中山大學
SUN YAT-SEN UNIVERSITY

# 本 科 生 毕 业 论 文（设计）


题　　目： 基于力与扭力信号对错误组装案例进行识别

院　　系： 软件学院

专　　业： 软件工程

学生姓名： 罗伟强

学　　号： 10389321

指导教师： Dr. Juan Rojas

（职 称）


二〇一四年三月

# 基于力与扭力信号
# 对错误组装案例进行识别


## 摘要

错误检测在现代工业进程以及机器人服务，特别是无规则环境的作业中，担任越来越重要的角色。我们主要研究了悬臂卡扣组装中的错误检测技术。它在工业应用以及个人机器人中有着重要的地位。

我们希望根据卡扣组装过程的力信号特征，用小集合的特征向量来抽象组装过程，并且据此来训练支持向量机分类法，从而在组装任务的不同阶段精确地检测组装过程中发生的错误。在我们的实验中，我们通过抽象行为特征训练了一个线性支持向量机，从而进行卡扣组装的错误检测。这个方法在早/晚期的错误检测中非常有效。在早期错误检测中，抽象程度较低的行为特征集相对于高度抽象集表现得更好，这是因为在其局部时间内颗粒度更加精细的原因。在晚期错误检测中，高度抽象集表现得更好，因为对于全局的组装动作，高度抽象集更好地表示了组装动作。



**关键词：**力信号；支持向量机；基于梯度变化的分类法

# Early Failure Assembly CasesDetection ased on Force/Torque Signal


## Abstract

Failure detection plays an increasingly important role in industrial processes and robots that serve in unstructured environments. This work studies failure detection on cantilever snap assemblies, which are critical to industrial use and growing in importance for personal use.

Our aim is to study whether an SVM can use a small set of features abstracted as behavior representations from the assembly force signature to accurately detect failure at different stages of the task. In this work, a linear SVM was embedded with abstract behavioral features was used to classify failure detection in cantilever snap assembly problems. The approach was useful in detecting failure both during early and late stages of the task. For early stages, low-abstraction behaviors sets performed better due to their granularity and local temporal nature. For late stage analysis, high-abstraction behaviors performed better as they capture representative and global behaviors better.



**Keywords:** Force/Toruqe;Support Vector Machine;Relative-Change-Based-Hierachical-Taxonomy-System

# 目录

# 第一章　Introduction

Uncertainty us the primary factor in robot manipulation failure. In assembly tasks, failure can lead to severe damage to both robot and tool parts. Detecting failures early in an assembly is important to avoid part's damage and increase performance. Recent work detecting assembly failures mostly focus on part assembly 1, tool breakage 2 3, and threaded fastener assembly 4. No work yet has focused on snap assemblies. In our work, we have used SVMs to monitor and identify failure cases early and at later stages of the assembly. This work builds on top of our previous snap assembly 5 and incorporates an SVM monitor to the state estimation capbilities of the framework.

Prior to this work, failure identification on snap assemblies was executed using a probabilistic approach named Probabilistic Relative Change-Based Hierarchical Taxonomy (pRCBHT) 6. The pRCBHT adds a Bayesian filter monitor on top of the RCBHT. The latter is a taxonomy composed of increasingly abstract layers that encode robot states in intuitive ways. The bayesian filter, examines the presence of encoded behavior as well as the duration of these to produce a belief about the state. In this paper, SVMs are used instead of Bayesian filtering since SVMs provide a simpler and more intuitive approach to the classification problem. In our cases, SVMs were used in conjunction with encoded labels from three different layers in the taxonomy. Levels with higher abstraction provide high-level (global spatio-temporal) information about the task, while lower levels provide more raw (local spatio-temporal) information about the task.

SVMs are currently one of the most widely used classification techniques. There is another work using Support Vector Machine for idenitying assembly failure 7, using the signals with certain time stamp to construct feature vectors. Rather than this approach, RCBHT encode the origin signal data into higher level of abstraction. Our approach takes account of labels by counting the labels and constuct a much smaller feature vector than those in 7. The accuracy of this classification approach indicates our work has comparable result with 7.

There are three layers of RCBHT used in our approach, Primitive layer, Motion Composite Layer, Low-level Layer, and they are increasingly abstracted. With labels of Primitive layer, early failure detection is done in our approach. With labels of Low-level Behavior layer and Motion Composite layer, (?)late stage failure detecting is done. (? Why I should talk about two stages with all approach?) The accuracy of early stage failure detection reached 93.67% and late stage failure detection reached 99.59%.

Other works including 8 and 9 analyze signals through the use of Hidden Markov Models (HMMs). HMMs have aided in creating generalizable methods for classification of manipulation and assembly strategies. Their methods are more general, but RCBHT was specially developed for the cantilever-snap assembly and increasingly abstracted layers will be more data-rich.

Our SVM classification yielded high accuracy results. However, there are also important limits. The first is that the RCBHT appraoch does not easily generalize for other assembly or manipulation problems. The second is that the failure assembly is assumed to happen from the very begining. If something happened in between the process, we can't detect it as soon but until a state finished.

In section 2, we will detail our previous work including assembly strategy and RCBHT taxonomy. In section 3, we will talk about using SVM with RCBHT to classify the failure cases. In section 4, we will give a brief review about SVM. In section 5, we will depict our experiment result and have a brief discuss about the result and limitation of the approach. In the last section, we will conclude this paper and talk about our future work.

