

Construction Method of Complex Product Repair Model for Process Execution End

Jianqiang Liu and Xiaojun Liu

School of Mechanical Engineering, Southeast University, Nanjing, China

ABSTRACT

The performance of complex products is directly related to their assembly quality, and as a result, the precision requirements for assembling complex products are becoming increasingly stringent. However, in the assembly process, precision deviations are inevitable, often leading to situations where the required accuracy cannot be met. In such cases, rework and adjustment methods are needed to modify certain geometric attributes of the product model to ensure assembly precision. Currently, most rework solutions rely on the personal experience of on-site workers and repeated disassembly and reassembly, lacking scientific guidance. Aiming to improve the accuracy of rework solutions, this paper proposes a rework model construction method tailored for the process execution phase. First, an error propagation function is established based on the Small Displacement Torsor (SDT) and homogeneous transformation matrices. High-precision measurement equipment is then used on-site to collect point cloud data of key features, which is processed to reconstruct the error model of critical features, thereby enabling a more accurate error propagation model. Subsequently, an optimization-based dynamic adjustment mechanism and an interval resampling mechanism are introduced to improve the particle swarm optimization (PSO) algorithm. Using the improved PSO algorithm, more accurate rework plans are generated to guide on-site rework operations. Finally, a case study is conducted using the rework process of a satellite product subassembly structure to validate the proposed method. The results demonstrate that the proposed approach can generate more reasonable and effective rework solutions, not only improving on-site rework efficiency but also increasing the first-time assembly success rate of complex products.

Keywords: SDT, IPSO, Repair plan, Assembly accuracy

INTRODUCTION

The performance of complex products, such as satellites, aircraft, and machine tools, is directly influenced by their assembly quality (2024). Consequently, the precision requirements in the assembly process continue to increase. With the rapid advancement of machining technology, the quality of manufactured parts has significantly improved. However, the impact of error factors in the assembly process on assembly quality has also become increasingly significant (2018). Therefore, in the assembly process of complex products, it is essential to strictly control the assembly accuracy of critical nodes. This is especially true for single-piece and small-batch complex

products like satellites, where there is often no room for selecting alternative components. When the required assembly accuracy cannot be achieved, rework solutions must be provided to guide on-site rework operations.

First, assembly accuracy analysis is key to controlling product assembly quality and is a crucial factor in evaluating whether assembly quality meets the required standards. Liu et al. (2018) used the Small Displacement Torsor (SDT) to model key tolerances along the assembly path of helicopter rotor system components. They then applied the homogeneous coordinate transformation method for error propagation analysis, ultimately obtaining the geometric rotational accuracy of the rotor system. Hu et al. (2024) employed SDT and skin model shapes to construct a comprehensive tolerance model for precision spindles. Zeng et al. (2024) built a coupling model of assembly deviation and static mechanics to analyze the relationship between assembly deviation and the mechanical characteristics of the bearing system. Yi et al. (2024) proposed a digital twin-based assembly accuracy prediction method for complex aerospace product assembly, utilizing virtual-physical integration for assembly process modeling and analysis to enhance assembly quality and performance stability.

On assembly sites, rework methods are typically used to adjust product assembly accuracy, making feasible rework solutions necessary for guidance. Sun (2022) addressed the weak correlation of stable knowledge in assembly processes by proposing a knowledge graph-based representation method for assembly processes. This method leverages the inferential capabilities of knowledge graphs to dynamically adjust assembly processes. Guo et al. (2023) established three single-objective optimization models for machining cost, quality loss, and rework cost. They then constructed a multi-objective tolerance optimization allocation model based on a weight parameter distribution method for multi-objective models, solving it using an accelerated particle swarm optimization algorithm to determine the error values for each component loop. Chen et al. (2022) tackled the issue of unpredictable rework volume for large aerospace structural components. They employed fuzzy comprehensive evaluation to quantify factors such as rework cost and difficulty, thereby recommending rework solutions.

Error Propagation Model Construction Method Based on Twin Data

In the field of assembly error modeling, the Small Displacement Torsor (SDT) model is commonly used to describe the tolerance variations of typical component features (2018). Additionally, homogeneous transformation matrices from rigid body kinematics in robotics are utilized to construct error propagation models (2023). The error propagation model of complex products serves as the foundation for generating rework solutions. Traditional error propagation models rely on the analysis and calculation based on the design data of complex products, which cannot promptly reflect the actual geometric variations of products in the assembly environment. Therefore, it is necessary to collect point cloud data using high-precision measurement equipment at the assembly site after completing critical assembly steps. This enables the reverse reconstruction of key features. The specific implementation steps are shown in Figure 1.

