6th annual IEEE Conference on Automation Science and Engineering
Marriott Eaton Centre Hotel
Toronto, Ontario, Canada, August 21-24, 2010

Failure Detection in Assembly: Force Signature Analysis

Alberto Rodriguez David Bourne Mathew Mason Gregory F. Rossano JianJun Wang

***Abstract*— This paper addresses failure detection in automated parts assembly, using the force signature captured during the contact phase of the assembly process. We use a supervised learning approach, specifically a Support Vector Machine (SVM), to distinguish between successful and failed assemblies. This paper describes our implementation and experimental results obtained with an electronic assembly application. We also analyze the tradeoff between system accuracy and number of training examples. We show that a less expensive sensor (a single-axis load cell instead of a six-axis force/torque sensor) provides enough information to detect failure. Finally, we use Principal Component Analysis (PCA) to compress the force signature and as a result reduce the number of examples required to train the system.**

***Index Terms*— Assembly, force signature, signature analysis, failure detection, SVM, PCA.**

I. INTRODUCTION

Flexibility, adaptability and efficiency are key goals of modern manufacturing systems. To accomplish these goals it is critical to identify and correct failures at the earliest opportunity. Most systems leave failure analysis and correction to human supervisors, while providing only the most minimal tools for evaluation.

The product life cycle of manufactured products is becoming short enough that it is cost effective neither to train humans nor to develop expensive hard automation systems. Despite work in academia [1], [2] and industry [3], [4], humans still account for most of the assembly work in small electronic assemblies, such as computers, cellphones and cameras. Automated assembly has a huge potential competitive impact in such electronics industries. However, it requires complex mechanical models for part interaction, which complicates the development of reliable assembly strategies. This continues to be a very active area of research.

Any source of uncertainty in modeling the assembly process increases the likelihood of failure, which is reflected on the quality of the product. This makes failure detection and failure correction a central feature for increasing the reliability of the manufacture process. Camarinha-Matos et al. proposed in [5] a four stage general framework for autonomous assembly: global coordination and dispatching; monitoring of the execution; failure diagnosis; and failure recovery. A flexible system has to be able to cope with execution failures, and a reliable failure detection system is a first step towards that.

In this work, we implement a flexible methodology for detecting failure, based on the force signature of the assembly. The system uses a small set of correct and incorrect assembly examples to learn the difference between success and failure.

Neither the sensing device (a force sensor) nor the learning methodology are customized for our particular assembly, and should be readily adaptable to other assembly problems.

II. PREVIOUS WORK

Previous work in signature analysis for error detection is extensive. Willsky [6] did an early survey of methodologies for detecting failure in dynamic systems. Surveyed methods detect and analyze abrupt changes in the evolution of the system to diagnose deviations from its expected behavior. Hodge and Austin [7] contains a more recent survey of outlier detection methodologies for similar purposes.

Depending on how the expected behavior of a system is specified, error detection approaches are divided into *model-based* and *data-driven* methods. Model-based methods identify potential failure modes based on a theoretical model of the system, while data-driven approaches populate both correct and incorrect behaviors by example.

Model-based methods are the most extensively explored in the literature. They have been used for detection of failure modes in induction motors [8], surveillance of mechanical systems [9], structural damage detection [10], [11], and tool condition monitoring in machining operations [12] among several other applications.

The use of data-driven approaches is not as widespread, mainly due to the difficulty of collecting enough data to accurately characterize both correct and incorrect behaviors of the system. When that is not a constraint, a data-driven approach avoids the difficulty of specifying complex process models. Support Vector Machines have been used by Cho et al. [13] and Hsueh and Yang [14] to detect tool breakage in milling operations based on the force signature of the process. Tax et al. [15] have proposed a closely related Support Vector method to analyze machine vibration and Althoefer et al. [16] have used a neural network to monitor the insertion of self-tapping threaded fasteners using torque signals.

With a data-driven focus, and using an approach similar to Fullmer's [17], we study the use of force signature to detect incorrect insertions in small electronic assemblies.

