovation Programme.

References

1. Li Z, Wang Y, Wang K (2017) A data-driven method based on deep belief networks for backlash error prediction in machining centers. Journal of Intelligent Manufacturing. doi:10.1007/s10845-017-1380-9

2. Rini DP, Shamsuddin SM, Yuhaniz SS (2011) Particle swarm optimization: technique, system and challenges. International Journal of Computer Applications 14 (1):19-26

3. Yue-Jiao G, Meie S, Jun Z, Kaynak O, Wei-Neng C, Zhi-Hui Z (2012) Optimizing RFID Network Planning by Using a Particle Swarm Optimization Algorithm With Redundant Reader Elimination. IEEE Trans Ind Appl 8 (4):900-912. doi:10.1109/TII.2012.2205390

4. Eberhart R, Kennedy J A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 1995. IEEE, pp 39-43

5. Yu Q (2015) New Approaches for Automated Intelligent Quality Inspection System Integration of 3D Vision Inspection, Computational Intelligence, Data Mining and RFID Technology.

6. Eberhart RC, Shi Y Comparing inertia weights and constriction factors in particle swarm optimization. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, 2000. IEEE, pp 84-88

7. Liu H, Abraham A, Zhang W (2007) A fuzzy adaptive turbulent particle swarm optimisation. International Journal of Innovative Computing and Applications 1 (1):39-47

8. Shi Y, Eberhart RC Fuzzy adaptive particle swarm optimization. In: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2001. IEEE, pp 101-106

9. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE transactions on Evolutionary Computation 6 (1):58-73

10. Kennedy J, Eberhart RC A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, 1997. IEEE, pp 4104-4108