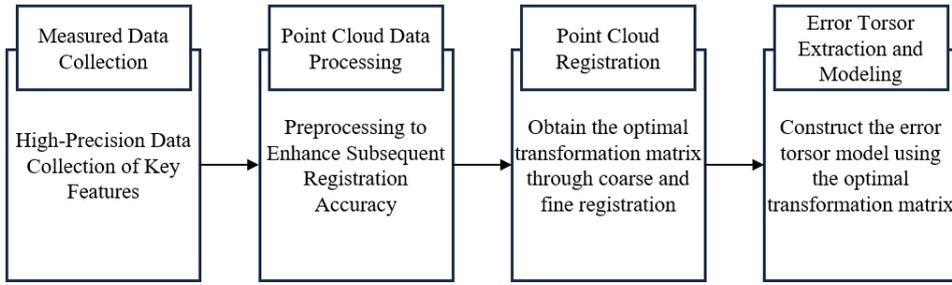


Figure 1: The process of key feature reconstruction.

After completing the actual assembly of critical assembly steps, high-precision measurement equipment at the assembly site is used to obtain the measured geometric feature data. Once the raw point cloud data is collected, preprocessing techniques such as denoising, hole filling, and data downsampling are applied to improve the accuracy of subsequent point cloud registration and enhance analytical efficiency.

Point cloud registration is the core step for aligning the measured data with the design model. Its objective is to eliminate pose deviations between the two through geometric transformations, enabling the precise extraction of assembly errors. Based on the preprocessed point cloud, the reconstruction of key assembly features is achieved through three main steps: coarse registration, fine registration, and error torsor extraction.

Coarse registration aims to rapidly estimate the initial transformation matrix between the measured point cloud and the design model by extracting prominent geometric features, thereby providing a good initial pose for subsequent fine registration. Its core process includes feature description, correspondence matching, and transformation matrix computation.

First, the Fast Point Feature Histogram (FPFH) descriptor is used to characterize the local geometric features of the point cloud. FPFH is highly robust to noise and variations in point cloud density, making it suitable for representing complex geometric features. By statistically analyzing the geometric property distribution within a point's neighborhood, FPFH generates a highly discriminative 33-dimensional feature vector. For a given point p_i , the computation of its FPFH features consists of three main steps: neighborhood construction, geometric property calculation, and histogram statistics.

Step 1: Neighborhood Construction

Centering on point p_i , a spherical neighborhood with a radius of $r = 5mm$ is constructed, containing neighboring $k = 50$ points.

Step 2: Geometric Property Calculation

Step 2.1: Calculation of the Normal Vector Angle

The angle between the normal vectors of point pair (p_i, p_j) is computed as:

$$\alpha = \arccos(\vec{n}_i \cdot \vec{n}_j)$$

where \vec{n}_i and \vec{n}_j are the normal vectors of points p_i and p_j , respectively.

Step 2.2: Calculation of Local Curvature

The local curvature is computed as:

$$\varnothing = \lambda_1 / (\lambda_1 + \lambda_2 + \lambda_3)$$

where $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the eigenvalues of the covariance matrix of the neighborhood.

Step 3: Histogram Statistics

The parameters α and \varnothing are discretized into a 33-dimensional histogram, generating the feature vector $f_i \in R^{33}$.

Next, the Random Sample Consensus (RANSAC) algorithm is applied to filter out the optimal correspondence point set and estimate the initial transformation matrix H_{0H_0} . RANSAC enhances iterative random sampling and model verification by randomly selecting three non-collinear point pairs from the matched point set to construct a minimal sample set and compute the rigid transformation matrix. The specific process is as follows:

Step 1: Random Sampling

Randomly select three non-collinear point pairs from the matched point set to form a minimal sample set (Minimal Set).

Step 2: Model Hypothesis

Based on the minimal sample set, compute the candidate rigid transformation matrix $H_{candidate}$, which includes the rotation matrix R and the translation vector \vec{t} .

Step 3: Inlier Selection

Set a distance threshold $d_{max} = 0.1\text{mm}$ and count the number of inliers that satisfy:

$$\left\| p_i^{meas} - H_{candidate} \cdot p_i^{design} \right\| \leq d_{max}$$

Step 4: Iterative Optimization

Repeat the above process $N = 1000$ times, and select the transformation matrix $H_{candidate}$ with the highest inlier ratio as the initial transformation matrix H_0 . The mathematical expression is:

$$H_0 = \arg \max_H \left(\frac{\text{Internal points}}{\text{Total points}} \right) \quad (1)$$

The physical significance of the initial transformation matrix H_0 is to map the design model from the ideal coordinate system to the measured coordinate system. Its mathematical form is given by:

$$H_0 = \begin{bmatrix} R_0 & t_0 \\ 0^T & 1 \end{bmatrix} \quad (2)$$

In the equation, $R_0 \in SO(3)$ represents the rotation matrix, and $t_0 \in R^3$ represents the translation vector. After coarse registration, the average alignment error between the measured point cloud and the design model is typically reduced to within 1 mm.

The accuracy of coarse registration is insufficient to support the precision prediction of high-precision complex products. Therefore, fine registration is required to achieve higher accuracy in point cloud alignment. Fine registration is performed using the Iterative Closest Point (ICP) algorithm, which further optimizes the transformation matrix and enables sub-millimeter-level point cloud alignment. The ICP algorithm is one of the most widely used fine registration methods in point cloud processing. Its objective is to iteratively optimize the alignment by minimizing the distance error between corresponding points in the two point clouds, ultimately achieving high-precision alignment.