# 第二章　Overview

## 2.1　Experimental Setup

In our work, the OpenHRP 3.0 simulator was used to produce a snap assembly process using HIRO, a simulated 6 DoF dual-arm anthropomorph robot. CAD derived male and female camera molds with 4 cantilever snaps were used where the male part was rigidly mounted on the robot's wrist while the female part with 4 snaps was fixed on the ground. The snap part of this task is cantilever snap. The cantilever snap is as the following picture.

For controlling, Side Approach strategy 10 is used to assemble the snap parts and Relative Change Based Hierarchical Taxonomy System (RCBHT) is used to sample the force/torque signals of the whole process. Details of this work can be seem in 11.

**图 2.1: Cantilever Snap**

## 2.2 Control Strategy

The control strategy is fixed. The assembly process contains four state: Approach state, Rotation state, Insertion state, Mating state. In approach state, the upper part approaches to the lower part. After approach state, the upper part rotates until the other two sides of the upper part and the lower part contact. Then force increase to insert the upper part into the lower part. The assembly is finished in Mating state.

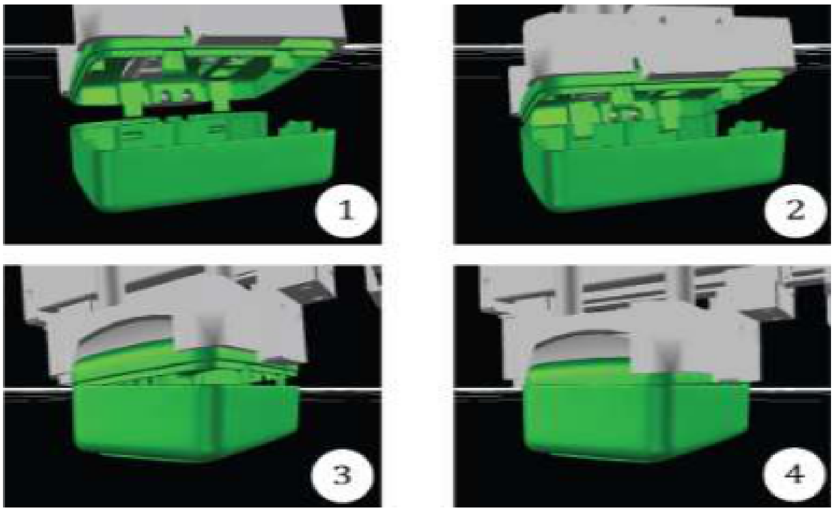


**图 2.2: Four control states**

## 2.3 RCBHT Introduction

The relative change-based hierarchical taxonomy is the state estimation technique that represents the states by hierarchically abstracting snap assembly for force/torque data in increasingly intuitive ways. Five increasingly abstracted layers were used to encode the relative change ub with the force signatures. The taxonomy is consisted by five increasing abstracted layers, including Primitive layer, Motion Composites layer, Low-level Behavior layer, High-level Behavior and Snap Verification Layer. The layers is shown in

In our work, we used three lower layers instead the five layers. The details

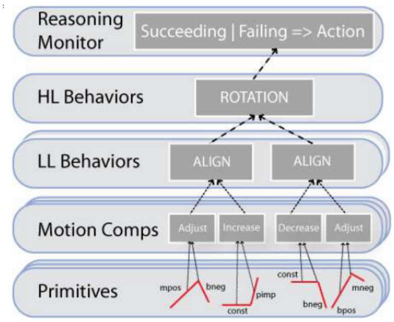


图 2.3: **Five Layers of RCBHT**

about RCBHT can be seen in 12. In this section, we will only describe the three lower layers.

### 2.3.1 Primitive layer

In the primitive layer, each signal is partitioned into linear segments with linear regression with a coefficient. Nine ranges of value of coefficient were labeled by nine different labels. The labels indicate nine ranges of gradients. The upper layers are based on Primitive layer labels.

### 2.3.2 Motion Composites layer

Motion Composites are consisted of multi-labels of Primitive layer. Two or above similar labels of Primitive layer produced a Motion Composite label. This more abstracted labels attenuate the noise and condense the information.

### 2.3.3 Low-level Behavior leyer

Being more abstracted, labels of Motion Composite Layer form another layer, Low-level Behavior layer. In this layer, information became more purer and represent the process more intuitively.

Details about RCBHT can be seen in appendix.

# 第三章 Classification with SVMs and the RCBHT

In this section we will introduce a binary classifier using Support Vector Machines and feature inputs from RCBHT labels. The first three layers of the RCBHT taxonomy are of relevant importance to the classification problem. Labels from these three layers will be used to construct fixed-length feature vectors. The premise of constructing a feature vector out of RCBHT labels is that success cases have similar patterns in all six axis and similar patterns indicate the similar gradient of each separation of the signal.

The left four signal figures are success cases and right four are failure cases before Rotation state. The success cases are in similar patterns while the failure cases are different from the success cases diversely. With diverse gradients of different patterns, RCBHT can be used in our work.

