Multi-dimensional Error Identification during Robotic Snap Assembly

Yusuke Hayami[1], Peihao Shi[1], Weiwei Wan[1,2], Ixchel G. Ramirez-Alpizar[1], and Kensuke Harada[1,2]

[1] Osaka University, Toyonaka 560-8531, Japan
[2] National Inst. of Advanced Industrial Science and Technology, Tsukuba 305-8651, Japan

Abstract. In this research, we propose a novel error identification method during robotic snap assembly aiming at automated recovery from error states. In the proposed method, we first obtain the feature quantities of force/torque from the simulated snap assembly by using functional principal component analysis(fPCA). Then, we cluster these data into success and several different error states based on the k-means clustering by using the decision tree considering the multi-dimensional feature of the force/torque signal. Furthermore, we try to predict an error state of a snap assembly task before the error actually happens. Finally, we show simulation results to show the effectiveness of the proposed method.

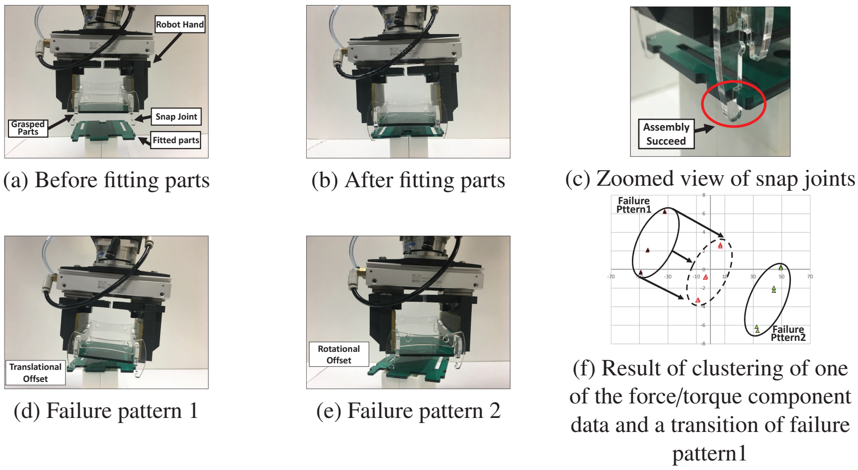
Keywords: Snap assembly, K-means clustering, Functional PCA

1 Introduction

In recent years, the robotic assembly has been introduced to production processes. However, some complicated assembly tasks remain difficult for robots to perform. Among such difficult assembly tasks, we focus on the so-called ”snap assembly” including parts fitting of snap joints. Performing the snap assembly is difficult due to the following two reasons. First, since the parts are usually made of plastics, we have to carefully treat the parts to avoid the robot to break the parts especially when the assembly fails. Secondly, it is extremely difficult to disassemble the parts with snap joints. Hence, it becomes crucial to predicting the error states before the error actually happens. Moreover, if the robot is to retry the assembly of parts, it becomes important to identify the error state rather than just discriminating failure cases from the successful cases of assembly. Fig. 1(a),$\cdots$,(e) show the overview of a snap assembly task and two typical failure patterns where the snap assembly failed due to the offsets in the initial pose of the parts.

Although identification of error states in the snap assembly has been researched ([1][2][3][4][5][6][7][8][9][10]), those methods can identify the error states only after finishing the assembly task and cannot identify the error states in the middle of an assembly task. To cope with the above problem in the snap assembly, we propose an identification method of error states in the middle of an assembly task.

Our proposed method has two distinctive features. The first is considering the six-dimensional feature of the force/torque information. Among six-dimensional force/torque information, only a few components mainly affect the error states in most of the cases.



(a) Before fitting parts (b) After fitting parts (c) Zoomed view of snap joints

(d) Failure pattern 1 (e) Failure pattern 2 (f) Result of clustering of one of the force/torque component data and a transition of failure pattern1

Fig. 1. Overview of a snap assembly task, its two typical failure patterns, cluster, and its transition in the feature space

Moreover, the components including necessary information to classify the error states depend on the error states (Fig. 1(f)). Taking this problem into account, our method tries to find such components by using a decision tree of the force/torque components.