Alberto Rodriguez, David Bourne and Mathew Mason are with the Robotics Institute at Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
albertor@cmu.edu
Gregory F. Rossano and JianJun Wang are with ABB Corporate Research, 2000 Day Hill Road, Windsor, CT 06095
gregory.f.rossano@us.abb.com

978-1-4244-5449-5/10/$26.00 ©2010 IEEE

III. FAILURE DETECTION BASED ON FORCE SIGNATURE

For the rest of the paper, we assume a concrete assembly problem. With it, both the assembly station (Fig. 1) and the strategy for mating the parts are fixed, but not explicitly modeled. We also suppose access to the force signature of the entire process by means of a force sensing device. This work builds on the assumption that the signal captured by that device correlates with the outcome of the assembly.

Fig. 1. Assembly station used for the experimental section. Gripper attached to a 6 DOF industrial manipulator and force sensor under the assembly base.

The assembly strategy is given and, at present, we are not concerned with how it is produced. For the experimental section, we design it by hand, without considering its effectiveness or optimality. One of the parts to be assembled lies on top of the assembly base, Fig. 1, while the other is firmly held by the robotic manipulator. The force sensor is located underneath the assembly base and captures the forces generated during the contact phase of the assembly of the parts.

The failure detection method proposed here does not generalize across different assembly problems, e.g. changing parts to mate, assembly strategy or structure of the assembly station. Once the system is trained, prediction of success or failure is only valid under the same conditions used during the training phase. However, the algorithm is model free, hence can be trained for any specific instances of those same conditions.

Force signature refers to the signal captured by the force sensing device. Fig. 2 shows an example of the X, Y and Z components of the force signature of a successful and some unsuccessful assemblies. As notation for the rest of the paper, let $F = (F_1 \ldots F_T)$ be any of the components of the force signature (F_X, F_Y, F_Z, T_X, T_Y or T_Z) at timestamps $1 \ldots T$.

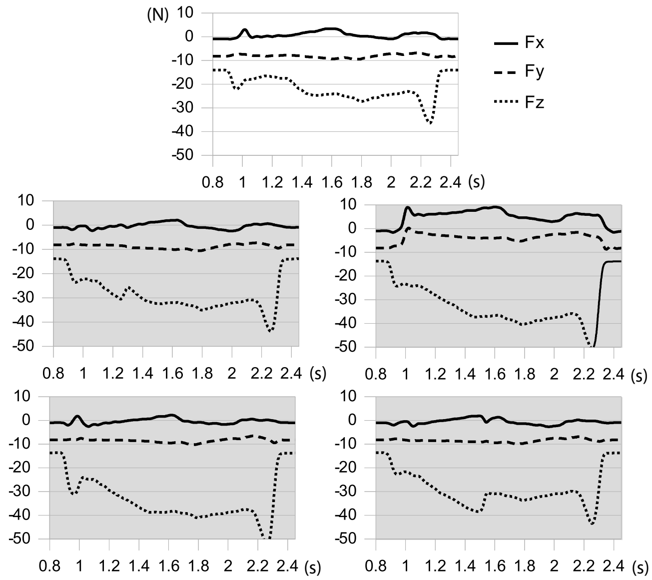

Fig. 2. Example of force signatures (F_x, F_Y and F_Z) for a successful assembly (top) and four failed attempts.

A. Learning to Detect Failure

While most previous work uses the model-based approach, three main reasons incline us to adopt a data-driven approach:

- It is rather challenging to find a contact model capable of predicting the forces involved in the assembly process.
- An assembly might fail or succeed for different reasons. Even with an accurate contact model, it is hard to enumerate by hand all possible failure modes.
- We want a methodology independent of the specific assembly problem. A model-based approach requires a specific analysis for every different assembly.

To overcome these challenges we train a classifier to automatically learn the decision rule between success and failure from the force signature of N hand labeled examples.

Among several different techniques for supervised classification, we chose linear Support Vector Machines [18], [19]. Linear SVMs model the separation boundary between classes as a hyperplane in a large feature space. Each reading of the force sensor corresponds to one dimension of the feature vector x. If the number of samples of the force signature is T and we use a 6 axis F/T sensor, the feature space becomes $6T$-dimensional. In its simplest form, an SVM finds the hyperplane $w \cdot x - b = 0$ that separates and maximizes the margin to both positive and negative examples, by solving the quadratic programming problem:

$$\min_{w,b} \|w\| \quad \text{s.t.} \quad c^{(i)} \left(w \cdot x^{(i)} - b \right) \leq 1 \ \forall i \qquad (1)$$

where $x^{(i)}$ is the feature vector of training example i and $c^{(i)} \in \{+1, -1\}$ the corresponding label. In this work we use SVMlight [20], an available SVM implementation free for scientific use.