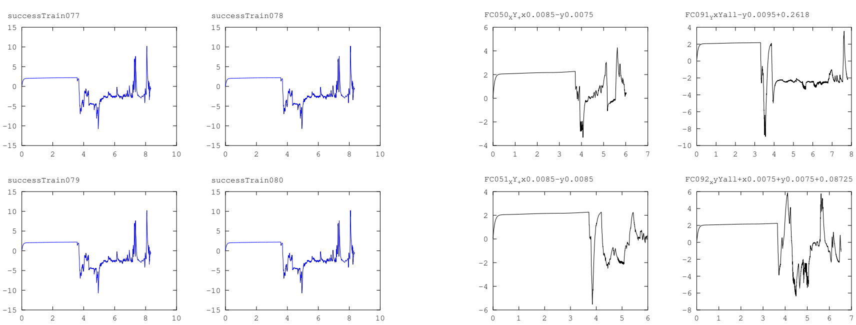


**图 3.1: Success Fx signal and Failure Fx signal**

Thus, feature vectors were created with RCBHT. Our feature vectors are composed of gradient labels at three different levels of abstraction that characterize an assembly task. The taxonomy RCBHT is based on relative change. THe reason for choosing RCBHT is that this taxonomy is specially developed for this task. Also, the feature vector constructed with RCBHT has much fewer dimension, at most 9, than typical feature, which may use 2000 sample points for 10-second assembly with sampling frequency of 200Hz.

This approach with SVM all based on one assumption when the classifier was trained, it could only be used for assembly with same configuration. Different geometric snap, changing assembly strategy, different sensor position etc. would change the signal patterns which would be misidentified. Any change of configuration required retraining under that configuration.

Then comes to our approach setup. In the first subsection following, we will describe our approach to construct feature in detail. The second subsection will talk about our work with SVM briefly.

## 3.1 Feature Vector Constructing

With the RCBHT, we sample the whole assembly using Primitive labels, Motion Composites labels, Low-level behavior labels. Each axis, $F_i$, contains n entries, each represent a label type $e_j$. Such that $F_i = e_1, e_2, \ldots, e_n$. The whole vector is consist of six axis, $F_x, F_y, F_z, M_x, M_y, M_z$. 9 labels independently of each axis is consisted in primitive layer, 7 of MC and 7 of LLB. The feature vector will be conducted containing 6 axis, as $[F_x, F_y, F_z, M_x, M_y, M_z]$. Take the vector representing Primitive layer if in $F_x$, bpos occurs twice, mpos occurs once, and the other axis do not have labels (which is an impossible case but just for better understanding), the final vector will be constructed as $[2, 1, \underbrace{0, 0, \ldots, 0}_{\text{52 zeros}}]$

Detailed representation for Primitive layer, Motion Composites layer, Low-level behavior layer can be seem in 3.1.

表 3.1: **Feature Vector Representation**

| Vector Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Primitive Layer | bpos | mpos | spos | bneg | mneg | sneg | cons | pimp | nimp |
| MC Layer | a | i | d | k | pc | nc | c | | |
| LLB Layer | FX | CT | PS | PL | AL | SH | U | N | |

## 3.2 Classify Feature Vector

After constructing feature vectors, classifer should be trained between the success and failure cases. Among different techniques for supervised classification, linear Support Vector Machine is chosen. In this section, we will briefly talk about classifier implementation and review Support Vector Machine and kernel function.

## 3.3 Support Vector Machine Review

The Support Vector Machine methos is to find a hyperplane that separate the cases with different labels. Nice prediction are made when all cases, represented by points, are far away from the hyperplane, which means all the hyphothesis are creditable. The hyperplane can be represented as:$\omega^T x + b = 0$. Here $\omega^T$ is the multiply factor of the hyperplane when b means the bias from the zero point. Each point, the deviation can be represented as:

$$\hat{\gamma}^{(i)} = y^{(i)}(\omega^{(i)}x + b) \tag{3.1}$$

Here $(y^{(i)}, x^{(i)}$ is a single case when $y^{(i)}$ is the label whether this case was success or not represented in 1, -1 respectively and $x^{(i)}$ is the vector input for training and testing. $\hat{\gamma}^{(i)}$ is the functional margin of this case. To have a nice hyperplane, we need to let as many as possible points to get as far away as well from the hyperplane.

That's to say,

$$\begin{aligned} max \quad & \gamma \\ s.t. \quad & \gamma = \min_{i=1,\dots,m} \hat{\gamma} \end{aligned}$$

Here $\gamma$ is the geometric margin of points from hyperplane. This equation shows that the nature of SVM is to find a hyperplane to maximize $\gamma$, which is the least functional margin of $\hat{\gamma}$.

## 3.4 Classifier Implementation

In our work, libsvm 13 is used with Gaussian Kernel 14, which is recommended as the first try kernel. After labeling each feature vector with 1 or -1 to present success or failure, the cases with labels were then trained with Support Vector Machine and get a hyperplane. The two kind of cases were separated in the two sides. With this hyperplane, new cases can be predicted.

# 第四章　Experiment Result

The SVM classifier is trained and tested with a total of 192 assemblies. Among those assemblies, 150 are failure cases with deviation of x, y, and $\phi$ direction, and 42 are success cases. Two scenarios are considered: (i) early failure detection, considering the labels of Approach state and (ii) late failure detection, consisting of labels in all four states of the Assembly task.

