Proceedings of 2014 IEEE
International Conference on Mechatronics and Automation
August 3 - 6, Tianjin, China

Cantilever Snap Assemblies Failure Detection using SVMs and the RCBHT.

Weiqiang Luo, Juan Rojas, TianQiang Guan
School of Software
Sun Yat Sen University
Guangzhou, Guangdong, 510006, China

Kensuke Harada, Kazuyuki Nagata.
Intelligent Sys. Research Institute, AIST
Tsukuba, Ibaraki, 305-8568, Japan

***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 abstracted behavioral features to classify failure detection in cantilever snap assembly problems. The approach was useful in detecting failure offline 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 due to their coarse and global representations.

I. INTRODUCTION

Failure detection and correction play an increasingly significant practical role both in the industrial processes with ever shortening life-cycles and with robots that serve in unstructured and uncertain environments. This work focuses on cantilever snap assemblies, which are often used in manufacturing, and will grow in importance for personal and service robots. In robotics, failure detection methods can be divided into model or model-free approaches. The former use theoretical system designs to identify failure, while the latter accumulate experimental data for both outcomes. Model-free approaches can be computationally expensive and damaging to robot equipment given that these experiment types need to be physically implemented Recent work has focused on implementing supervised learning techniques that require relatively few trials to learn failure detection [1]. Failure detection has traditionally focused on parts assembly [2], tool breakage [3], [4], and threaded fastener assembly [5]. Recent work has used support vector machines and principal component analysis to classify successful or failed assemblies upon completion. [1], [3]. A similar work tried to continuously monitor a task and identify early failure by combining relevance vector machines with a Markov chain model [6]. No work has yet to focus on snap assemblies. The latter are challenging due to their elastic nature and complex geometrical configuration. Our work, focuses on detecting failure in cantilever snap assemblies using support vector machines (SVMs) to perform supervised learning and classification. For cantilever snaps, there are three common fastener types, out of which Cantilever snaps are the most common [7]. Detecting failure and failure modes in snap assemblies is complex due to intricate force signatures that stem both from complex hardware configurations and selected motion strategies to accomplish a task. The complexity is such that assembly planning methods like contact-state graphs [8] have yet to be used in this context.

Some recent work on assembly strategies has been implemented for cantilever snap assemblies. In [9], a strategy exploited parts' hardware design and constrained the motion while in [10] empirically devised instructions were created. In regards to estimation, [9] a continuous non-parametric system produced high-level abstractions and used rule-based and Bayesian models to reason about the task. Likewise, [11] used a sticky-Dirichlet Process Hidden Markov Model to perform continuous estimation of human-labeled nominal executions. With respect to failure and failure sub-mode classification, [12] studied cases in which assemblies deviated from a nominal trajectory in 1 of 3 directions. They empirically correlated salient features with single deviation directions and studied if they could classify failure modes even when deviations were superimposed in 2 or 3 directions. They also computed correlation data to validate if the classified deviation-directions actually took place.

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 in a cantilever snap assembly. We would also like to study whether failure can be detected early in the assembly whenever the initial contact of parts is inadequate. This question is of importance for two reasons: (i) the behavioral representations are part of a data-rich framework that can be used for a variety of estimation settings, and (ii) the dimensionality of the label set is much smaller than that of the six dimensional force vector. For this work we consider an SVM that uses features that represent abstracted (labelled) behaviors from the assembly force signature. These abstracted behaviors are produced by the Relative change-based Hierarchical Taxonomy (RCBHT) framework presented in [13]. The RCBHT consists of four increasingly abstract layers plus a monitor (the SVM in this work) and is based on the premise that patterns of relative-change, aided by contextual information, can be encoded through a small label set and abstracted to a human intuitive level (see Fig. 3). Each layer in the RCBHT consists of a fixed set of labels. The higher the abstraction

978-1-4799-3979-4/14/$31.00 ©2014 IEEE

level, the smaller the label set. Furthermore, contextual information refers to what: task state, force axis, abstraction level, and behavior sequence pertain to an encoded behavior. The framework is designed to generalize to cantilever parts of varying complexity through an optimization routine defined in [14]. The RCBHT also works in concert with an assembly strategy that generalizes to cantilever snaps of varying geometrical complexity, as in [15]. The strategy exploits hardware design to constraint the motion and subdivide an assembly into four states: Approach, Rotation, Insertion, and Mating. The Support Vector classifier is tested against three different sets of feature labels from the RCBHT. These label sets come from the first three layers of the taxonomy and are tested separately to analyze the accuracy of the classifier in detecting failure. Furthermore, we are interested in examining which label sets are better to detect failure early in the task (we only look at the Approach stage) or the entire task (only consider the Approach state).