There is a tradeoff between the learning capabilities of the SVM and the number of examples needed to estimate the hyperplane. The relationship is set by the dimension of the feature space. The rule of thumb is: the bigger the dimension of the feature space, the more capable the SVM is, but the more examples it needs to learn the hyperplane.

Section IV-A presents an estimation of the accuracy in discriminating success from failure, based on the force signature of the assembly.

Section IV-B analyzes the effect of simplifying the assembly station and switching from a six axis F/T sensor to a single axis load cell.

Finally, Section IV-C explores the relationship between the dimension of the feature space, the learning capabilities of the SVM, and the amount of data required for the SVM to learn an accurate model.

B. Preprocessing of Force Signatures

For comparisons between force signatures to make sense, samples need to be consistent across assembly attempts. For this reason, we need to preprocess the force signatures:

- *Resampling*: If the sampling of the force signature is not completely deterministic, we need to resample it to obtain samples aligned in time. For that, we use Locally Weighted Regression [21] with a gaussian kernel. The smoothing parameter γ (bandwidth of the interpolation) is automatically chosen to minimize the cross validation error [22].
- *Alignment*: After resampling, signatures are shifted in order to get a close alignment. The misalignment is automatically estimated with the cross-correlation of the signatures and a reference signature of a successful assembly. Fig. 3 shows the histogram of the misalignment of the signatures captured in the experimental section.

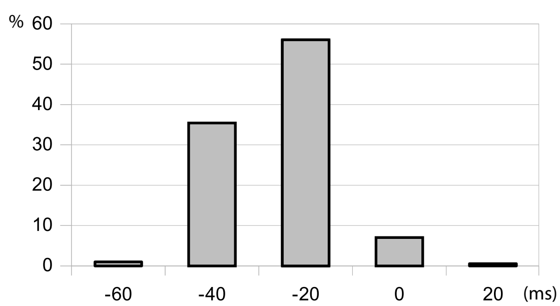

Fig. 3. Histogram of misalignments between force signatures.

- *Normalization*: With the objective of simplifying the learning process, a standard procedure in data-driven applications is to normalize each feature of the signature across examples. For that, we shift the values of each signature $y_i(t)$ as $\tilde{y}_i(t) = y_i(t) - m_y(t)$ where:

$$m_y(t) = \frac{1}{N}\sum_{i=1}^{N} y_i(t) \tag{2}$$

With N being the number of examples.

All results shown in the rest of the paper are based on the data after resampling, alignment and normalization.

IV. EXPERIMENTAL RESULTS

The results presented in this section refer to the assembly of a metallic shield can into a cellphone PCB, Fig. 4. The assembly strategy was designed by hand and involves several open loop motions for alignment of both parts, pressing on the four corners of the shield can and a final vertical *tap*. To evaluate the classifier, we capture the force signatures of 400 assembly attempts at a nominal sampling frequency of 50 Hz.

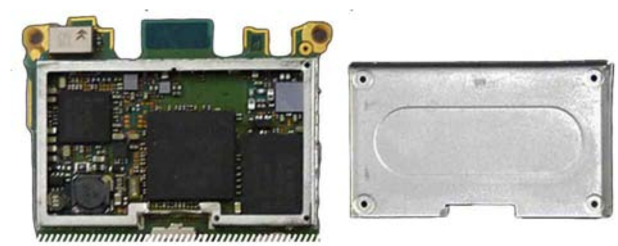

Fig. 4. Parts to mate in the experimental section. The metallic shield can (right) lies on top of the assembly base while the robot firmly holds and manipulates the cellphone PCB (left).

A. Accuracy Analysis

In this section, we aim to evaluate the accuracy of the learning system (percentage of correct predictions of the assembly outcome). We also analyze how the system accuracy changes with the number of examples used in SVM training. For that, we select N examples from the dataset ($\frac{N}{2}$ successful examples and $\frac{N}{2}$ failed ones), train the SVM with them, and then estimate the accuracy using the remaining examples. For consistency across the dataset, we repeat the same experiment several times randomizing the initial selection of the training set and averaging the results.