First, a three-dimensional KD-tree (K-Dimensional Tree) data structure is constructed for the design model point cloud P_{design} , reducing the time complexity of nearest neighbor search from $O(N^2)$ to $O(N \log N)$. A bidirectional consistency check is then performed to ensure the accuracy of the corresponding point pairs. Next, the Singular Value Decomposition (SVD) method is used to solve for the optimal transformation matrix. For the k -th iteration, given N pairs of corresponding points, the optimal transformation matrix is obtained by minimizing the objective function. After decentralizing the coordinates, the covariance matrix is computed and decomposed using SVD to derive the optimal rotation matrix and translation vector. The specific steps are as follows:

Step 1: Nearest Point Search

Utilizing a KD-tree for accelerated searching, each point p_i^{source} in the source point cloud is matched with its nearest corresponding point p_i^{target} in the target point cloud based on geometric distance.

Step 2: Transformation Matrix Estimation

Based on the corresponding point pairs, the optimal rigid transformation $T = (R, t)$ is solved using the **least squares method**. The specific steps are as follows:

Step 2.1: Compute the Centroids of the Source and Target Point Clouds

$$\mu_{source} = \frac{1}{N} \sum_{i=1}^N p_i^{source} \quad (3)$$

$$\mu_{target} = \frac{1}{N} \sum_{i=1}^N p_i^{target} \quad (4)$$

Step 2.2: Covariance Matrix Construction

$$C = \sum_{i=1}^N (p_i^{source} - \mu_{source}) (p_i^{target} - \mu_{target})^T \quad (5)$$

Step 2.3: SVD Decomposition

Perform Singular Value Decomposition (SVD) on the covariance matrix C

$$C = U \Sigma V^T \quad (6)$$

The optimal rotation matrix and translation vector are given by:

$$R = VU^T \quad (7)$$

$$t = \mu_{target} - R\mu_{source} \quad (8)$$

Step 3: Convergence Criteria and Termination

The ICP algorithm iteratively adjusts the transformation matrix until the registration error is minimized or the predefined maximum number of iterations is reached. A common convergence criterion is that the change in registration error falls below a certain threshold. Specifically, the iteration terminates when the change in Root Mean Square Error (RMSE) between two consecutive iterations is smaller than the threshold. The residual error at the k -th iteration is denoted as $\varepsilon^{(k)}$:

$$\varepsilon^{(k)} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|R^{(k)}p_i^{source} + t^{(k)} - p_i^{target}\|^2} \quad (9)$$

Additionally, a maximum iteration count k_{max} can be set to prevent the algorithm from getting stuck in an infinite loop.

By following the above steps, the ICP algorithm continuously optimizes registration accuracy until it converges to the optimal solution, thereby achieving precise alignment between point clouds.

After fine registration, the transformation matrix needs to be decomposed into rotation and translation components to construct a SDT model for quantifying assembly errors. The rotation matrix can be converted into a rotation vector using the matrix logarithm mapping. The rotation vector and translation vector are then combined into an error torsor, which intuitively represents deviations in six degrees of freedom (three translations and three rotations), providing a mathematical basis for subsequent tolerance analysis.

Following the above process, the optimal transformation matrix H_{meas} obtained from point cloud registration using the ICP algorithm is:

$$H_{meas} = \begin{bmatrix} R_{meas} & t_{meas} \\ 0^T & 1 \end{bmatrix} \quad (10)$$

Rotation vector extraction: The rotation matrix is converted into a rotation vector θ_{meas} using the matrix logarithm mapping:

$$\theta_{meas} = \log(R_{meas}) \quad (11)$$

The specific calculation process is as follows:

$$\theta = \arccos\left(\frac{\text{tr}(R) - 1}{2}\right) \quad (12)$$

$$\vec{u} = \frac{1}{2 \sin \theta} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \quad (13)$$

$$\theta_{meas} = \theta \vec{u} \quad (14)$$

where \vec{u} is the unit vector of the rotation axis, and θ is the rotation angle.

Translation Vector Extraction: The translation component is directly obtained as t_{meas} .

Summary, The error torsor model T_{error} is given by:

$$T_{error} = \begin{bmatrix} \theta_{meas} \\ t_{meas} \end{bmatrix} \quad (15)$$

By applying the above methods, the reverse reconstruction of key features can be achieved, allowing the correction of the model using measured data. This process helps in constructing a more accurate error propagation model, thereby laying a stronger foundation for the generation of subsequent rework solutions.

A Rework Solution Generation Method Based on an Improved Particle Swarm Optimization Algorithm

In the assembly process of complex products, when the assembly accuracy does not meet the inspection requirements, it is necessary to rework certain structural features of components. However, the current rework approach primarily relies on the experience of on-site workers, using repeated trial assembly, measurement, and adjustment methods. Since the rework range is often estimated roughly, the entire process involves significant uncertainty, making it difficult to ensure rework quality while balancing cost control and operational efficiency.