II. Experimental Setup

NX-HIRO, a 6 DoF dual-arm anthropomorph robot was simulated in the OpenHRP 3.0 environment [16]. Male and female 4-snap cantilever camera parts were used. The male part was mounted on the robot's wrist, while the female snap was rigidly fixed to the ground as in Fig. 1. In this work a generalizable cantilever snap assembly strategy named the Pivot Approach (PA) strategy was used. The strategy exploits snap parts' hardware design to constraint the task's motion about a docking pivot (see Fig. 1) and generate both similar motions and similar sensory-signal patterns across assembly tasks. The strategy systematically discretizes the assembly into four intuitive states: the Approach, Rotation, Insertion, and Mating states shown in Fig. 2. For this paper, it suffices to know that the Approach state drives the male part to contact the docking pivot on the anterior side of the female snap at an angle. The Rotation state, rotates the male part about the world's y-axis until contact is made by the posterior snaps. The Insertion state drives the male part into the female part while further aligning the parts until the parts snap. Finally, the Mating state maintains both parts stably connected.

As for failure generation, we have enacted a strategy that inserts noise deviations in linear and angular directions. For linear translations we selected the $+x$ and $\pm y$ axes with respect to the world frame. For rotations we selected the z-axis ($\pm\phi$). We highlight the deviation direction for two reasons: (i) the FT signals have very different signatures given the asymmetrical part's geometry, and (ii) we are interested in detecting the types of failure (failures sub-modes) that occur in future works. The deviations were constrained within the following bounds: $0.0075m \leq x \leq 0.0105m$; $\pm 0.0075m \leq y \leq \pm 0.0105m$; and ± 0.1745rads $\leq \theta \leq \pm 0.5066$rads.

III. The Relative Change-Based Hierarchical Taxonomy

The RCBHT yields state representations by hierarchically abstracting snap assembly force-torque (FT) data in increasingly intuitive ways. The hierarchical taxonomy is composed of four increasingly abstract layers that encode relative-change in the task's force signatures and one task monitor. The system's taxonomy (see Fig. 3) is built on the premise that relative-change patterns can be classified through a small set of categoric labels and aided by contextual information. The RCBHT analyzes FT signatures from all force axes independently and yields encoded behavior according to a set of contextual rules that consider the task state, FT axis, taxonomy layer, and behavior sequence of a task.

A. Primitive Layer

The first layer, the Primitives, partitions FT data into linear data segments and classifies them according to gradient magnitude. Linear regression along with a correlation measure are used to segment data when a minimum correlation threshold is flagged. Gradient classification has three main subgroups: positive, negative, and constant gradients. Both the positive and negative sets are subdivided into 4 ranges: small, medium, large, and very large. Contact phenomena is characterized by abrupt changes in force signals almost approximating an (positive or negative) impulse, it is such

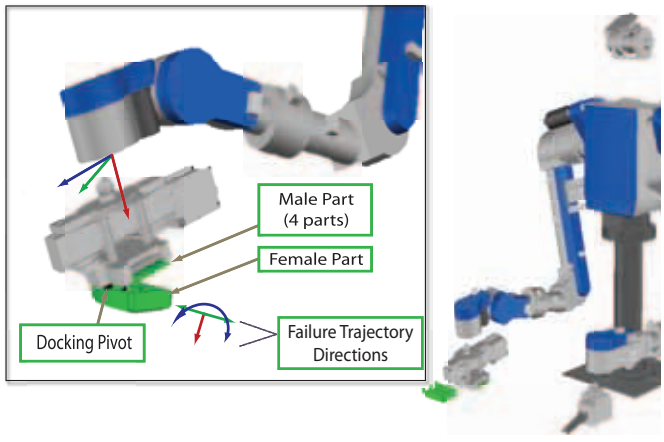

Fig. 1. The HIRO-NX robot performing a cantilever snap assembly.