Fig. 5 shows the change in accuracy while increasing N. The accuracy increases with the amount of training data until saturating at around $N \sim 65$.

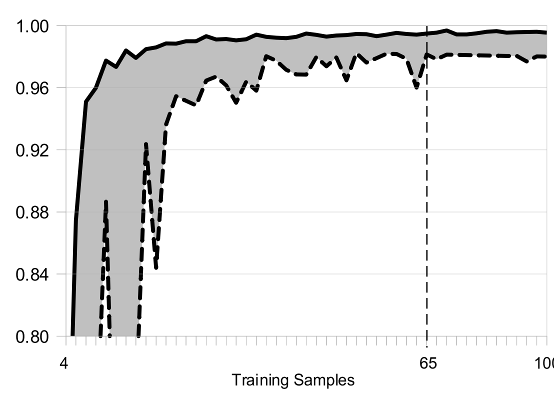

Fig. 5. Average (continuous line) and worst (dashed line) system's accuracy obtained over 100 randomly picked training sets.

B. Most Discriminative Force Direction

In the previous section, the force signature is captured by a 6 axis F/T sensor. In this section, we show that a load cell, which captures force in just one direction, still provides enough information to detect failure accurately.

With the replacement of the force sensor by the load cell, the dimension of the feature space is divided by 6. We will see that if we chose a proper orientation for the load cell, that reduction does not degrade system performance and that it leads to more efficient learning.

First, we find the optimal orientation of the load cell—the orientation that maximizes the system accuracy. That orientation depends on the assembly strategy itself. Intuitively, one would expect that for a roughly vertical assembly, the load cell should be aligned vertically. The next experiment confirms that intuition, and allows us to determine how sensitive the accuracy is to deviations from the optimal orientation.

We simulate the readings of a load cell aligned in any particular direction w of the space by projecting the vector (F_X, F_Y, F_Z) to w. Fig. 6 shows the accuracy obtained as a function of w. We obtain the distribution evaluating the performance of the system along several random orientations of the load cell and a posterior regression of the results. For each orientation we average the results obtained with 100 random training sets of 100 samples each one.

Fig. 6 shows that the optimal orientation of the load cell deviates slightly from the vertical. However, the same analysis shows that statistically significant degradation in performance occurs only after considerable deviation from the optimal orientation. For simplicity, it is ok to assume that, in subsequent experiments, the load cell is vertically aligned.

Fig. 7 shows the results of the same experiment used in Section IV-A but with a vertically aligned load cell. Comparison with the results obtained with the complete force signature (Fig. 5) yield that:

- There are no negative effects in the asymptotic accuracy of the system.
- As a consequence of the reduction of the dimension of the feature space, the system learns faster.

C. Force Signature Compression

In this section, we explore the relationship between the dimension of the feature space and the amount of training examples required to achieve the asymptotic accuracy of the system. For that, we compress the force signature using *Principal Component Analysis* (PCA).

PCA [23] is a common tool used to reduce the dimension of datasets in learning problems. It finds a linear transformation of the data into a smaller number of linearly uncorrelated variables (principal components) while retaining most of the variability of the original data. The first principal component is aligned with the direction of maximum variance in the original dataset. Each successive principal component is