## 4.1 Training and Testing Methodology

Two methods are conducted in training and testing. The 192 assemblies are separated into two groups, for testing and training. (i) For testing, we randomly selected 96 samples (75 failures and 21 success). (ii) For training, we used the remaining 96 samples (Also 75 failures and 21 success). For the first method, named preset training (PT), we started our training from 5 cases (1 success and 4 failures). Then we append samples into our training group. For every 4 failure cases we include 1 success case. For the second method, named random training (RT), we randomly selected the training samples, increasing from 5 to 96 samples but keep the ratio of success samples and failure samples of 1/4.

We ran the two methods for 100 times respectively. In the following subsections we will describe our findings of the two methods.

## 4.2 Early Failure Detection

For early failure detection we construct our feature vectors consisting of those labels that show only during the Approach state for all six force/torque axes. With feature vector consisting P labels only, the classifier reacged an average asymptotic maximum value of 93.72% and a minimum of 89.6% for method using the RT method. The values for the PT method are 93.67% and 89.6% respectively.

The figure left is with the PT method while the right one is with the RT method.

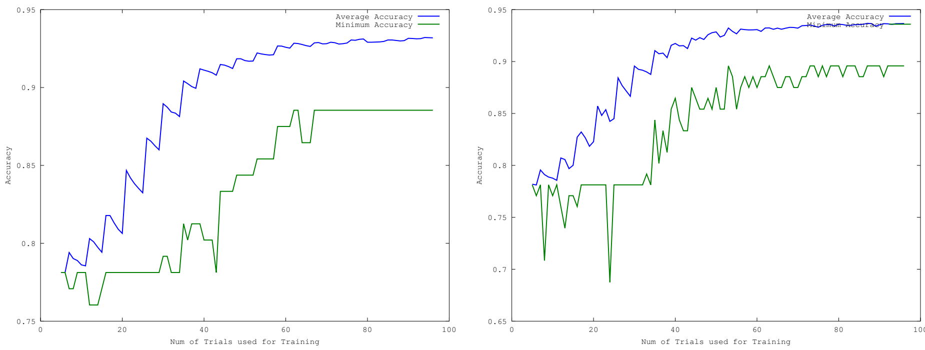


图 4.1: **Classifier with P labels**

## 4.3 Late Failure Detection

For the late failure detection we construct our input feature vector consisting of those labels that show throughout all four states of the task (Approach-Mating) for all six force/torque axes. With feature vector consisting MC and LLB, we conducted the same experiment as it of early failure detection. For the LLB and MC layers, with the PT method, the classifier had an average asymptotic maximum value of 99.59% and 99.25% respectively and a minimum of 98.9% and 93.8% respectively. The MC classifier reached asymptotic value after about 70 trails while the LLB classifier did so after approximately 22 trials.

The figure left is with the PT method while the right one is with the RT method.

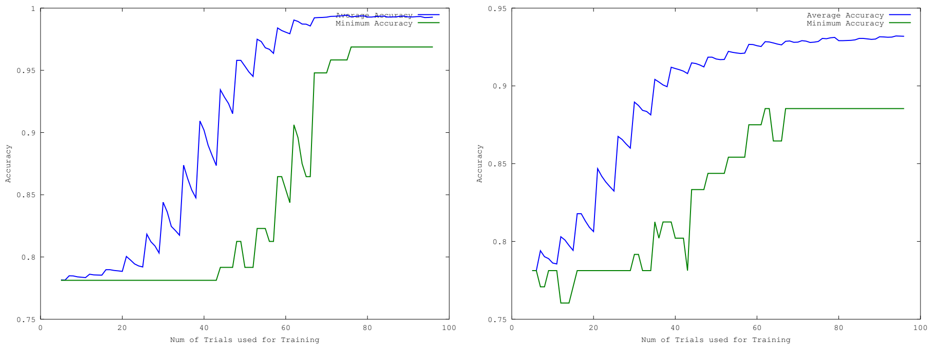


**图 4.2: Classifier with MC and LLB with the PT method.**

With RT method, the result is very much the same as PT method.

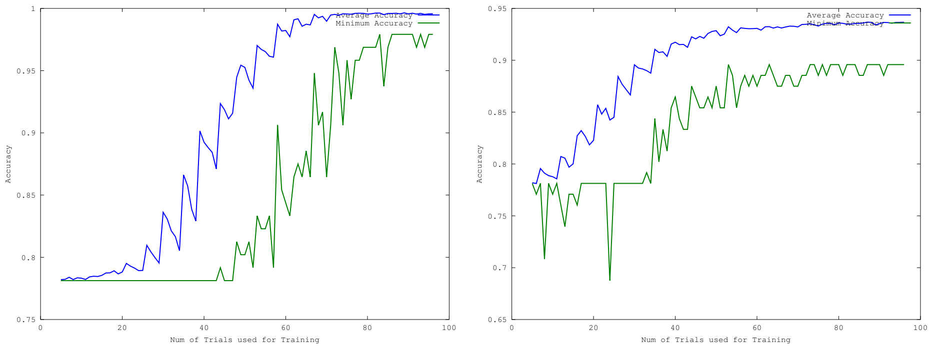


**图 4.3: Classifier with MC and LLB labels with RT method**

Both the LLB and MC classifiers can separate classes with very high accuracy. This result indicates that there must be inherent difference between success and failure classes in the atemporal feature vector. We also noted that when only early failure detection is tested, the MC and LLB classifiers have too few data to identify failure. In some cases, in some force axes, there is only one LLB in the Approach state (and whose duration lasted the entire state). On the other hand, for the P classier and late failure detection, P labels contain too much noise at a very granular level. When considering only the Approach state, the noise accumulation is not that significant, however, after an entire assembly task has been conducted, the noise disturbance is too significant for proper classification to take place.