Secondly, our method tries to predict the error states before the error actually happens. We define the feature quantity used to classify the error states by using the scores of the functional principal component analysis (fPCA) of the force/torque data. Then, the error states are classified by using the k-means clustering method. To identify the error state in the middle of an assembly task, we consider using the force/torque data between the initial time and the middle time of an assembly task. We will check how this middle time can be set far from the end time (Fig. 1(f)).

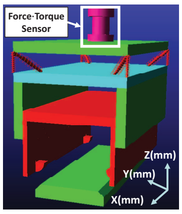
In addition, to easily collect enough number of training data, we construct a physics simulator of robotic snap assembly.

In this paper, we introduce related researches on error identification in the robotic snap assembly in Section 2. Then, we explain the proposed method in Section 3. To show the effectiveness of the proposed method, we show the results of numerical simulation in Section 4.

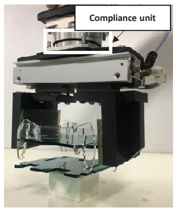
2 Related Works

There have been several works on identifying the success and failure of robotic assembly tasks([1][2][3][4][5][6][7]). For example, Rodriguez et al. [1] identified the result of assembly task by using SVM and PCA. On the other hand, the number of researches on robotic assembly identifying the error states is not large([8][9][10]). Rojas et al. [8][10] proposed an identification method of the error states in cantilever-type snap assembly tasks. Enrico Di Lello et al. [9] used the Bayesian-sequential model for identifying.

However, the above methods cannot identify the error states before the error actually happens.



(a)Simulated snap assembly



(b)Real snap assembly

Fig. 2. Comparison between simulated and real snap assembly environments

3 Proposed Method

Our proposed method consists of three steps for building the classifier of the error states and a single step for its verification. For building the classifier, we first collect the training data through physics simulation. Secondly, we extract feature quantities from these data. We thirdly classify those feature quantities. To verify the classifier, we use feature quantities obtained from the force/torque data starting from the initial state to the middle of assembly tasks.

3.1 Training Data Collection

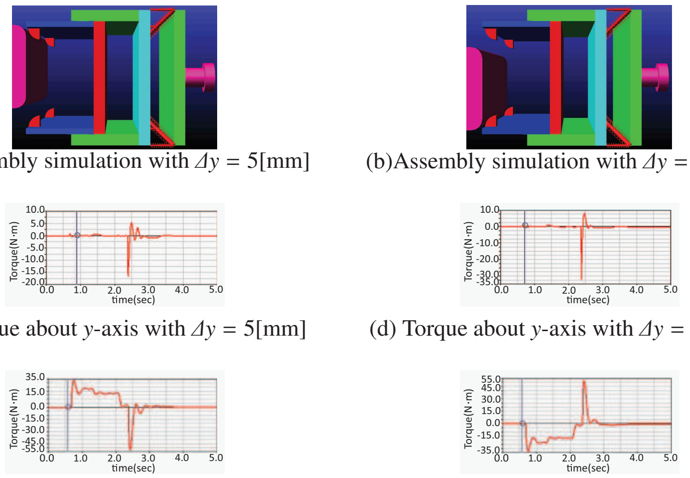
We collect six-dimensional time trajectories of force/torque data from simulated snap assembly [11]. We show the environment for simulated assembly in Fig. 2 in comparison with the actual assembly environment. To avoid excessive force applied to the parts during tasks, we attach a spring unit at the wrist as shown in the Fig. 2. We assume six-axis force/torque sensor attached at the wrist. To collect the training data, the part held by a gripper moves in the $-z$ direction with changing the offset of the fitted part's initial pose. Let Δx, Δy and $\Delta\theta$ be the offsets of the fitted part's initial pose of the translation in x direction, translation in y direction and rotation about z-axis, respectively. Proposed method tries to classify the error states according to the signature of Δx, Δy and $\Delta\theta$.