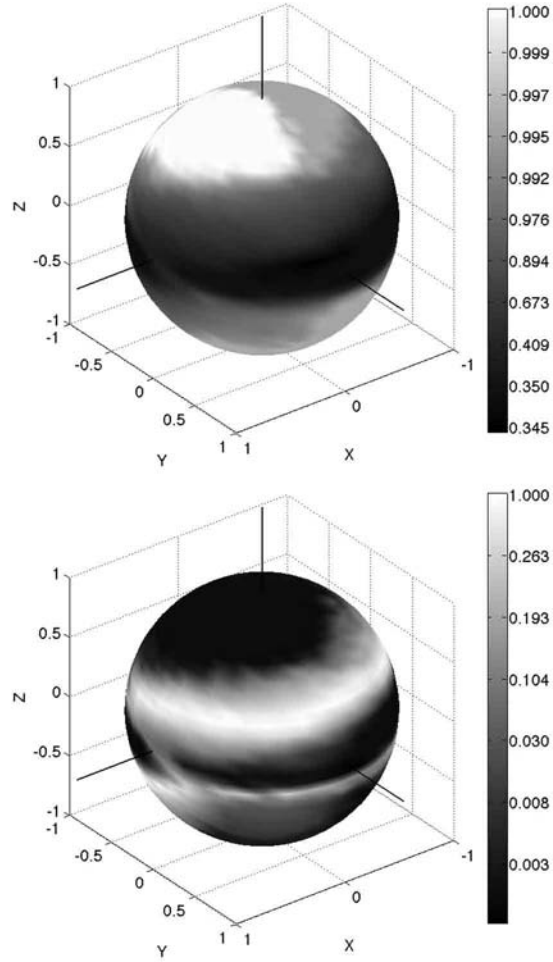

Fig. 6. (Top) Average accuracy as a function of the orientation of the load cell. The system has high accuracy in light regions. (Bottom) Variance of the accuracy across different trainings of the system. The accuracy is stable (low variance) in dark regions. Note that the color scales are not linear.

aligned with the direction of maximum remaining variability not captured by previous components. Ideally, the first variables after the projection should be the most informative for the separation between success and failure.

If we stack all feature vectors x_i in a data matrix X^T, and $U\Sigma V^T$ is the singular value decomposition of X, U^T is the PCA projection matrix and $U^T x_i$ is the projected/compressed feature vector.

In each experiment, the PCA projection matrix is estimated with the samples selected for posterior training of the SVM. This introduces a new tradeoff in reducing the number of required training examples because the fewer training examples we use, the worse will PCA estimate the optimal projection.

In the specific example used in this analysis (shield can + PCB), experiments show that, after PCA projection, approximately all the energy of the original signal is compressed in the first 5 principal components. This is empirical evidence that, from a learning perspective, the real dimension of the problem we are trying to solve is much smaller than the

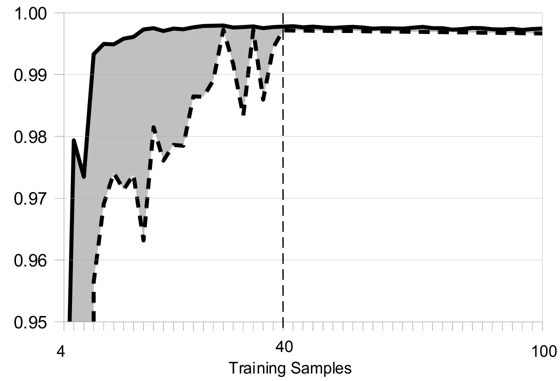

Fig. 7. Average (continuous) and worst (dashed) system's accuracy with force signature captured by a load cell oriented in the Z direction. The results have been averaged over 100 randomly picked training sets.

length of the force signature.

Fig. 8 shows the first three principal components and their relationship to the force signature. The first principal component can be interpreted as the average of the signature during the *contact phase*. The second principal component focuses on the initial contact between the parts and the ending vertical tap. The third principal component finds significant differences between the first and the second half of the signature which correspond to different steps of the assembly strategy. Force signatures of successful and failed assemblies happen to be most different in those events. Intuitively this shows how PCA captures the necessary information to detect failure.

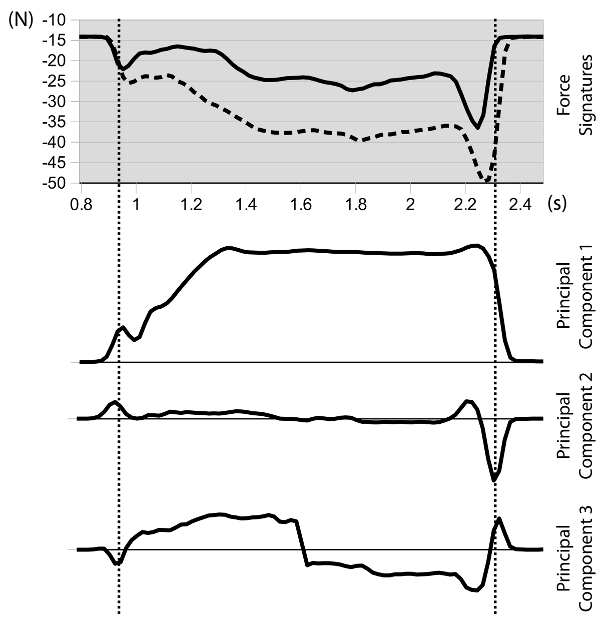

Fig. 8. Example of force signature for correct (continuous line) and failed (dashed line) assemblies and first three principal components of the PCA.