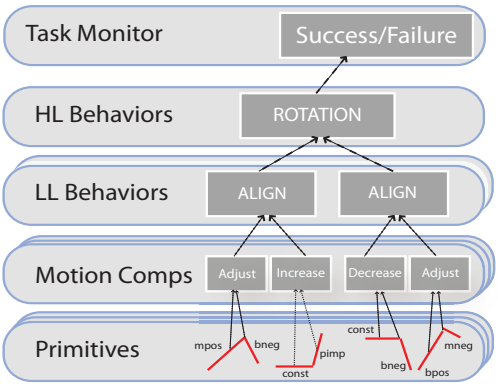

Fig. 3. The RCBHT taxonomy consists of four layers and a monitor. Each succeeding layer encodes behavior at increasing levels of abstractions. Each layer has a particular set of labeled behaviors that can be use by the monitor to estimate the state of the assembly.

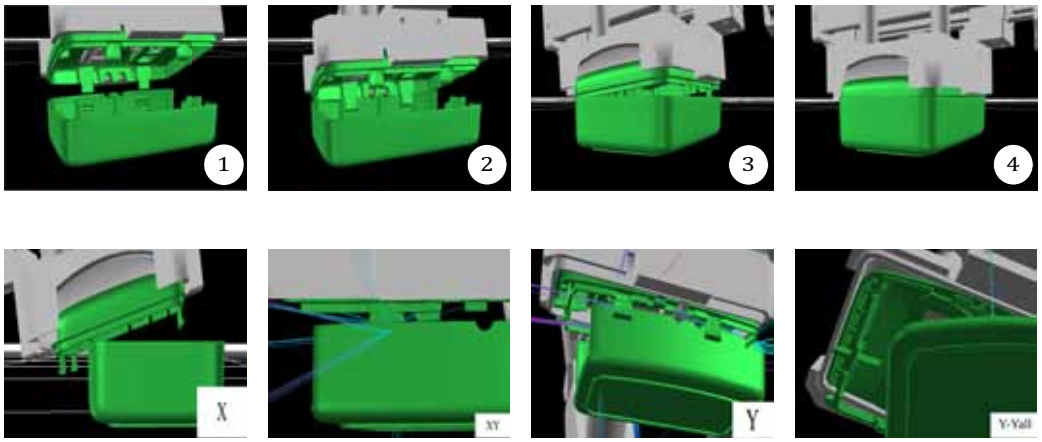

Fig. 2. Top: A successful assembly strategy consists of four states: Approach, Rotation, Snap, and Mating. Bottom: Failed assemblies consisted of modifying the trajectory of the male part about the x and y axis, and rotated about x (yaw) as well. Here you can see representative deviations in the x, y, x-y, and y-yaw directions.

impulses that are characteristic of snap motions. Negative and positive gradient ranges are labelled as:"sneg, mneg, bneg, nimp" and "spos, mpos, bpos, and pimp" respectively. Constant gradients are those whose change is trivial. They are classified as such if their gradient magnitude is lower than the absolute value of a calibrated threshold and are labeled as "const". In order to generalize the parameter thresholds for this layer, an optimization routine devised in [14] uses contextual information derived from the assembly strategy to determine where the largest gradients and the near-constant gradients are located according to task state and axis. For details see [14]. Filtering is executed for this and every layer of the taxonomy. Filtering merges extracted behaviors based on three different criteria: (i) a time-duration context, (ii) repeated behaviors, and an (iii) amplitude value context. The time-duration context seeks to eliminate behaviors whose duration is so small that they become negligible. The repeated behavior principles merge repeated behaviors and clean noisy signals. The amplitude criteria examines adjacent signals magnitudes. If one is much larger than the other (except for impulse signals), the lower amplitude signal is merged with the larger one. Each layer runs a filtering cycle 2-3 times to reduce the label number to a most representative number. There are other details concerning filtering which are not necessary for this discussion but can be found in [17].