# 第五章　Discussion and Future Work

## 5.1　Discussion

Our approach successfully used a small feature set of behavior representations from the RCBHT. We noted that early failure detection is possible with low abstraction levels but not otherwise. Similarly, late failure detection was possible with higher-level of abstraction labels. With LLB label set the classifier reached 99.59% with around 20 trials, which is comparable to 7 but with a much smaller feature vector size.

However, the limitation of this approach is fatal. When something wrong happens in the robot from the begining, we can surely detect the failure. But if the machine hand got into problem during the assembly, we could not realized at the very moment. In the lateral situation the system is lack of prevention to preserve the machine which may cause severe problems.

## 5.2　Future Work

We should consider these three improvements we could make: (1)Classify the failure cases into subsets and (2) be able to detect failure as soon as it happens.

Classifying the failure cases with the force/torque signal into subsets, we would have the posterior of the diviation distribution and the corresponding diviation of specific signal. Failure cases with only one or two axis of diviation can be success classified currently but we could not effectively implement it with three axis of diviation.

Being able to detect failure at the moment it got away from the right path is necessary because these kinds of robotic problems would cause problems. The damage would be acute at the first wrong behavior. As our limitation, the hypothesis could be made after this behavior occurs. The machine arm had been some damage when it's too late.

# 参考文献

[1] D. M. Fullmer. Parts assembly using signature analysis. Google Patents, 1989-08 8.

[2] S. Cho *et al.* Tool breakage detection using support vector machine learning in a milling process. International Journal of Machine Tools and Manufacture, 2005, 45(3): 241–249.

[3] Y.-W. Hsueh and C.-Y. Yang. Prediction of tool breakage in face milling using support vector machine. The International Journal of Advanced Manufacturing Technology, 2008, 37(9-10): 872–880.

[4] K. Althoefer *et al.* Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2008, 222(6): 1081–1095.

[5] J. Rojas and R. P. II. Analysis of Autonomous Cooperative Assembly Using Coordination Schemes By Heterogeneous Robots Using A Control Basis Approach. J. of Autonomous Robots, 2012, 32(2): 97–172.

[6] J. Rojas *et al.* Probabilistic State Verification for Snap Assemblies using the Relative-Change-Based Hierarchical Taxonomy (in-print). IEEE-RAS Intl. Conf. on Humanoid Robots, 2012.

[7] A. Rodriguez *et al.* Failure detection in assembly: Force signature analysis. Automation Science and Engineering (CASE), 2010 IEEE Conference on, 2010: 210–215.

[8] A. Rodriguez *et al.* Abort and retry in grasping. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011: 1804–1810.

[9] E. Di Lello *et al.* Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, 2013: 5827–5833.

[10] J. Rojas *et al.* Cantilever Snap Assembly Automation using a Constraint-Based Pivot Approach. IEEE Intl. Conf. on Mechatr. & Automation, 2012.

[11] J. Rojas *et al.* Towards snap sensing. International Journal of Mechatronics and Automation, 2013, 3(2): 69–93.

[12] J. Rojas *et al.* A Relative-Change-Based Hierarchical Taxonomy for Cantilever-Snap Assembly Verification. IEEE Intl Conf. on Robots and Systems, 2012.

[13] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, 2: 27:1–27:27.

[14] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. Neural computation, 2003, 15(7): 1667–1689.

# 附录 A The RCBHT

The hierarchical taxonomy's goal was to connect human apropos actions like: "approaching", "rotating", "aligning", "snapped", and "mated" with LLB's in a context-sensitive manner. One of the main challenges encountered in interpreting force signals is their inherent noise and spatio-temporal complexity. However, the force signals do inherently possess characteristics that describe the task at hand. The authors hypothesized that such characteristics could be extracted by looking at how temporal relative changes were associated to each other and contextualized by the state in which they occur. In so doing, intuitive behavior sequence's can be extracted and their outcome examined. This level of discrimination is significant as it can be expanded to a real-time implementation and allow to reason about the state to perform corrective motions if necessary. To bootstrap the approach, we partitioned the data into linear segments that approximate the data and classify the gradients according to magnitude per a small set of criteria. The next layer of abstraction examines at ordered-pair primitive sequences, and according to the gradient patterns presented and a small set of classification criteria, they are categorized into one of several types of motion compositions. The third layer abstracts sequences of motion compositions to identify LLB's, while the fourth layer looks at what LLB's are present in which states to determine if desired high-level behaviors are present. The final layer outputs the verification process results' according to whether or not the desired sequence of HLB's is present or not. A visualization of the hierarchical taxonomy can be seen in Fig. A.1.