Since its introduction, the particle swarm optimization (PSO) algorithm has been widely applied in various fields and has become an essential tool for solving complex optimization problems. With further research, many improved PSO algorithms (such as hybrid PSO and modified PSO) have been proposed, expanding its adaptability and optimization effectiveness in practical applications.

In the field of assembly, the application of PSO for generating rework solutions still faces the following challenges:

1. In the assembly process, there are usually multiple constraints, such as assembly sequence, spatial limitations, and types of geometric tolerances. Since these constraints are often nonlinear and non-convex, the solution space is typically highly complex and contains multiple local extrema, making it easy for PSO to fall into a local optimal solution, resulting in poor optimization results.
2. The optimization results of the PSO algorithm are specific numerical values rather than feasible rework intervals, which does not meet the practical requirements of rework.

Therefore, to better apply the method for generating rework solutions, this paper proposes a rework solution generation strategy based on an improved particle swarm optimization (IPSO) algorithm, as shown in Figure 2, which mainly adds the following mechanisms:

1. **Optimization Dynamic Adjustment Mechanism (ODAM):** This introduces random weights and asynchronous learning factors, allowing

the particles' velocities to be dynamically adjusted during iterations, thus avoiding the algorithm from falling into local optimal solutions.

2. **Repair Interval Upsampling Mechanism (RIUM):** In this paper, the optimization targets for IPSO particles are the geometric tolerances along the error propagation path. Based on the characteristics of the error propagation model, the relationship between geometric tolerances and assembly accuracy of the target features is linear. Therefore, the repair interval upsampling mechanism is introduced to record feasible solutions that meet inspection requirements, generating multiple sets of feasible rework solutions. The upsampling method, inspired by signal processing, forms the repair interval upsampling mechanism to create feasible rework intervals.

(1) Optimization Dynamic Adjustment Mechanism

Therefore, this paper introduces random weights and asynchronous learning factors to enhance the optimization capability of the algorithm. The iterative update equations for the velocity and position in the improved particle swarm optimization (IPSO) algorithm are given as follows:

$$v_i(t+1) = \omega \cdot v_i(t) + c_1 \cdot r_1 \left(p_i^{best} - x_i(t) \right) + c_2 \cdot r_2 \left(g_i^{best} - x_i(t) \right) \quad (16)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (17)$$

Where: $v_i(t)$ represents the velocity of the i -th particle at iteration t , $x_i(t)$ represents the position of the i -th particle at iteration t , ω is the random relationship weight, c_1 and c_2 are asynchronous learning factors, r_1 and r_2 are random factors, typically generated within the range $[0,1]$, p_i^{best} is the historical best position of particle i , g_i^{best} is the global best position.

The random weight is defined as:

$$\omega = \mu_{min} + rand() \times (\mu_{max} - \mu_{min}) + \sigma \times randn() \quad (18)$$

where: μ_{min} and μ_{max} are the minimum and maximum values of the random inertia weight, respectively, $rand()$ is a uniformly distributed random number in the range $[0, 1]$, $randn()$ is a normally distributed random number with a mean of 0 and a standard deviation of 1, σ is the standard deviation, which controls the magnitude of the random fluctuation.

The asynchronous learning factors are defined as:

$$c_1 = (c_{1e} - c_{1s}) (t - t_{max}) / t_{max} + c_{1e} \quad (19)$$

$$c_2 = (c_{2e} - c_{2s}) (t_{max} - t) / t_{max} + c_{2e} \quad (20)$$

where: c_{1s} and c_{2s} are the starting values of the learning factors c_1 and c_2 , with $c_{1s} = c_{2s} = 2.5$, c_{1e} and c_{2e} are the ending values of the learning factors c_1 and c_2 , ensuring that c_1 and c_2 change asynchronously.

(2) Repair Interval Upsampling Mechanism (RIUM)

To enhance the exploration capability of feasible solution spaces for geometric tolerances in complex assembly systems, this paper proposes a repair interval upsampling mechanism (RIUM). The core idea is to integrate the particle swarm optimization algorithm with the upsampling concept from the signal processing domain, expanding the feasible solution set through interpolation and generating multiple rework intervals.

First, during the iterative process of the improved PSO algorithm, feasible solutions that satisfy the constraint conditions in each dimension are recorded to construct an initial feasible solution set $F = \{x_1, x_2, \dots, x_n\}$, where x_i represents a feasible solution in a specific dimension, and N is the total number of feasible solutions.

Based on this set, considering that the distribution of feasible solutions in the solution space exhibits non-uniform characteristics, a Gaussian Process Interpolation (GPI) method is adopted to balance local precision and global adaptability.

Subsequently, the feasible solution set undergoes interpolation to generate candidate solutions. Specifically, for two adjacent feasible solutions x_i and x_j , an intermediate solution x_{ij} is generated using an interpolation coefficient ($0 < \alpha < 1$). The expression is given by:

$$x_{ij} = x_i + \alpha (x_j - x_i) \quad (21)$$

By repeating this operation, an upsampled candidate solution set $F_{upsampled}$ is formed. Candidate solutions that do not meet the constraint conditions are then eliminated to ensure the feasibility of the solutions.