B. Motion Composition Layer

The next layer analyzes ordered-pair Primitives sequences to create "Motion Compositions" (MC's). By studying patterns in the ordered-pairs, seven sets of higher-level abstractions are extracted. These actions represent force-torque behaviors labeled as: adjustments, 'a', contacts 'c', increases, 'i', decreases, 'd', constants, 'k', and unstable motions, 'u'. Adjustments are motions where a positive/negative gradient is followed by a negative/positive gradient respectively. They represent a small rattle motion between male and female snap parts. Additionally, if two positive or negative gradients succeed each other, it points to actions in which the force or torque is increasing or decreasing respectively. With respect to our previous statements, we treat impulse gradients (pimp, or nimp) independently. We acknowledge that when a positive impulse is followed by a negative one or vice-versa a contact action between male and female parts is likely. Additionally, if two positive or negative impulses succeed each other, it may lead to unstable behavior.

Besides the assigned label, each MC possess quantitative data, including: average magnitude values, maximum signal values, average amplitude values, and starting and ending times for each of the primitives[1].

C. Low-Level Behaviors

The taxonomy's third layer considers MC ordered pairs and uses the same contextual information (signal duration and amplitude values) to yield another set of higher-abstraction classifications called low-level behaviors (LLB's). Behaviors at this level become more intuitive. The process of continuing to abstract in the same way, reveal high-level details that are not apparent at more granular levels of the taxonomy. Seven LLBs were identified and labeled as: push, 'PS', pull, 'PL', contact, 'CT', fixed, 'FX', alignment, 'ALIGN', shift, 'SH', noise, 'N'. The LLB formulation criteria is similar to those at the MC level.

D. High-Level Behaviors

The fourth layer, the high-level behaviors (HLB's) layer is different than the previous three. This layer connects a top-down approach with the bottom-up approach we have been describing and makes temporal inferences about "what key LLB's should appear in a: (i) given coordinate axis, and (ii) task states"; for the Pivot Approach's individual task states to be identified. For this work, the HLB layer is not used with the SVM classifier to detect failure. To test the accuracy of the classifier, we utilize behavior labels from the first *three layers* of the taxonomy as feature inputs to the SVM. A visual representation of the RCBHT is show in Fig. 3.

[1]From this work onwards, the RMS value field in the RCBHT has been updated to represent a maximum signal value

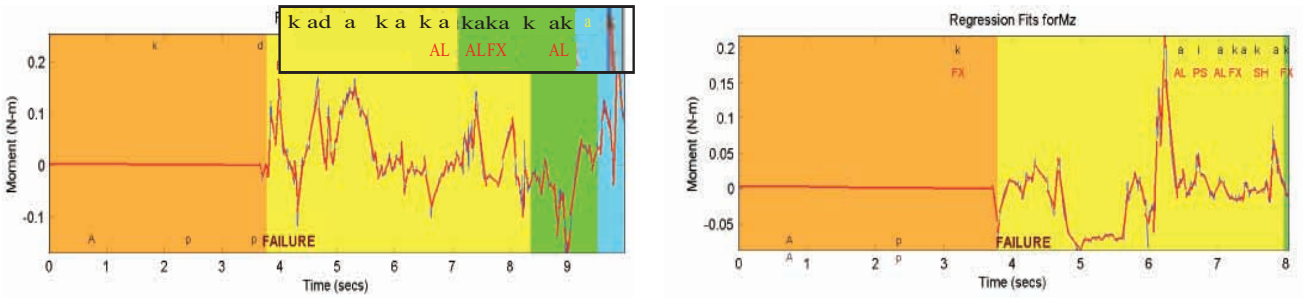

Fig. 4. RCBHT example in My axis for a successful case (left) and failure case (right) with deviation in the x-direction. The RCBHT shows primitives with red lines, MC's with black labels, LLB's with red labels. The orange, yellow, green, and blue boxes represents the Pivot Approach's states. On left sub-figure, ata the top right corner, labels have been zoomed in for clarity.

Additionally, in Fig. 4 the sub-figure on the left shows an example of RCBHT labelling in the My axis for a successful assembly, and the sub-figure on the right shows a labelling example for a failure case with deviation in the x-direction.

IV. SVM

In this work a linear Support Vector Machine was used as a binary classifier between successful and failed cantilever snap assembly tasks. In our work the classifier used an indirect measure of force signatures; namely, RCBHT encoded behaviors. The problem is interesting as the feature size is significantly decreased, a one dimensional vector by the size of the RCBHT label set for six axes. A typical feature vector based on a force signature may be: the six-force dimensions by 2000 sample points for a 10 second assembly sampled at 200Hz. Furthermore, if the classification works, it ratifies the effectiveness of the encoding in the RCBHT.