3.2 Feature Quantity Extraction

So as to identify the error states before the error actually happens, we define the feature quantities by considering the functional principal component analysis(fPCA) [12][13][14] to the 6-dimensional waveform data obtained from the force/torque sensor. For the j-th component of the foce/torque data $f_{ij}(t)$ obtained during the i-th trial of assembly task ($i = 1, \cdots, N$, $j = 1, \cdots, 6$), the k-th principal function $\xi_{jk}(s)$ ($k = 1, \cdots, p$) is obtained by solving the following eigen value problem:

$$\int v(s,t)\xi_{jk}(s)dt = \rho_{jk}\xi_{jk}(t) \quad (1)$$

$$v(s,t) = \frac{1}{N}\sum_{i=1}^{N}(f_{ij}(s) - \bar{f}(s))(f_{ij}(t) - \bar{f}(t)) \quad (2)$$



(a)Assembly simulation with $\Delta y = 5$[mm]
(b)Assembly simulation with $\Delta y = -5$[mm]
(c) Torque about y-axis with $\Delta y = 5$[mm]
(d) Torque about y-axis with $\Delta y = -5$[mm]
(e) Torque about x-axis with $\Delta y = 5$[mm]
(f) Torque about x-axis with $\Delta y = -5$[mm]

Fig. 3. Waveforms of torque about the x-axis and y-axis corresponding to different error states

where $v(s,t)$ denotes the covariance function. In the above equations, the principal functions $\xi_{jk}(s), k = 1, \cdots, p$ are sorted by the eigen value ρ_{jk}. The feature vector can be obtained from the principal functions as scores of the functional principal component analysis. We set $p = 2$ in this research.

3.3 Classifier Construction

In this subsection, we consider constructing the classifier of error states. Fig. 3 shows an example of the torque data during snap assembly simulations with assuming the offsets $\Delta y = \pm 5$[mm]. Among two results of assembly simulations with different offsets in the y direction, the waveform data of the torque about the x-axis are apparently different while the waveform data of the torque about the y-axis are almost same. We can find that, depending on the error state, the component of force/torque affecting the error state is different. Hence, to classify the error states, it's important to find the force/torque component affecting the error states. Our method firstly prepares a set of force/torque data labeled as success or one of the error states. By using the k-means clustering, the proposed method constructs a decision tree classifying given force/torque data as the success or one of the error states. The classifier is constructed by using Algorithm 1.

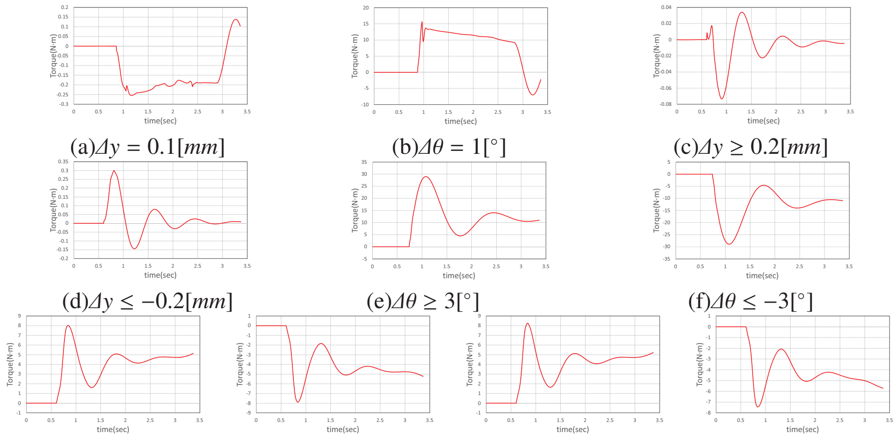
We now briefly explain Algorithm 1. In **Step 1**, we consider classifying the snap assembly as the success/failure cases. Assuming $C = 2$, we apply the k-means clustering for each force/torque component of fPCA score sets. We then find a force/torque component well classifying the success/failure cases. In **Step 2**, we try to find both the number of clusters and a force/torque component well discriminating error states. We terminate the algorithm if all the error states are completely discriminated(Algorithm 1, line19).

