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

罗伟强　软件工程

导师：Dr. Juan Rojas

摘要

我们使用了基于变化的多层分类法 (RCBHT) 来对力/扭力 (FT) 信号进行分析。RCBHT 会逐层抽象原始 FT 信号为标签，再对这些标签使用支持向量机进行分类。RCBHT 是为悬臂卡扣组装专门设计的信号分析方法，它对卡扣组装的 FT 信号进行分析并提炼出一系列的动作；而支持向量机是现在广为使用的分类法。在我们的工作当中，这两种方法被结合起来用以对错误案例进行分析，将它们区分出来。我们最终得到的高符合度测试结果显示 RCBHT 能够良好地表示组装过程，而对于 RCBHT 的结果，支持向量机可以很好地进行分类。分类的结果最后可用于纠正错误组装，从而减少因错误带来的损耗。

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

Waldron Luo Software Engineering

Directed by Dr. Juan Rojas

Abstract

Our work uses Relative Change Based Hierarchical Taxonomy System (RCBHT) to sample and represent specific force/torque signal with increasingly abstracted labels and uses Support Vector Machine (SVM) to identify failure cases. RCBHT is a taxonomy that are designed for recognizing the behavior sequence ofcantilever snap assembly. Support Vector Machine is one of the most widely used methods for classification. In our work, the two methods are combined to detect the failure cases among all trial cases. Our high accuracy hyphothesis result indicates that RCBHT is able to intuitively present this assembly process and SVM is good at classifying for this task. The classification will benefit the prevention of the detriment from happening and correcting the failure behaviors.

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

目录

第一章 Introduction

Uncertain failure occurs during a assembly task to the severe detriment to the assembly parts and the robot hands. Failures should be detected as early as possible and actions should be taken. In our work, we used SVM to identify the failure cases and detect the failure early in the assembly process. In our previous work, we built the framework for snap assembly in varying geometric complexity, including the generalizable strategy and controllers1 2, the state estimation3, the failure characterization4. All our contribution are integrated in 5.

The failure identification technique previously used was pRCBHT5. pRCBHT is consist of RCBHT, which is a taxonomy used multi layer labels to sample the assembly process, and bayesian filter, which takes the labels as input and predict the result of the assembly with Markovian assumption. In this paper, rather than using the probabilistic methods, Support Vector Machine(SVM) is used instead. Compared to bayesian filter, SVM is an easier approach in implementing and more intuitive to the classification. In our case, SVM was used with Primitive labels, Motion Composites labels (MC), Low-level Behaviors labels which are increasingly abstracted. The more abstracted labels contains less information but more refining. On the contrary, the less abstracted labels contains more information embodying the abundant noise.

SVM is one of the most widely used classification approach contemporarily. Compared to data-driven method using SVM6, RCBHT is Rather than data-driven, the labels of RCBHT can be said as gradient-driven. The premise of data-driven methods is that the success cases would have similar curves of signals. Similar curves also had similar upwards and downwards, and so we tried RCBHT with SVM to classify the assembly cases.

With LLB labels and MC labels of the whole process, failure cases can be identified. With the Approach state of Primitive labels, early failure detection is done in our approach. This is significant when failure detection is done early in assembly because cancellation and redoing can be called in time for prevention and cut the cost.

Other approaches including 7 and 8, signals were trainedwith Hidden Markov Model (HMM). Their methods are more general, but RCBHT was specially developed for the cantilever-snap assembly and increasingly abstracted layers will be more data-rich.

Though our approach has a high accrucy in identify failure cases. ....There is two limits. The first one is that with RCBHT, this approach is not general for other force/torque signal analysis. The second one 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, a simulator was used to produce the snap assembly process. HIRO, a simulated 6 DoF dual-arm anthropomorph robot was used in the OpenHRP 3.0 environment. CAD derived male and female camera parts were used and the male parts were mounted on the 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 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.

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

图 *2.1: Cantilever Snap*

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. Details about this can be seen in Appendix.

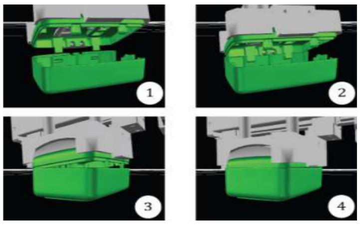

图 *2.2: Four control states*

2.3 Relative Change Based Hierarchical Taxonomy System

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 about RCBHT can be seen in 3. 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

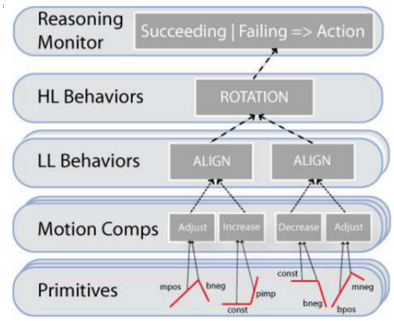

图 *2.3: Five Layers of RCBHT*

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 RCBHT using SVM

In this section we would introduce our classifier. Labels of three lower RCBHT layers will be used to construct fix-length vector. The premise of constructing RCBHT labels is that success cases have similar signal patterns in all six axis and similar patterns indicate the similar upwards and downwards.