Furthermore, statistical analysis is conducted to determine the range of the rework interval. The mean μ and standard deviation σ of the candidate solution set are calculated as follows:

$$\mu = \frac{1}{M} \sum_{k=1}^M x_k \quad (22)$$

$$\sigma = \sqrt{\frac{1}{M} \sum_{k=1}^M (x_k - \mu)^2} \quad (23)$$

where M is the number of candidate solutions. Based on this, the rework interval can be defined as:

$$T_{\text{interval}} = [\mu - k \cdot \sigma, \mu + k \cdot \sigma] \quad (24)$$

In the equation, the expansion coefficient k is used to adjust the interval width. Finally, the repair interval T_{interval} is divided into multiple sub-intervals, with each sub-interval corresponding to a set of feasible rework solutions. This process forms a high-density feasible region within the solution space. This mechanism dynamically expands the solution space, effectively enhancing the diversity and robustness of rework solutions. It provides a more comprehensive optimization foundation for complex assembly scenarios.

When the assembly accuracy of critical assembly steps in the complex product assembly process does not meet the required standards, the improved particle swarm optimization algorithm can enhance the efficiency of particle optimization and prevent it from getting trapped in local optima. This enables the generation of more efficient and precise rework solutions. The dimensions of the particle swarm represent the number of rework regions, and particles perform optimization iterations within the solution space. Each particle corresponds to a generated rework solution. After each iteration, the results are substituted into the assembly accuracy analysis model to obtain the corresponding assembly accuracy, which serves as the function fitness. The rework solution generation method based on IPSO, as shown in Figure 2, yields feasible rework solutions.

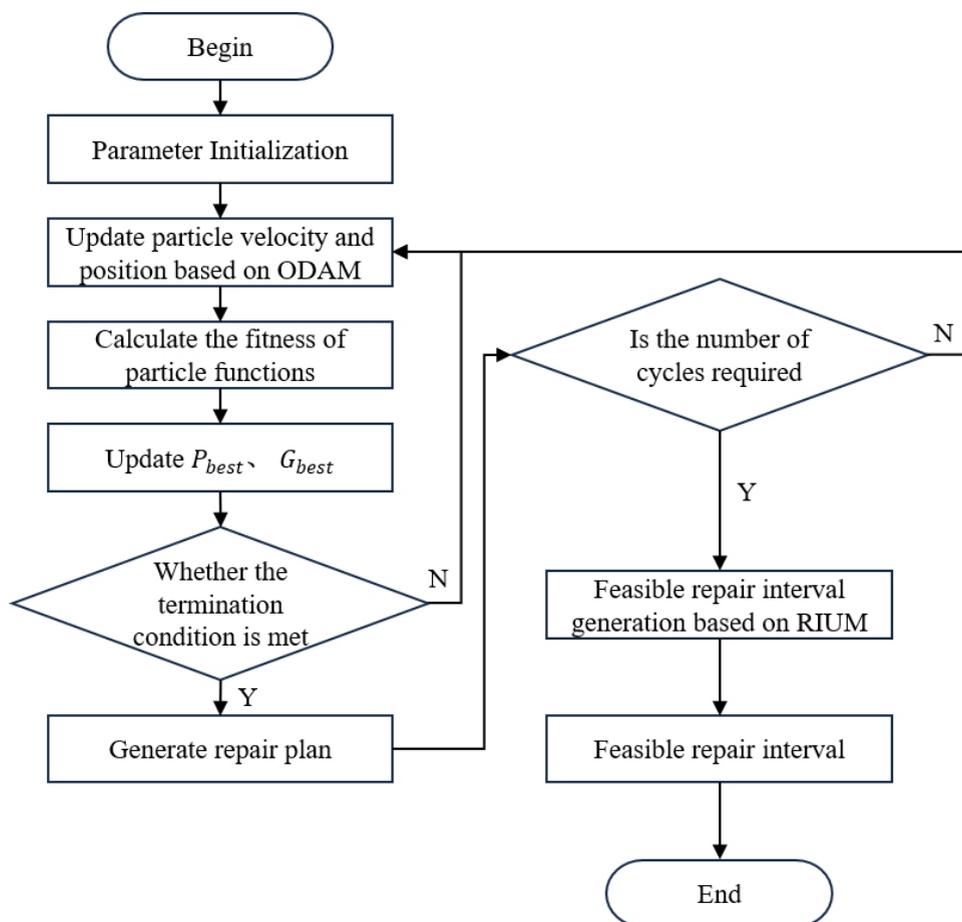


Figure 2: Method for generating repair plans based on IPSO.

CASE STUDY

To validate the effectiveness of the proposed rework solution generation method for the process execution phase, a case study is conducted using the assembly process of a satellite product subassembly structure. This

subassembly structure consists of multiple assembly steps, including the base plate, vertical plates, intermediate partitions, side panels, and top plate, involving 15 error terms in the error propagation paths.

By performing reverse reconstruction of key features and generating rework solutions, the accuracy of the proposed method is verified.