Algorithm 1: Classifier construction based on k-means clustering

```
Data: fPCA scores and success/failure states
      Success pattern index: i ← 0
      Failure pattern index: i ← 1, ··· , F
      Force/torque component: j ← 1, ··· , 6
      Index of waveform data: k ← 1, ··· , N(i)
      Waveform data: WD(i, j, k)
      Number of clusters used in k-means: C
Result: Construction of a decision tree for classification
begin
    Step1 classify success/failure cases
    id ← 0
    node(id).pattern_id ← [0, 1, ··· , F]
    for j ← 1, ··· , 6 do
        cluster(j, 2) ← k-means(fPCAScore(WD(i, j, k)), C ← 2),i ← 0, ··· , F
        accuracy(j, 2) ← calcAccuracy(cluster(j, 2))
    node(id).component ← argmax(accuracy(j, 2))
    node(id).cluster ← 2
    node(id).children ← [id + 1, id + 2]
    node(id + 1).pattern_id ← [0]
    node(id + 2).pattern_id ← [1, ··· , F]
    Step2 classify failure patterns
    id ← 2
    max_id ← 2
    while 1 do
        if size(node(id).pattern_id == 1) then
            if id == max_id then
                return node
            id ← id + 1
            continue
        for j ← 1, ··· , 6 do
            for C ← 2, ··· , 5 do
                cluster(j, C) ←
                k-means(fPCAScore(WD(i, j, k)), C),i ← node(id).pattern_id
                accuracy(j, C) ← calcAccuracy(cluster(j, C))
        node(id).component ← argmax_{1≦j≦6}(accuracy(j, C))
        node(id).cluster ← argmax_{2≦C≦5}(accuracy(j, C))
        node(id).children ← [max_id + 1, ··· , max_id + node(id).cluster]
        for i ← 1, ··· , node(id).cluster do
            node(max_id + i).pattern_id ← i − thClassified_ids
        max_id ← max_id + node(id).cluster
        id ← id + 1
end
```

3.4 Identification of error states during an assembly tasks

In this subsection, we consider identifying the error state before the error actually happens. We identify the error state by using time trajectories of force/torque starting from



(a)$\Delta y = 0.1[mm]$ (b)$\Delta\theta = 1[°]$ (c)$\Delta y \geq 0.2[mm]$

(d)$\Delta y \leq -0.2[mm]$ (e)$\Delta\theta \geq 3[°]$ (f)$\Delta\theta \leq -3[°]$

(g)$\Delta y = 0.3[mm]$, $\Delta\theta = 0\sim3[°]$ (h)$\Delta y = 0.3[mm]$, $\Delta\theta = -3\sim0[°]$ (i)$\Delta y = -0.3[mm]$, $\Delta\theta = 0\sim3[°]$ (j)$\Delta y = -0.3[mm]$, $\Delta\theta = -3\sim0[°]$

Fig. 4. Waveforms of torque about the *z*-axis

the initial time t_0 to the ending at the time $t_{classify}$ at the middle of an assembly task. Identification can be done by using the decision tree obtained by using Algorithm 1. We can expect that, if the waveform data ends close to the final state, identification is expected to be accurate. On the other hand, if the waveform data ends far from the final state, identification can be performed before the error actually happens with high possibility.

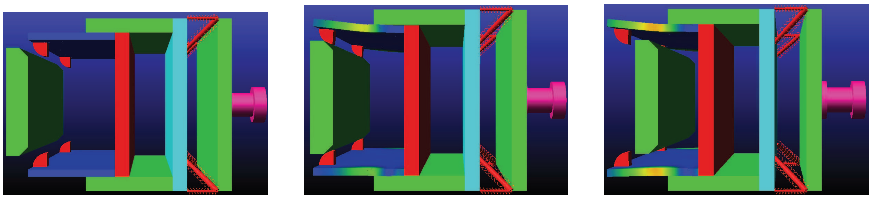
4 Simulation Results

We collected the training data of the simulated snap assembly. Given various offset patterns (45 patterns) including the translation in y direction and the rotation about z-axis, we simulated snap assemblies and observed the error states. When the offset is one of the following eight patterns, we confirmed that the assembly task failed: (1)$\Delta y \geq 0.2[mm]$, (2)$\Delta y \leq -0.2[mm]$, (3)$\Delta\theta \geq 3[°]$, (4)$\Delta\theta \leq 3[°]$, (5)$\Delta y = 3[mm]$, $\Delta\theta = 0\sim3[°]$, (6)$\Delta y = 3[mm]$, $\Delta\theta = -3\sim0[°]$, (7)$\Delta y = -3[mm]$, $\Delta\theta = 0\sim3[°]$ and (8)$\Delta y = -3[mm]$, $\Delta\theta = -3\sim0[°]$.