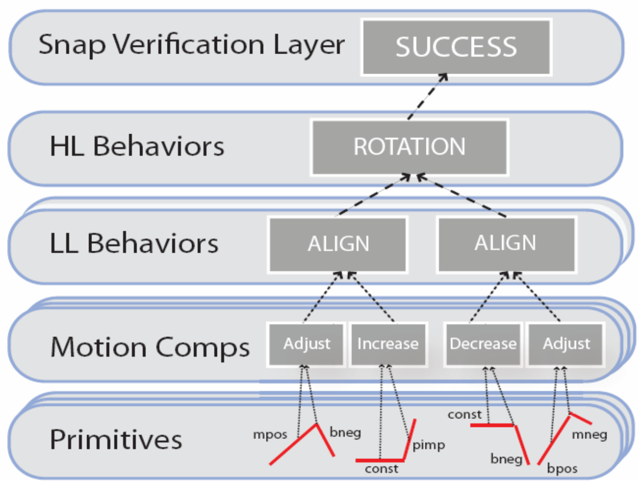


图 A.1: **The Relative Change-Based Hierarchical Taxonomy (RCBHT) for Cantilever-Snap Assembly Verification.**

## A.1 Primitive Layer

The primitives layer requires that each signal is partitioned into linear segments of data that closely approximate the original signal. Linear regression in concert with a correlation measure (the determination coefficient $R^2$) is used to partition the data whenever a minimum correlation threshold is crossed. If the determination coefficient drops under a given threshold the linear fit is partitioned and a new regression is started. The $R^2$ coefficient is a correlation measure that studies the ratio of the sum of the squares of the residual errors between the original data $y$ and the fit data $\hat{y}$ to the sum of the variance ${\sigma_y}^2$ as shown in Eqtn. A.1.

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{{\sigma_y}^2} \tag{A.1}$$

The threshold used to partition the data was set at 0.70, such that if the correlation dropped to under 70%, a linear segment or "partition" would be generated, and a new one would start at the next data point. The data was traversed by a window equal to five data points (the data was sampled at a frequency of 1kHz by the simulation). The threshold values and the window length were empirically selected to partition the data sufficiently to capture relevant changes in the signals. Each partition was accompanied by a data structure with seven types of information about itself: the average value across data points, the maximum value, the minimum value, the start time, the end time, the gradient value, and a gradient label. With respect to the latter, nine gradient labels (positive impulse, ‘pimp’; big, medium, and small positive gradients, ‘b/m/spos’; constant gradients, ‘const’; and their negative equivalents, ‘nimp’, ‘b/m/s/neg’) were assigned according to ranges summarized in Fig. A.2. The classification first attempts to separate instances of data in which

| | |
|---|---|
| pimp | $m \geq 70$ |
| bpos | $46 \leq m < 70$ |
| mpos | $23 \leq m < 46$ |
| spos | $1.0 \leq m$ 23 |
| const | $-1.0 < m < 1.0$ |
| sneg | $-23 < m \leq -1.0$ |
| mneg | $-46 < m \leq -23$ |
| bneg | $-70 < m \leq -46$ |
| nimp | $m \leq -70$ |

**图 A.2: Gradient values classification for the Primitives layers.**

contact or mating takes place. On the one hand, contact phenomena is characterized by very rapid and large changes in force signals, almost approximating an impulse. To this end, positive and negative impulses were categorized for gradients with values greater or less than 70. On the other hand, for mating situations, there is little or no change in force, for this reason a constant label was assigned to signals with gradient values less than the absolute value of 1. In between these two extremes we chose to have three gradient categories for both positive and negative signals to give a general idea of the magnitude change registered for a signal. Fig.A.5 shows how the segmentation looks like across all five states (which are represented by five colored boxes) for the force signal in the x direction for PA10 related-experiments.

## A.2 Composites

The next layer of abstraction identifies seven basic motions compositions (MC) by looking at ordered-pair sequences of primitives. The MC layer set is comprised of: adjustment, increase, decrease, constant, contact, positive contact, negative contact, and unstable motions. The positive and negative contacts, imply the sign of the (gradient) of the action. A protocol was followed to minimize the effects of noise or erroneous segmentation. With respect to adjustments, primitives with big-to-small positive or negative gradients were considered as a positive or negative primitive category respectively. If a positive grouped primitive was followed by a negative grouped primitive an adjustment classification would be assigned to the ordered pair. Adjustments are motions in which the wrist records a quick 'back-and-forth" motion typically seen during alignment or insertion operations as the force controller tries to minimize residual errors. The reason to group positive and negative gradients is to maximize the likelihood of group adjustments even when the rate of change may be slightly different. Furthermore, for this particular category, we used a window of two data points instead of one to look for a matching pair (all other categories looked at the contiguous primitive). That is, if after finding a positive or negative gradient, and if the next data point was not negative or positive respectively, we would look at the next data point to look for a pair. Such procedure mitigates the presence of spurious signals that could prevent the proper grouping of an adjustment movement. The ordered pair groupings for motion composition classification are summarized in Fig. A.3. Note that the table contains sub-tables representing eight possible motion composition classifications. Each of which can be comprised of different sets of primitive groupings as illustrated in Fig. A.3. As with the primitives layer, 11 pieces of information were collected for each MC:

| MC's | Adjustment 'a' | | Increase 'I' | | Decrease 'd' | | Constant 'k' | |
|---|---|---|---|---|---|---|---|---|
| Gradient Combinations | Positive | Negative | Positive | Positive | Negative | Negative | Constant | Constant |
| | Positive | Adjustment | Positive | Constant | Negative | Constant | | |
| MC's | Positive Contact 'pc' | | Negative Contact 'nc' | | Contact | | Unstable 'u' | |
| Gradient Combinations | Positive | Pimp | Positive | Nimp | Pimp | Contact | Pimp | Pimp |
| | Negative | Pimp | Negative | Nimp | Nimp | Contact | Nimp | Nimp |
| | Pimp | Constant | Nimp | Constant | | | | |

**图 A.3: Motion Compositions according to primitive pairs in any order.**

composition label, average value, root means square value, amplitude, the labels of the first and second primitives, the starting and ending times for both primitives, and the average time for both primitives.