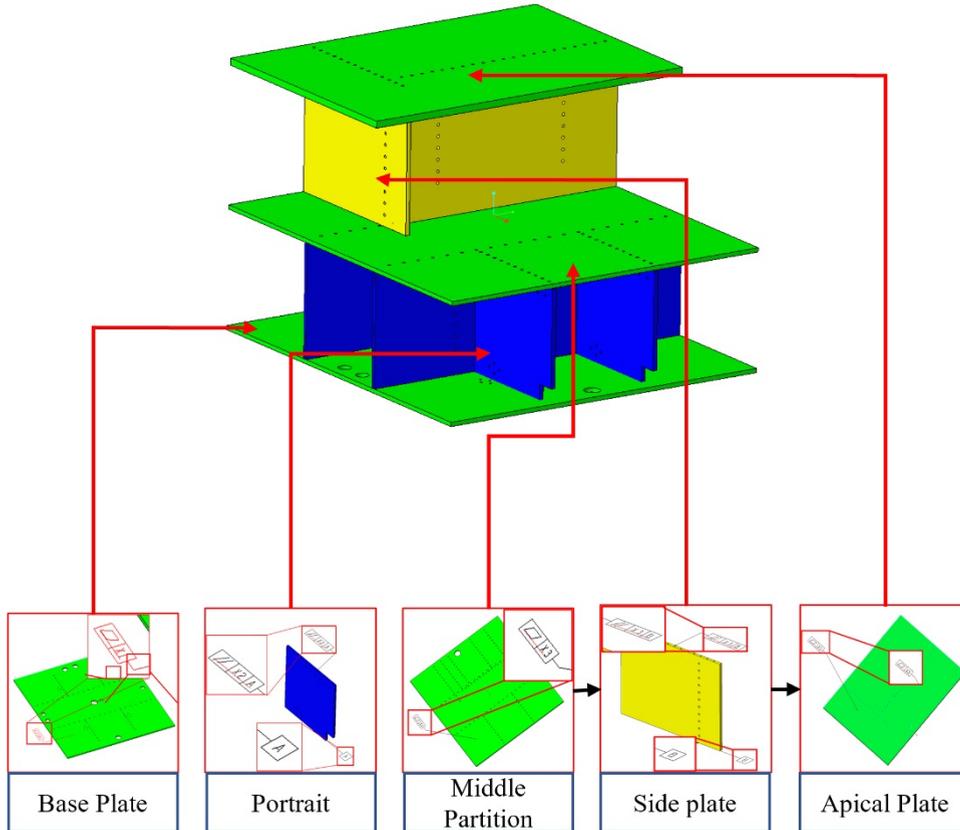


Figure 3: Transmission path of structural errors in a certain satellite component.

Table 1: Structural error term and number of a certain satellite.

Geometric Features	Error Term	Error Number
Surface on the base plate	$\delta_{Z1}(X), \varepsilon_{X1}(X), \varepsilon_{Y1}(X)$	1, 2, 3
Surface on the portrait	$\delta_{Z2}(X), \varepsilon_{X2}(X), \varepsilon_{Y2}(X)$	4, 5, 6
Surface on the middle partition	$\delta_{Z3}(X), \varepsilon_{X3}(X), \varepsilon_{Y3}(X)$	7, 8, 9
Surface on the side plate	$\delta_{Z4}(X), \varepsilon_{X4}(X), \varepsilon_{Y4}(X)$	10, 11, 12
Surface on the apical plate	$\delta_{Z5}(X), \varepsilon_{X5}(X), \varepsilon_{Y5}(X)$	13, 14, 15

Taking the satellite product subassembly structure shown in Figure 3 as an example, high-precision measurement equipment (laser tracker) is used to collect actual assembly data. The error term values range from $\pm 50 \mu\text{m}$ (positional error) to ± 0.001 rad (angular error).

Using the base plate mounting surface as a case study, its geometric features are reconstructed from high-precision point cloud data, and the error torsor model is extracted. A 3D scan of the base plate mounting surface is performed, capturing approximately 60,000 raw point cloud data points with a point spacing of 0.1 mm and a scanning accuracy of ± 0.02 mm.

The point cloud data is then preprocessed, reducing the number of points to 40,000, achieving a retention rate of approximately 67%. A subset of the measured point coordinate data is shown in Figure 4.