Fig. 9 shows the results of the same experiment proposed in Section IV-A after the PCA projection. We see how, thanks to the compression, the required number of training examples reduces to $N \sim 20$.

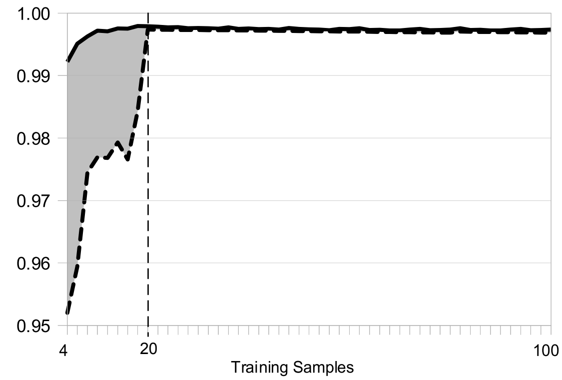

Fig. 9. Average (continuous) and worse (dashed) system's accuracy after the PCA compression. The results have been averaged over 100 randomly picked training sets.

V. FAILURE DETECTION RECIPE

This is a list of the steps to be followed for using the proposed method. Suppose we are given an assembly station, an assembly strategy and that we have access to the force signature of the entire process by means of a force sensing device. The recipe breaks down as:

TRAINING – *offline*

1) The operator executes N assembly trials and hand labels them as success or failure. In the case presented in the paper, analysis shows that 20 tests (10 successful and 10 failed) is enough.
2) After the capture process, all training data is resampled, aligned and normalized.
3) Estimate the optimal PCA projection matrix with all training examples. Force signatures are then compressed. In the presented example, the first 5 principal components are enough to account for most of the information.
4) Use the compressed force signatures and the corresponding hand labels to train the SVM classifier.

TESTING – *online*

5) For every new assembly we go step by step through resampling, alignment, normalization, compression and final classification for detecting success or failure.

Once the system is trained, step 5 is fast enough to allow real time monitoring of the assembly.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

In this work, we have used the force signature of an assembly to detect failure. We have implemented a supervised data-driven approach where captured forces, both from successful and failed assembly attempts, are used to train an SVM that distinguishes between them. Results show high accuracy in failure detection even when no information of the specific assembly is used for training the system.

We have shown that a simple load cell properly aligned provides enough discriminative information. The alignment of the load cell allows for a considerable deviation from its optimal orientation before the effects in the performance of the system are statistically significant.

Finally, we have also shown that compressing the force signature with PCA reduces considerably the amount of required examples for training the system. Fig. 10 compares the average accuracies obtained with the original data, using only the Z component of the force vector and after the PCA compression.

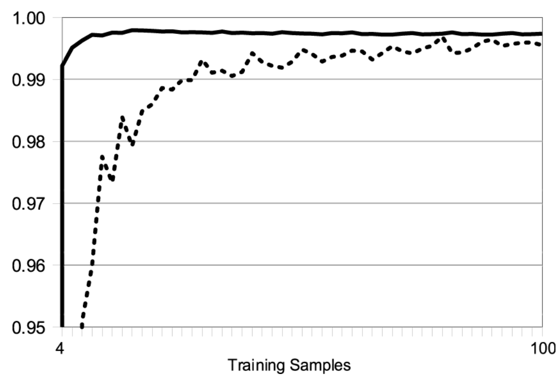

Fig. 10. Comparison of the performance of the failure detection system — (dotted) Original data — (dashed) Just Z component of the force — (continuous) After PCA compression.

Overall, the algorithm detailed in Section V is a flexible methodology for detecting failure in assembly problems that are fixed over time. Changes either in the assembly station or the assembly strategy, would change the force signatures and would require the system to be taught again.

B. Future Work

The next objective is to test the method on a variety of assembly problems. We want to analyze how the amount of required training data varies when changing the complexity of the assembly.