By using these training data, we constructed the classifier and confirm how accurately the classifier can predict the error states.

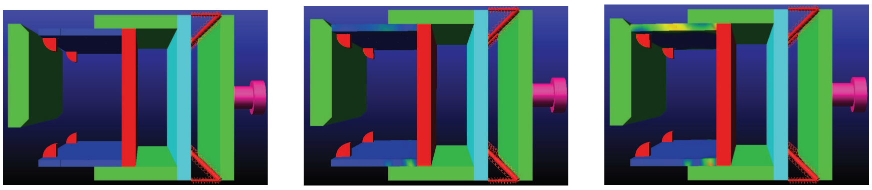
Fig. 4 shows the waveforms of the simulated torque component about the z-axis. Figs. 5 and 6 show the snapshots of assembly tasks for a successful case ($\Delta y = 0.1[mm]$) and a failure case ($\Delta y = 0.8[mm]$), respectively.

We can see from Fig. 4 that different offset patterns have different waveforms of force/torque. Hence, we can expect to predict the error states by using our proposed method.



(a) (b) (c)

Fig. 5. Successful case of assembly when $\Delta y = 0.1[mm]$

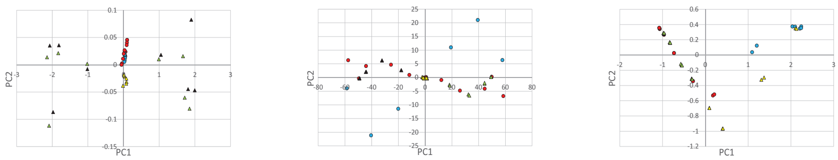


(a) (b) (c)

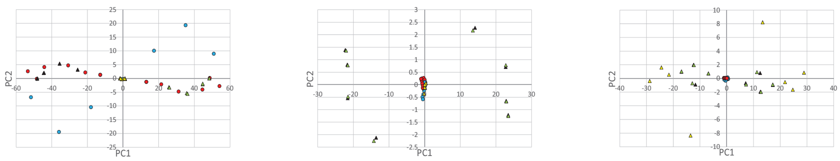
Fig. 6. Failure case of assembly when $\Delta y = 0.8[mm]$

4.1 Feature Quantity Extraction

We show scores up to the second principal component score in Fig. 7 since more than 90 % of contribution ratio is realized by using up to the second principal component function.



(a)Force in x direction (b)Force in y direction (c)Force in z direction



(d)Torque about x-axis (e)Torque about y-axis (f)Torque about z-axis

Fig. 7. Plots of principal component scores where the red ○, the blue ○, the yellow △, the green △ and the black △ denote $\Delta y = -0.8{\sim}-0.2[mm]$, $0.2{\sim}0.8[mm]$, $\Delta y = -0.15{\sim}0.15[mm]$, $\Delta\theta = -4{\sim}4[°]$, $\Delta y = 0.3[mm]$, $\Delta\theta = -4{\sim}4[°]$ and $\Delta y = -0.3[mm]$, $\Delta\theta = -4{\sim}4[°]$, respectively.

4.2 Classifier

We classified the error states by applying the k-means clustering based on Algorithm 1. Decision tree obtained from Algorithm 1 is shown in Fig. 8.