### A.2.1 Refinement

After the MCs are generated, a refinement phase was used to filter less significant signals and augment more significant signals. To do so, the compositions were analyzed under three contexts: (1) a composition's time duration, (2) a composition's amplitude magnitude, and (3) composition repetition patterns.
- Time Duration Context: this filter examines two contiguous MCs. If either composition is seven times bigger than the other, the smaller composition is merged to the larger one and all data is updated correspondingly. The duration ratio was determined empirically.
- Amplitude Value Context: this filter pertains to the formation of adjustment signals and constant signals. We considered three possibilities: (i) If there are contiguous primitives of types PC/NC or NC/PC, and if their amplitude is ten times smaller than the largest amplitude registered in the assembly, then treat them as an adjustment. This criteria seeks to disambiguate real contact signals and false ones by looking at their amplitude. Real contacts are characterized by large values. (ii) Similarly, if their is either an increase followed by a decrease and vice-versa, and both compositions have a similar amplitude (within (50%) of each other and they have a similar average value (100%) of each other, then merge as an adjustment and update the data correspondingly. (iii) If there is a sequence of an increase followed by a constant, or a decrease followed by a constant and vice-versa, and they have a similar amplitude (150%)and similar average value (100%), then merge them as a constant and update their information. This last filter targets small noisy signals that appear as increases or decreases but that in effect are constants. The amplitude threshold value is larger here to give more possibilities of catching increases or decreases within the narrow range of the constant's amplitude. - Repeated Compositions: the last filter takes signals that repeat and merges them as one. This filter is run iteratively until no more repetitions occur in the data.
The post-refinement composition layer results are shown in Fig. A.5 for a force signal sample in a Pivot Approach trial.

## A.3 Low-Level Behaviors

The taxonomy's third layer considers motion composition ordered pairs along with signal duration and amplitude to yield classifications. Eight LLB classifications were derived and labeled as: push, 'PS', pull, 'PS', contact, 'CT', fixed, 'FX', alignment, 'ALIGN', shift, 'SH', and noise, 'N'. The LLB formulation criteria is similar to those at the MC level. That is, for a pair of increase MCs labels, or decrease MCs labels, or constant MCs labels or adjust MCs labels; pull, push, fixed, or adjust LLBs are assigned respectively. As for contacts, if there is a positive contact followed by a negative one, or vice-versa, a contact LLB is assigned. One major difference between the MC level and the LLB level is introduction a shifting behavior 'SH'. Shifts and alignments are similar but differ in that, whenever there are two contiguous adjustment compositions, if the second composite's amplitude is larger than the first, label it as 'SH' LLB, if smaller label it 'ALGN'. With regards to the time duration context, if any motion composition lasts more that 100 milliseconds, it can by itself be a low-level behavior, or if the contiguous composition is of the same classification they can also merge correspondingly. If any composition is less than the allotted duration and it does not have a matching pair, it is considered a noisy signal. With regards to the amplitude context, if there are two adjustments within a window of 2 data points, and their amplitudes decrease, render such a pair as an alignment, otherwise consider it a shift (or growing de-alignment). As for paired increase, decreases, and constants, they will yield pull, push, and fixed low-level behaviors correspondingly. Finally, as for contacts, if there is a positive contact followed by a negative one, or vice-versa, or even a stand-alone contact motion primitive, render this is a low-level contact behavior.

### A.3.1 Refinement

The LLB layer is also followed by a refinement phase. The latter filters based on the same three contexts as used before:

- Time Context: this filter examines two contiguous behaviors (except for contacts).

If either behavior is five times bigger than the other, then merge towards the longer behavior and update the data correspondingly. LLB's are longer than compositions, so this threshold value is to be smaller than the one used for the composition's time duration filtering.

- Amplitude Context: the amplitude context pertains to alignments and shifts and there are four possible scenarios: (i) If there is a push-pull pair in either order and they have similar amplitudes (150%) and similar average values (100%) render then an alignment. (ii) If there is a shift followed by an alignment, or an alignment followed by a shift, where the second behavior has a smaller amplitude, then merge these as an alignment. This kind of merging is interesting because it can only be seen at this level of abstraction. While there may be a contiguous alignment-shift pair that was irreconcilable earlier, it can now be identified as an alignment. The same is done for a shift. (iii) Finally if there is an alignment followed by a pull or push or viceversa and they have similar amplitude (50%) and similar average value (100%), then merge as an alignment. In this case, after the previous refinement steps have been executed, if there are outstanding alignment-push or -pull pairs, the second behavior is a considered a continuation of the alignment and is merged. Shifts are treated similarly.

- Repeated Behaviors: As in the compositions layer, any two repeated behaviors can be merged as one. The post-refinement LLB layer is shown in Fig. A.5 for a sample signal in the Pivot Approach.