In particular, we wish to analyze the accuracy of the classifier by using three different input features vectors. Each vector corresponds to the encoded behaviors from each of the first three RCBHT layers (and for each of the six FT axes). The first layer is the Primitives layer. The latter yields the most granular behavior at the expense of higher noise levels. It is useful to understand local and short-duration events. This layer has a 9 label set (see Section III-A, however, the total label count during one assembly trial will be much greater than any of the other RCBHT Layers. Next is the MC layer which gives a higher level of abstraction and consists of a 7 label set. The latter's granularity level is between that of the P and the LLB layer–appropriate for medium trends. Finally, the LLB layer offers the highest abstraction level and captures more appreciable behaviors in the task. The LLB also consists of a 7 label set.

To ensure that our input feature vector maintains the same size across different assembly trials, we construct the input feature vector as a one dimensional vector that consists of a given number of labels l_i, across six force-torque axes F_j, for a total of $1\text{x}\mathbb{R}^{(l_i * F_j)}$. Each entry of the input feature vector *counts* the total number of corresponding labels that appeared in the assembly task for the appropriate force axis. The dimension for our feature vectors would be as follows: P-layer, $1\text{x}\mathbb{R}^{54}$; MC- and LLB-layers, $1\text{x}\mathbb{R}^{42}$. While our current labeling scheme does not consider time, the presence of specific labels in a given axes has a characterizing ability. We will later analyze which of the three layers is more useful for classification.

Linear Support Vector Machines approximate a boundary to separate binary classes through a hyperplane for large feature spaces. Our feature vector x is of dimension $\mathbb{R}^{(l_i * F_j)}$ and it is used to learn a hyperplane: $\omega^T x - b = 0$, where ω are the weights and b is the bias from the zero point. In effect, the separation of each training point from the hyperplane is the functional margin $\hat{\gamma}^{(i)}$ and can be modeled as:

$$\hat{\gamma}^{(i)} = y^{(i)}(\omega^{(i)} x + b) \tag{1}$$

Here the pair $\{y^{(i)}, x^{(i)}\}$ represent task outcome (success or failure) as $y^{(i)} \in \{1, -1\}$ and $x^{(i)}$ is the input vector for training and testing. The SVM optimizes the functional margin by maximizing the distance to both successful and failure cases by solving the quadratic programming problem:

$$\begin{aligned} &max\ \gamma \\ &s.t.\ \gamma = \min_{i=1,\ldots,m} \hat{\gamma} \quad , \end{aligned} \tag{2}$$

where, γ is the geometrical margin of the input points from the hyperplane. The larger the geometrical margin the more accurate the classifier. Our linear classifier was implemented using the open-source "libsvm" algorithm along with a Gaussian kernel [18].

It is worth noting here that the SVM classifier constructed here is limited in its adaptability by its training conditions. This classifier will work for cantilever snap assemblies of 4 snap parts, but would need to be retrained if the snap number changed or the strategy changed. Nonetheless, the training procedure is simple and can be re-trained for different conditions.

V. Experiments

As part of our experimental setup, the SVM classifier was trained and tested using a Gaussian kernel. A total of 192 assemblies were executed. Of these, 150 were failure cases and 42 were success cases. Note that as part of the failure cases, we have failed assemblies with trajectory deviations in the $x-$, $\pm y-$, $\pm\phi-$ directions, as well as combinations of all of these. Half of the failure and success cases were used

for training, while the other half were used for testing. With regards to input feature vectors, three separate vectors were constructed. Each feature vector was implemented by using one label set for each of the first three layers described in Sec. IV. Note that as part of our classification problem, we wish to identify the classification accuracy as part of two different scenarios: (i) early failure detection, consists of labels just in the Approach state; and (ii) late failure detection, consisting of labels in all four states of the Assembly task (Approach-Mating).