A standard procedure when dealing with failures in dynamical systems is to distinguish among a small set of failure modes and build a tree of actions. A straightforward way of doing that is, during the training phase, to ask the human operator to provide feedback on the type of failure and train a classifier to detect both the presence of failure and predict the failure mode.

However, naming causes or failure modes is not necessarily easy for a human operator. We would like to explore a less supervised solution where, during training of the system, the operator provides binary feedback (success/failure) at the end of the assembly, and failure modes are discovered.

We have shown how an SVM can be trained for classifying force signatures as success or failure. The same way, an structured SVM can be trained to detect what parts of the force signature are likely to relate with a correct or failed assembly. In that formulation, when facing a failed assembly, we should be able to detect when things begin to go wrong and then discover and identify failure modes.

VII. Acknowledgments

The authors would like to thank ABB Corporate Research and in particular Thomas A. Fuhlbrigge for providing support for this research. Alberto Rodriguez is the recipient of a graduate fellowship sponsored by Caja Madrid.

References

[1] J. Krishnasamy, M. Jakiela, and D. Whitney, "Mechanics of vibration-assisted entrapment with application to design," in *Proceedings of IEEE ICRA*, no. April, Minneapolis, 1996, pp. 838–845.

[2] G. Boothroyd, P. Dewhurst, and W. A. Knight, *Product Design for Manufacture and Assembly*, 2nd ed. CRC Press, 2001.

[3] H. Hitakawa, "Advanced parts orientation system has wide application," *Assembly Automation*, vol. 8, no. 3, pp. 147–150, 1988.

[4] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *Annals of the CIRP*, vol. 48, no. 2, pp. 527–540, 1999.

[5] L. Camarinha-Matos, L. Lopes, and J. Barata, "Integration and learning in supervision of flexible assembly systems," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 2, pp. 202–219, April 1996.

[6] A. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[7] V. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, October 2004.

[8] M. El Hachemi Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 984–993, 2000.

[9] A. Smyth, S. Masri, T. Caughey, and N. Hunter, "Surveillance of Mechanical Systems on the Basis of Vibration Signature Analysis," *Journal of Applied Mechanics*, vol. 67, no. 3, p. 540, 2000.

[10] S. Doebling, C. Farrar, M. Prime, and D. Shevitz, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review," Los Alamos National Laboratory, 1996.

[11] S. Barai and P. Pandey, "Vibration signature analysis using artificial neural networks," *Journal of Computing in Civil Engineering*, vol. 9, no. 4, pp. 259–265, 1995.

[12] T. I. El-Wardany, D. Gao, and M. A. Elbestawi, "Tool condition monitoring in drilling using vibration signature analysis," *Int. Journal of Machine Tools and Manufacture*, vol. 36, no. 6, pp. 687–711, 1996.

[13] 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.

[14] Y.-W. Hsueh and C.-Y. Yang, "Prediction of tool breakage in face milling using support vector machine," *The Int. Journal of Advanced Manufacturing Technology*, vol. 37, no. 9-10, pp. 872–880, 2007.

[15] D. M. Tax, A. Ypma, and R. P. Duin, "Support Vector Data Description applied to Machine Vibration Analysis," in *ASCI'99*, Heijen, Netherlands, January 1999.

[16] 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," *Journal of Mechanical Engineering Science*, vol. 222, no. 6, pp. 1081–1095, 2008.

[17] D. M. Fullmer, "Parts assembly Using Signature Analysis," US Patent 4,855,923, August 1989.

[18] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[19] C. Cortes and V. Vapnik, "Support-Vector Networks," in *Machine Learning*, 1995, pp. 273–297.

[20] T. Joachims, *Making large-scale SVM learning practical*. Cambridge, MA, USA: MIT Press, 1999, ch. Advances in kernel methods: support vector learning, pp. 169–184.

[21] W. S. Cleveland and S. J. Devlin, "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, vol. 83, pp. 596 – 610, 1988.

[22] S. Schaal and C. Atkeson, "Robot juggling: implementation of memory-based learning," *Control Systems Magazine*, vol. 14, pp. 57–71, 1994.

[23] L. Smith, "A tutorial on principal components analysis," Cornell University, 2002.