## A.4 High-Level Behaviors

The fifth layer contextualizes the process monitoring by asking what low-level behaviors principally describe the high-level human apropos behaviors found in the Pivot Approach: Approach, Rotation, (Alignment), Snap Insertion, and Mating[1]. Then, if key combinations of LLB's across the six force axes for a specific state are identified, then a certain HLB can be ascertained. For each state and corresponding

[1] In actuality, we do not directly assess the Approach stage given that there are no contact forces at this stage. But if the rotation analysis is successful we assume that the approach was too.

HLB an LLB or sequence of LLB's are matched with a particular force axis as part of the selected criteria. The criteria is connected both to the PA states, to the Controller Templates, and to the coordinate frame assignments in local coordinates (see Fig. ?? and Fig. ??). Given that we have two implementations of the PA and Controller templates for the PA10-Two Snaps configuration and the HIRO-Four Snaps configuration we have two sets of key LLB criteria. They are presented in Fig. A.4.

| HLB | LLB | Force Axis |
|---|---|---|
| Rotation | FX | Fx |
| | PL | Fz |
| | ALGN | Mx |
| Alignment | ALGN | Fx-My |
| | FX | Mz |
| Snap | CT+ALGN | Fz |
| | ALGN\|\|FX | Fx-Mz |
| Mating | FX | Fx-Mz |

| HLB | LLB | Force Axis |
|---|---|---|
| Rotation | FX | Fx |
| | FX | Fz |
| | FX | My |
| Snap | CT | Fx, My |
| | AL \|\| FX | Fy-Mx, Mz |
| Mating | FX | Fx-Mz |

**图 A.4: Comparison of the LLBs for both the PA10 and HIRO experimental configurations.**

### A.4.1 Key LLB's for PA10-Two Snap Experimental Configuration

The reasoning behind the selection of LLB's and axis for the Pivot Approach is intuitive. In state 2, the Rotation state, the wrist maintains a constant force along the z-direction, while the force along the y-direction diminishes as the wrist aligns itself with male part. The rotation about the x-axis can be seen through a series of large alignments along the moment's x-direction. For state 3, all axes are aligning in some form. For force elements, there is an alignment in position, for moment elements there is an alignment in orientation. The only exception to this is the moment about the z-direction. A pattern emerges where the moment axis that corresponds to the wrist's direction of motion for the insertion (i.e. the z-axis for the Pivot Approach) experiences little to no change throughout the assembly due to the nature of parts in the assembly task. For state 4, in the insertion state, Rusli studied typical force patterns for manually effected snap assemblies and states that initial resistance is characterizes the insertion until the snap-catch slips behind the undercut in the mating part, at which time an interlock occurs. In other words, one a large increase in force is expected upon contact, followed by a large decrease in force. Hence, we expect to see a contact label followed by an alignment label. Other axis can expect to experience an alignment at this stage. Finally, for the mating state, all signals should present no motion change and thus be classified with a FX behavior.

### A.4.2 Key LLB's for HIRO-Four Snap Experimental Configuration

In this experimental configuration, the Rotation controller is applying a constant force in both the x- and z-directions and a constant moment in the y-direction. For this reason we expect to find FX LLB tags in this state. Then, as for the insertion stage, experimental results consistently show that for successful assemblies there are CT LLB tags both in the force's x-direction and in the moment's y-direction. Both are correlated in that they represent the robot's downwards assembly motion. The rest of the axes are either aligning or fixed, that is, ALGN or FX tags should be seen in them. Finally, for the Mating state, there should be no movement and hence no change in gradient values if the structure is stable. FX labels are expected in all axis. The fourth layer results are shown in Fig. A.5. If the high-level behaviors can be ascertained, they print on the plot in green color. If they cannot be verified, they plot in red color representing failure.

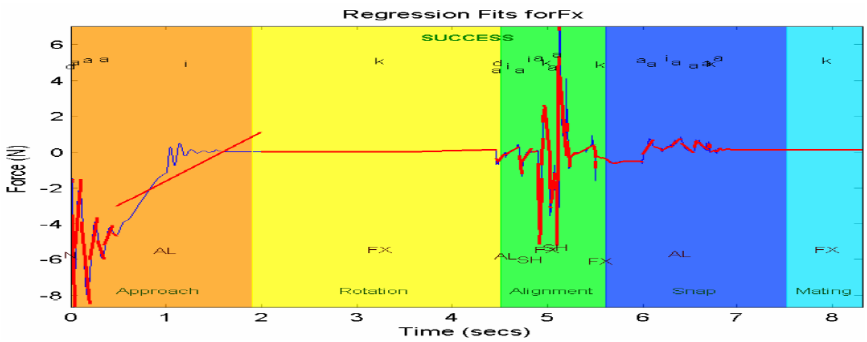


图 A.5: **This figure shows data related to the first four layers of the RCBHT. (1) The Primitive Layer: red line linear segments try to approximate original data and represent primitives. (2) The Composites Layer: composed by analysis of neighboring primitives. Corresponding labels appear in black at the top-most part of screen. (3) The LLB Layer: LLBs composed by analysis of neighboring composites. Corresponding labels appear in uppercase red letters below the graph. (4) The HLB Layer: HLBs derive from key LLBs. Corresponding labels appear in green at the bottom-most part of screen.**

# 致谢

感谢 Dr. Juan Rojas 在本论文写作提供的大力支持。没有他的支持，我没法完成这篇论文。感谢 zhibo 的中山 latex 模板，没有这个模板，本论文写作不会这么顺利。