文件	编辑	查看	文件	编辑	查看	文件	编辑	查看			
p101	5000.000	6000.000	7000.000	p49	2500.000	3000.000	3500.000	底板.txt	1040.345	1530.456	1960.567
p102	5005.123	5995.234	7010.345	p50	2505.123	2995.234	3505.345	p9	1045.678	1465.789	2050.890
p103	5010.456	6005.567	6990.678	p51	2510.456	3005.567	3490.789	p10	1050.012	1540.123	1950.234
p104	5015.789	5990.890	7020.901	p52	2515.789	2990.890	3480.012	p11	1055.345	1455.456	2060.567
p105	5020.012	6010.123	6980.234	p53	2520.012	3010.123	3470.345	p12	1060.678	1550.789	1940.890
p106	5025.345	5985.456	7030.567	p54	2525.345	2985.456	3460.678	p13	1065.012	1445.123	2070.234
p107	5030.678	6020.789	6970.890	p55	2530.678	3020.789	3450.901	p14	1070.345	1560.456	1930.567
p108	5035.012	5975.123	7040.234	p56	2535.012	2975.123	3440.234	p15	1075.678	1435.789	2080.890
p109	5040.345	6030.456	6960.567	p57	2540.345	3030.456	3430.567	p16	1080.012	1570.123	1920.234
p110	5045.678	5965.789	7050.890	p58	2545.678	2965.789	3420.890	p17	1085.345	1425.456	2090.567
p111	5050.012	6040.123	6950.234	p59	2550.012	3040.123	3410.123	p18	1090.678	1580.789	1910.890
p112	5055.345	5955.456	7060.567	p60	2555.345	2955.456	3400.456	p19	1095.012	1415.123	2100.234
p113	5060.678	6050.789	6940.890	p61	2560.678	3050.789	3390.789	p20	1100.345	1590.456	1900.567
p114	5065.012	5945.123	7070.234	p62	2565.012	2945.123	3380.012	p21	1105.678	1405.789	2110.890
p115	5070.345	6060.456	6930.567	p63	2570.345	3060.456	3370.345	p22	1110.012	1600.123	1890.234
p116	5075.678	5935.789	7080.890	p64	2575.678	2935.789	3360.678	p23	1115.345	1395.456	2120.567
p117	5080.012	6070.123	6920.234	p65	2580.012	3070.123	3350.901	p24	1120.678	1610.789	1880.890
p118	5085.345	5925.456	7090.567	p66	2585.345	2925.456	3340.234	p25	1125.012	1385.123	2130.234
p119	5090.678	6080.789	6910.890	p67	2590.678	3080.789	3330.567	p26	1130.345	1620.456	1870.567
p120	5095.012	5915.123	7100.234	p68	2595.012	2915.123	3320.890	p27	1135.678	1375.789	2140.890
p121	5100.345	6090.456	6900.567	p69	2600.345	3090.456	3310.123	p28	1140.012	1630.123	1860.234
p122	5105.678	5905.789	7110.890	p70	2605.678	2905.789	3300.456	p29	1145.345	1365.456	2150.567
p123	5110.012	6100.123	6890.234	p71	2610.012	3100.123	3290.789	p30	1150.678	1640.789	1850.890
p124	5115.345	5895.456	7120.567	p72	2615.345	2895.456	3280.012	p31	1155.012	1355.123	2160.234
p125	5120.678	6110.789	6880.890	p73	2620.678	3110.789	3270.345	p32	1160.345	1650.456	1840.567
p126	5125.012	5885.123	7130.234	p74	2625.012	2885.123	3260.678	p33	1165.678	1345.789	2170.890
p127	5130.345	6120.456	6870.567	p75	2630.345	3120.456	3250.901	p34	1170.012	1660.123	1830.234
p128	5135.678	5875.789	7140.890	p76	2635.678	2875.789	3240.234	p35	1175.345	1335.456	2180.567
p129	5140.012	6130.123	6860.234	p77	2640.012	3130.123	3230.567	p36	1180.678	1670.789	1820.890
p130	5145.345	5865.456	7150.567	p78	2645.345	2865.456	3220.890	p37	1185.012	1325.123	2190.234
p131	5150.678	6140.789	6850.890					p38	1190.345	1680.456	1810.567

Figure 4: Actual measurement data of the base plate installation surface (partial).

Based on the RANSAC algorithm, three sets of non-collinear points are randomly sampled. The inlier threshold is set as $d_{max} = 0.1$ mm. After 1000 iterations, the initial transformation matrix H_0 is obtained:

$$H_0 = \begin{bmatrix} 0.9998 & 0.0053 & -0.0172 & 0.12 \\ -0.0051 & 0.9999 & 0.0086 & -0.09 \\ 0.0173 & -0.0085 & 0.9998 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

Then, based on the ICP algorithm, fine registration is performed. After 15 iterations, the final transformation matrix H_{meas} is obtained:

$$H_{meas} = \begin{bmatrix} 0.9999 & 0.002 & -0.003 & 0.01 \\ -0.002 & 0.9998 & 0.001 & -0.02 \\ 0.003 & -0.001 & 0.9999 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

Based on the final transformation matrix H_{meas} , the error torsor is extracted. Using equations (11) to (14), the calculated rotation angle is:

$$\theta_X = \arcsin(R_{32}) = \arcsin(-0.001) \approx -0.001rad \quad (27)$$

$$\theta_Y = \arcsin(R_{13}) = \arcsin(-0.003) \approx -0.003rad \quad (28)$$

The translation vector can be directly obtained as:

$$t_Z = -0.05 \text{ mm} \quad (29)$$

Therefore, the error torsor model T_{error} for the reconstructed base plate mounting surface feature based on point cloud data is given by:

$$T_{error} = \begin{bmatrix} t_Z \\ \theta_X \\ \theta_Y \end{bmatrix} = \begin{bmatrix} 0.005 \\ -0.001 \\ -0.003 \end{bmatrix} \quad (30)$$

Using Creo software, a sensitivity analysis of the model is conducted, revealing that the upper surface of the intermediate partition has high sensitivity. Therefore, rework simulation is performed specifically for this surface.