A. Training and Testing Methodology

In analyzing the accuracy of the classifier, we must note that there is a tradeoff between the size of the input feature vector and the accuracy of the latter. To this end, during training we varied the number of training samples from 5 to 96 samples. We started with 5 so as to include 1 success case for every 4 cases (this is the approximate ratio between our 150 failure cases and our 42 failure cases). When incrementing the number of samples for training, we do not randomize. We append new trials to the ones already selected. This was done in consideration of the significant variability in failure deviations as part of our failure assembly task set. If we randomize training samples while only training with a small number of trials, testing will show (great) variability as a function of what training cases were selected. For testing, all 96 (75 failure and 21 success) cases are used to measure accuracy. Furthermore, we repeat a test 10 times (a total of 960 trials) and compute the average accuracy across the 10 tests.

B. Early Failure Detection

For early failure detection we construct our input feature vector consisting of those labels that show only during the Approach state for all six FT axes. Training and testing is conducted as per Section V-A. Interestingly enough, when just considering the Approach state, the classifier trained with MC labels and LLB labels cannot detect any success cases, rendering the classifier inconsequential here. However, when the classifier is trained with the P labels, the results are satisfactory as seen in Fig. 5. For the P-trained classifier, the classifier reached an average asymptotic maximum accuracy value of 93.72% and a minimum of 89.6%. It took about 60 trials to reach the asymptotic value.

C. Late Failure Detection

For late failure detection we construct our input feature vector consisting of those labels that show throughout all 4 states of the task (Approach-Mating) for all six FT axes. Training and testing is conducted as per Section V-A. In this case, the classifier trained with P labels could not identify any success cases. On the other hand, MC and LLB trained classifiers performed well as shown in Fig. 6 and Fig. 7. For The LLB and MC layers, the classifier had an average asymptotic maximum accuracy value of 99.59% and 99.25% respectively and a minimum of 98.9% and 93.8%. The MC

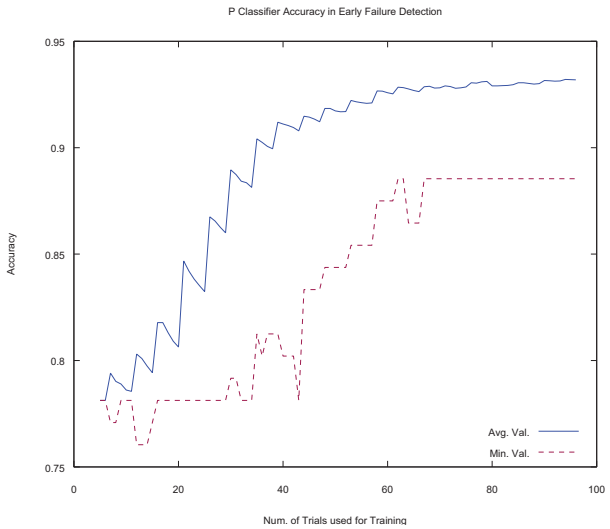

Fig. 5. Early detection failure accuracy for classifier trained with Primitive layer labels.

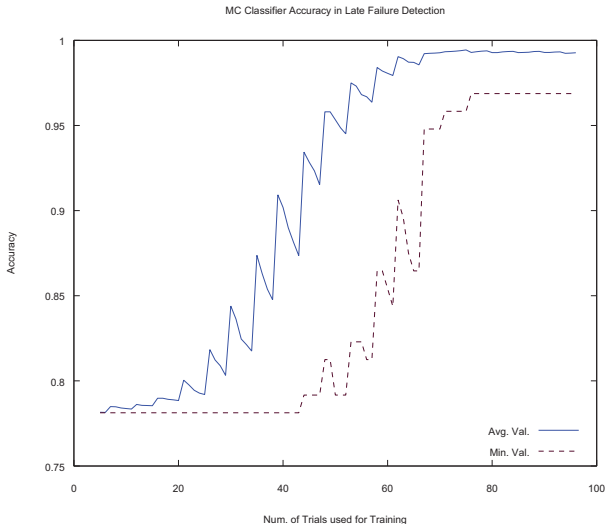

Fig. 6. Early detection failure accuracy for classifier trained with Primitive layer labels.

classifier reached asymptotic value after about 70 trials while the LLB classifier did so after approximately 22 trials.

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.