In this section our approach will be introduced. With SVM, there's two ways to construct feature vector. In 6, they constructed feature vector based on the data itself. They used data at certain time stamp and combine them to be a fix-length vector. This is called data-driven method. With fix length vector, they used Gaussian kernel function to classify success and failure cases. The limit for this approach is that when there is some delay in between the states, the vector will be quite different and the case will be characterized as failure though the delay didn't effect the whole process.

We used RCBHT to sample our process, which is not data-driven method but gradient-driven. The taxonomy RCBHT is based on relative change. When a delay happens, usually, at the very begining of the whole process, with RCBHT, the vector will be the same as the delay didn't happen because the delay will be catagorized with neighbour label cons (In Primitive layer). Another reason for choosing RCBHT is that this taxonomy is specially developed for this task. The increasingly abstracted hierarchical taxonomy provides date-rich information. Varying layers of labels can construct to be different feature vectors and data-rich feature vectors provide more possible for this task.

This two approachs 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 RCBHT, we sample the whole process using Primitive labels, Motion Composites labels, Low-level behavior labels. Each axis, F_i, contains n entries, each represent a kind of label e_j. Such that $F_i = e_1, e_2, \ldots, e_n$. The whole vector is consist of six axis, $F_x, F_y, F_z, M_x, M_y, M_z$, and the vector will be conducted as $[F_x, F_y, F_z, M_x, M_y, M_z]$. Take the vector representing Primitive layer for example. 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 ω^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,

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

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

3.4 Implementation of classifer

In our work, libsvm 9 is used with Gaussian Kernel 10, 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

We conduct 100 tests for the classfication. In each test, we randomly selected half of the failure cases and half of the successful cases fir training and the other salves for testing. There are 150 failure cases and 42 successful cases. The diviation of each cases contains the diviation in x-axis, in y-axis, in z-axis and in Yall and Roll.

In RCBHT taxonomy, there are labels of Primitive layer (P), of Motion Composites layer (MC) and of Low-level behavior layer (LLB). For The LLB and MC way, they had average of 99.59% and 99.25% and minimum 98.9% and 93.8% accrucy respectively among the 100 tests. But if they were used with only the Approach stage, no success cases can be identified.

For P, we used only the labels from approach stage to classify. Though not as good as the previous two cases, but it still have 93.72% accrucy. the minimum of it is 89.6% among 100 tests. But if we used P with the whole process of labels, it also would misidentify all sucess cases as failure cases.

第五章 Discussion and Future Work

5.1 Discussion

The high accuracy of using the whole process of LLB and MC indicates that there are difference between success cases and failure cases and the taxonomy, RCBHT, is able to differentiate it. However, when only the Approach stage was measured, there are little information to identify the failure. In some cases, there was only one LLB in Approach state, which began at the very start and finished in the Rotation state, for some of their axis. While little information caused the lack of diversity, too much information with noise also prevent a good hypothesis. The Primitive labels of the whole process contain too much noise of viberation and friction. For only the approach stage, though the noise disturbed the accuracy, the Classifier still give good hypothesis. For the whole, the noise is too much.

The high accuracy of identifying failure cases proves that this approach is feasible. Gradient-driven signal processing is practical in differing signals when it's data rich. When the information is less, it's harder to give a good hypothesis. Those with such difficult, we conducted the prediction with only the Approach state, though the result is not good enough. It can divide most of the failure cases but not able to identify success cases very accurately.

Having said so, the ongoing case early in the assembly process is able to be classified as success or failure. When the prediction can be made at the begining of Rotation state, it's more likely for us to refine lateral actions, which may prevent the damage of the machine arm or breaking the cantilever snap.

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 and (3) tried to find more general solutions.

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 huge problems. The damage would be acute at the first wrong behavior. If the hypothesis could be made after this behavior occurs, or the second time it began, the machine arm had been have some damage.

RCBHT is specially developed for this task but when it comes to dual hand assembly, this taxonomy is not yet proved to be suitable. After all, there may be more kinds of snap assembly including but not limited in different geomatric difficulty, different machine hand pre-setting. And so, solutions should be general enough for all these configurations.

参考文献

[1] 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.

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

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

[4] 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.

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

[6] A. R. D. B. M. Mason and G. F. R. J. Wang. Failure Detection in Assembly: Force Signature Analysis.

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

[8] 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.

[9] 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.

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