The particle swarm is configured with a population size of 50 and a maximum of 200 iterations. The learning factor c_1 decreases linearly from 2.5 to 1.0, while c_2 increases from 1.0 to 2.5. The inertia weight ω is set within the range [0.4, 0.9].

During the particle swarm iteration process, feasible solutions satisfying the constraints are recorded as $F = \{-0.07, -0.08, -0.09\}$. Using equation (21) for interpolation between adjacent solutions with $\alpha = 0.05$, intermediate solutions are generated as $x_{12} = -0.075, x_{23} = -0.085$, expanding the candidate solution set to $\{-0.07, -0.075, -0.08, -0.085, -0.09\}$. Substituting these values into the error propagation function confirms that they all meet the accuracy requirements.

Then, using equations (22) and (23), the mean is calculated as $\mu = -0.08$ and the standard deviation as $\sigma = 0.007$. Applying equation (24), the feasible rework interval is:

$$T_{interval} = [\mu - 1.5\sigma, \mu + 1.5\sigma] = [-0.0905, -0.0695]$$

In conclusion, when the top plate assembly accuracy does not meet the requirements, the interference error in the Z direction can be eliminated by filing the embedded parts on the contact surface between the intermediate partition and the vertical plate. The feasible rework range is $[-0.0905, -0.0695] \text{ mm}$.

Implementing the rework process based on the proposed method significantly improves the feasibility and efficiency of the rework solution, providing scientific guidance for the rework of complex products.

CONCLUSION

This paper addresses the issue of generating rework schemes in the complex product assembly process and proposes a rework model construction method oriented toward the process execution end. High-precision measurement equipment is used to obtain point cloud data of target features, enabling reverse reconstruction of these features based on the point cloud data. This approach facilitates the construction of a more accurate error propagation model. Additionally, an Optimization Dynamic Adjustment Mechanism (ODAM) and a Repair Interval Upsampling Mechanism (RIUM) are introduced to enhance the particle swarm optimization (PSO) algorithm. Based on the improved PSO algorithm, feasible rework intervals are generated. Case studies validate that this method effectively improves the first-pass assembly success rate and rework efficiency for complex products.

Future research can further optimize the algorithm for generating rework schemes by incorporating additional practical assembly constraints and verifying the model with more real-world cases to enhance its accuracy and applicability. Furthermore, with the continuous advancement of digital twin technology, the real-time performance and intelligence level of the rework model will be further improved, providing more reliable assurance for the assembly quality of complex products.

REFERENCES

- Chen S, Bai T, Zhang Y, Fang Z, Chen Z. Geometric error suppression of six-axis machine tool for blisk full-shape surface grinding via constrained error sensitivity analysis. *Precis Engineering*, 2024;88: 1–14.
- Chen Shuai, Guo Feiyan, Meng Yuemei, et al. Iterative Optimization and Evaluation Method for Repair Quantity of Aviation Structural Parts Considering Measured Data[J]. *China Mechanical Engineering*, 2022, 33(17):2061-2070+2078.
- Guo Feiyan, Song Changjie, Zhang Shuo, et al. Multi-objective model fusion method for tolerance allocation orienting to accuracy and process improvement[J]. *Computer Integrated Manufacturing Systems*, 2023, 1–28.
- Hu Xiaokun, Yang Yitao, Zhao Qiangqiang, et al. Rotation accuracy analysis for precision spindles based on skin model[J]. *Aeronautical Manufacturing Technology*, 2024, 67(16): 74–84.
- J. W. Cheng, W. J. Bu, L. Shi, et al., A real-time shaft alignment monitoring method adapting to ship hull deformation for marine propulsion system, *ech. Syst. Signal Process.*197(2023)110366.
- Khan, H. A.; Butt, S. U.; Baqai, A. A.; Saeed, H. A. A new approach to predict the flexibility and precision of manufacturing systems using geometric constraints and small displacement torsors. *Procedia Manuf.* 2018, 17, 294–301.
- Liu Jianhua, Sun Qingchao, Cheng Hui, et al. The State-of-the art, connotation and developing trends of the products assembly technology[J]. *Journal of Mechanical Engineering*, 2018, 54(11): 2–28.
- Sun Xuemin. Theoretical Research and Application of High Precision Products Assembly-Commissioning Driven by Digital Twin[D]. Shanghai: Donghua University, 2022.
- Yi Y., Zhang A. Q., Liu X. J., et al. Digital twin-driven assembly accuracy prediction method for high performance precision assembly of complex products[J]. *Advanced Engineering Informatics*, 2024, 61:102495.
- Zeng W. H., Lu W. L., Xun F. Model-based low-speed rotation error prediction for the rigid shaft-bearing system considering the assembly deviation[J]. *Mechanism and Machine Theory*, 2023, 184:105294.