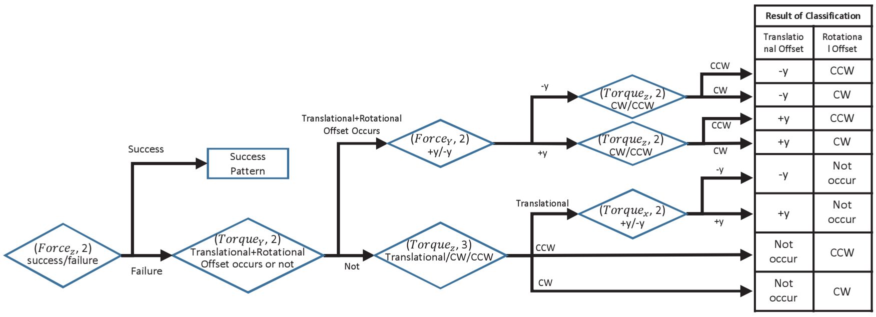
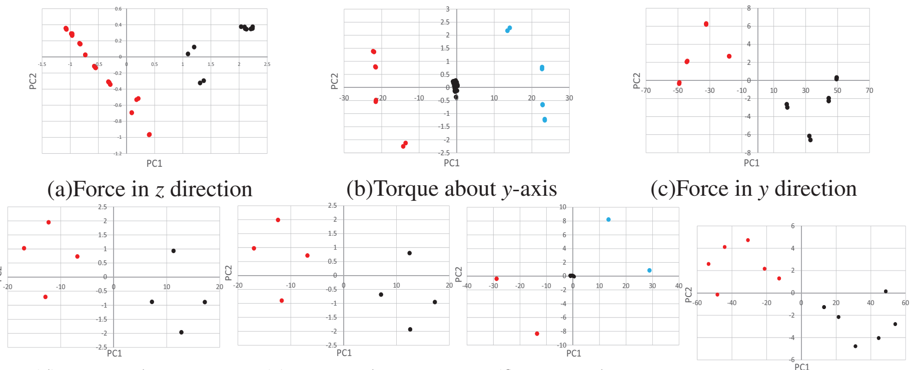


Fig. 8. Decision tree for classifying error states



(a)Force in z direction (b)Torque about y-axis (c)Force in y direction

(d)Torque about z-axis(1) (e)Torque about z-axis(2) (f)Torque about z-axis(3) (g)Torque about x-axis

Fig. 9. Result of clustering

4.3 Identification during assembly task

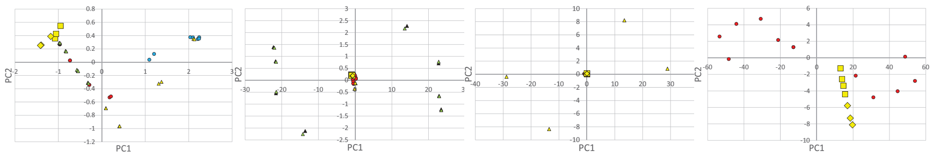
We identify error states by using the obtained decision tree. We perform the simulated snap assembly assuming following offset patterns: (1)$\Delta y = 0.075[mm]$, (2)$\Delta y = 0.2[mm]$, (3)$\Delta y = -0.85[mm]$, (4)$\Delta\theta = 0.5[°]$, (5)$\Delta\theta = 5[°]$, (6)$\Delta\theta = -3[°]$ and (7)$\Delta y = 0.2[mm]$ and $\Delta\theta = -3[°]$. We labeled the force/torque data of assemblies as '$Success$' and $Fail_1$: $\Delta y > 0$, $Fail_2$: $\Delta y < 0$, $Fail_3$: $\Delta\theta > 0$, (6)$Fail_4$: $\Delta\theta < 0$ and $Fail_5$: $\Delta y > 0,\ \ \Delta\theta < 0$.

By changing the time $t_{classify}$ from $t_{classify} = 3.375[s]$ to $t_{classify} = 2[s]$, we extracted feature quantity vectors. And we identify the error states by using the decision tree shown in Fig. 8. Results of identification are shown in Figs. 10, $\cdots$, 12 where yellow *square* and black *rhombus* denote the feature quantity vectors with changing from $t_{classify} = 3.375[s]$ to $2.5[s]$ and that changin from $t_{classify} = 2.5[s]$ to $2[s]$, respectively.

In Fig. 10, every feature quantity vector is always included in the cluster labeled as $Success$ after $t_{classify} = 2[s]$. In Fig. 11, every feature quantity vector is always included in the same cluster of failure states after $t_{classify} = 2[s]$ and we got same results in the cases of data(5) and (7). However, in Fig. 12, the classifier could identify the error state correctly from $t_{classify} = 3.375[s]$ to $2.5[s]$, but it could not identify correctly from $t_{classify} = 2.5[s]$ to $2[s]$ and we got same result in the case of data(6). It implies that we have to set $t_{classify} = 2.5[s]$ to correctly identify the success and error states.