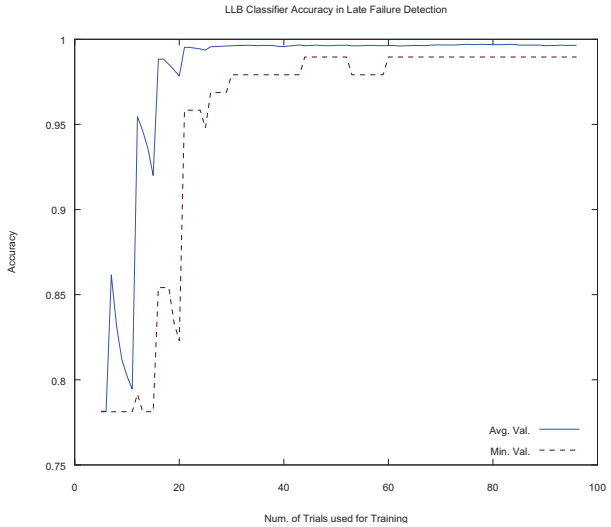

Fig. 7. Early detection failure accuracy for classifier trained with Primitive layer labels.

VI. DISCUSSION

The SVM 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. In fact, the LLB label set performed extremely well. Reaching an accuracy level of 99.59% in just over 22 trials. This is comparable to what others have done as in [1]. However, being able to classify using indirect behavior feature sets is an interesting approach. Such labels and their quantitative data play an important role in estimation to help the robot perform active sensing and later error correction.

For our future work, we would like to extend our classification to sub-failure modes and do so in real-time. We also will seek to implement solutions that can generalize more flexibly to other types of problems.

VII. CONCLUSION

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. This work contribution consists in creating a robust input feature vector using a very small set of encoded behavior features, which in turn play an important role in state estimation tasks and error correction mechanisms.

REFERENCES

[1] A. Rodriguez, D. Bourne, M. Mason, G. F. Rossano, and J. Wang, "Failure detection in assembly: Force signature analysis," in *IEEE Conference on Automation Science and Engineering*, 2010.

[2] D. M. Fullmer, "Parts assembly using signature analysis," US Patent 4,855,923, Aug. 1989.

[3] S. Cho, S. Asfour, A. Onar, and N. Kaundinya, "Tool breakage detection using support vector machine learning in a milling process," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 3, pp. 241–249, 2005.

[4] 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*, vol. 37, no. 9-10, pp. 872–880, 2008.

[5] K. Althoefer, B. Lara, Y. H. Zweiri, and L. D. Seneviratne, "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*, vol. 222, no. 6, pp. 1081–1095, 2008. [Online]. Available: http://pic.sagepub.com/content/222/6/1081.abstract

[6] A. Rodriguez, M. T. Mason, S. S. Srinivasa, M. Bernstein, and A. Zirbel, "Abort and retry in grasping," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1804–1810.

[7] R. Soddhi and M. Sonnenberg, "Use of snap-fit fasteners in the multi-life-cycle design of products," in *IEEE Intl. Symp. on Electr. & Env.*, 1999.

[8] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. D. Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 218–231, 2007.

[9] omitted, "for review," *omitted*, vol. 00, no. 2, pp. 00–00, 9999.

[10] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, "Force controlled assembly of emergency stop button," in *IEEE Int'l Conf. on Robotics & Automation*, 2011.

[11] E. Di Lello, M. Klotzbucher, T. De Laet, and H. Bruyninckx, "Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5827–5833.

[12] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, and K. Nagata, "Early failure characterization of cantilever snap assemblies using the pa-rcbht," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014 (in-print).

[13] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai, "A relative-change-based hierarchical taxonomy for cantilever-snap assembly verification," in *IEEE Intl Conf. on Robots and Systems*, 2012.

[14] ——, "Gradient calibration for the rcbht cantilever snap verification system." in *IEEE Int'l Conference on Robotics and Biomimetics*, 2012.

[15] ——, "Cantilever snap assembly automation using a constraint-based pivot approach," in *IEEE Intl. Conf. on Mechatr. & Automation*, 2012.

[16] F. Kanehiro, H. Hirukawana, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *Intl. J. of Robotics Res.*, vol. 23, no. 2, pp. 155–165, 2004.

[17] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai, "Towards snap sensing," *International Journal of Mechatronics and Automation*, vol. 3, no. 2, pp. 69–93, 2013.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.