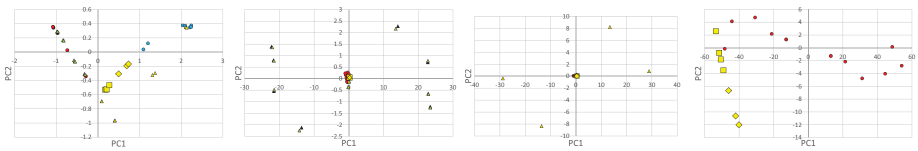
(a) Force in z direction of data(1) (b) Force in z direction of data(4)

Fig. 10. Results of identification at the middle of assembly tasks about force/torque data(1) and (4)



(a)Force in z direction (b) Torque about y-axis (c) Torque about z-axis (d) Torque about x-axis

Fig. 11. Result of Identification at the middle of assembly task about force/torque data(3)



(a) Force in z direction (b) Torque about y-axis (c) Torque about z-axis (d) Torque about x-axis

Fig. 12. Result of misidentification at the middle of assembly task about force/torque data(2)

5 Colclusions

In this paper, we proposed an error identification method including several failure patterns during robotic snap assembly. In the proposed method, we gather waveform data from simulated snap assemblies at various offsets and extract functional principal component scores as feature quantity vectors by using fPCA. Then, we classify success/failure patterns by using phased k-means clustering. And we predict the failure of assembly and detail failure patterns by comparing learned clusters and feature quantity vectors extracted from waveforms obtained from the beginning of the assembly to the middle of it. Numerical simulation showed that every given unknown success/failure pattern waveform data could be identified correctly during assemblies. We consider the error recovery motion after identifying success/failure patterns as a future research.

References

1. A.Rodriguez，D.Bourne，M.Mason，G.F.Rossano，J. Wang. “Failure Detection in Assembly: Force Signature Analysis ”. IEEE International Conference on Automation Science and Engineering, 2010.
2. A.Scolt，M.Linderoth，A.Robertsson，R.Johansson. “Detection of contact force transients in robotic assembly ”. IEEE International Conference on Robotics and Automation (ICRA)，2015.
3. J.Rojas,K.Harada,H.Onda,N.Yamanobe,E.Yoshida,K.Nagata. “Early failure characterization of cantilever snap assemblies using the PA-RCBHT ”. IEEE International Conference on Robotics and Automation (ICRA)，2014.
4. Ge Ming，Xu Yangsheng，Du Ruxu. “ An Intelligent Online Monitoring and Diagnostic System for Manufacturing Automation ”. IEEE Transactions on Automation Science and Engineering，2008.
5. L.Rusli，A.Luscher，C.Sommerich. “ Force and tactile feedback in preloaded cantilever snap-fits under manual assembly ”. International Journal of Industrial Ergonomics，2010.
6. S.Gen，R.W.M.Jr.，G.A.Gabriele. “A hierarchical classification scheme to define and order the design space for integral snap-fit assembly ”. Research in Engineering Design，1998.
7. J.Ji，K.-M.Lee，S.Zhang. “Cantilever Snap-Fit Performance Analysis for Haptic Evaluation ”. Journal of Mechanical Design，2011.
8. J.Rojas,K.Harada,H.Onda,N.Yamanobe,E.Yoshida,K.Nagata，Y.Kawai. “A Relative-Change-Based Hierarchical Taxonomy for Cantilever-Snap Assembly Verification.”. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.
9. E.D.Lello，M.Klotzbucher，T.D.LAet，H.Bruyninckx . “Bayesian Time-Series Models for Continuous Fault Detection andRecognition in Industrial Robotic Tasks ”. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.
10. W.Luo，J.Rojas，T.Guan，K.Harada，K.Nagata. “ Cantilever snap assemblies failure detection using SVMs and the RCBHT ”. IEEE International Conference on Mechatronics and Automation，2014.
11. Adams’MSC Software’ http://www.mscsoftware.com/product/adams
12. Y.Yamanishi, Y.Tanaka, “ Principal Component Analysis for Functional Data ” Journal of Faculty of Environmental Science and Technology, Vol.6, 25-34, 2000
13. Besse, P. and Ramsay, J. O. ”Principal components analysis of sampled functions”, Psychometrika, 51, 285-311, 1986
14. Ramsay, Y. O. and Silverman, B.W. Functional Data Analysis, Springer, 1997