Data-driven Classification of Screwdriving Operations

Reuben M. Aronson, Ankit Bhatia, Zhenzhong Jia, Mathieu Guillame-Bert, David Bourne, Artur Dubrawski, and Matthew T. Mason

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
`{rmaronson,ankitb,zhenzjia}@cmu.edu,mathieug@andrew.cmu.edu,`
`{db,awd,matt.mason}@cs.cmu.edu`

Abstract. Consumer electronic devices are made by the millions, and automating their production is a key manufacturing challenge. Fastening machine screws is among the most difficult components of this challenge. To accomplish this task with sufficient robustness for industry, detecting and recovering from failure is essential. We have built a robotic screwdriving system to collect data on this process. Using it, we collected data on 1862 screwdriving runs, each consisting of force, torque, motor current and speed, and video. Each run is also hand-labeled with the stages of screwdriving and the result of the run. We identify several distinct stages through which the system transitions and relate sequences of stages to characteristic failure modes. In addition, we explore several techniques for automatic result classification, including standard maximum angle/torque methods and machine learning time series techniques.

Keywords: screwdriving, automation, assembly, manufacturing

1 Introduction

Screwdriving is one of the most common assembly operations, yet automating the process has remained difficult despite substantial effort [1]. Since the operation is so prevalent, even a one percent failure rate in screwdriving can generate tens of thousands of bad products; yet, a mechanical process will never be flawless. Therefore, our goal is to instead enable the mechanical system to automatically identify failures and recover from them. Thus, the system can be made more robust without additional, more expensive mechanical changes such as tighter tolerances. In addition, a better understanding of failure classes and of the overall screwdriving process can motivate alternate strategies for improving accuracy.

In order to explore this strategy for enhancing robustness, we collected multimodal, time-synchronized data sets on 1862 screwdriving operations. Using this data, we identified several stages through which a screw can pass and the success or failure case to which they correspond. In addition, we explored techniques for using the sensor data to predict the result of a screwdriving operation. With this dataset, we can provide a deeper understanding of the failure cases for screwdriving and provide a structure for a fault detection system.

1.1 Related Work

Among common assembly methods, screwdriving has been especially difficult to fully automate [2] due to our incomplete understanding of the underlying process, particularly the initial mating step [3]. A comprehensive review of screwdriving, including theoretical fundamentals, tools, control strategies, failure detection, and industrial applications, is available in several references [4], [5], [1].

Simpler characterizations of screwdriving, such as the torque-angle curve (or signature) [4], are commonly used to identify the screwdriving process. It has been used for ISO rotary tool evaluation standards [6], screwdriving control [7] and failure detection [8], [9], [10] and [11]. Other efforts, such as Giannoccaro *et al.* [11], have used an online data acquisition approach to fit a theoretical model of the screwdriving process. However, we know of no data sets that have been collected or analyzed to the complexity of the data presented here.

2 Data Collection

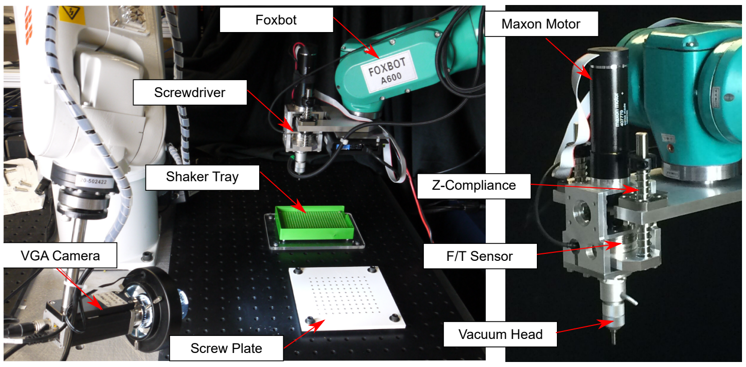


Fig. 1. The instrumented screwdriver station used for data collection.

We have designed an instrumented vacuum screwdriver (Fig. 1) to measure relevant process parameters. The screwdriver floats on a Z-axis compliant stage, which holds a motor (Maxon RE-30 with a 1-stage gear) mounted on a six-axis force-torque sensor (ATI mini40). The motor shaft passes through a bearing into a sealed chamber and terminates in a Phillips bit. The sealed chamber connects to the vacuum head, which conforms to the screw head. To acquire a screw, the sealed chamber is lowered over a screw located in the screw feeder and an external vacuum source is turned on. The vacuum pulls the screw head against the bottom of the vacuum head. Once the screw is lowered into its hole, the bit drops to completely engage the screw.

The screwdriver is mounted on an industrial manipulator (Foxconn Foxbot A600). A shaker tray to hold screws and an aluminum plate with threaded inserts are installed in the manipulator's workspace; the plate is first calibrated on a coordinate measuring machine to ensure that the manipulator can precisely align the screws to the holes.

During each run, the screwdriver first picks up a screw from the shaker tray using the vacuum system. Then the motor turns on, the manipulator moves the screwdriver to a specified height above the hole, and data collection begins as the screw is lowered into the hole. During the run, six-axis force-torque data, motor current and speed, robot position, and video (using a camera mounted on a second manipulator) are all collected. The run concludes when either the motor current or the motor angle reaches a specified limit. Finally, the screwdriver returns to its starting height and proceeds to the next run.

For about half of the collected data, the process above was unaltered. For a quarter of it, a position offset sampled from a Gaussian distribution was introduced between the screw and hole axes. For the final quarter of the data, a Gaussian angular disturbance was introduced, and the whole apparatus was rotated about the screw tip to simulate an axial misalignment with the plate.

We collected a total of 1862 screwdriving runs. Each run consists of six axes of force and torque, motor current and speed, and video data, all synchronized in time. In addition, each run has been hand-labeled with the stages through which the operation progresses, as determined by subsequent review of the data, and with its result class (see below for detail). This dataset forms the largest collection of screwdriving data that we know of, and it provides a variety of insights into how the screwdriving process works.

3 Analysis

Examining the captured force-torque signature and video data, we empirically came up with a list of stages and result classes. The result classes correspond with outcomes common in previous studies [4] [12] with additional classes added to represent less-studied outcomes. These stages required human judgment to identify, and alternate characterizations are possible; however, they provide useful information for understanding how the screwdriving operation proceeds. The result categories are listed in Table 1, and the stages in Table 2. Conceptual force-torque signatures for a few examples are shown in Figure 2.

One way to visualize the stages and results of the collected data is by considering the state transition graph, which appears in Figure 3. Vertices represent stages through which the screwdriving operation passes, with the terminal states corresponding to result classes. Vertices and edges are weighted, colored, and sized according to the number of runs that pass through them. From the figure, it is clear that all runs begin at the *approach* stage. The successful runs then proceed through *initial mating*, *rundown*, and *tightening*; some of them pass through *hole finding* either before or in place of *initial mating.* The failures, in contrast, are highly varied in their stages. The most common failure, the failure to acquire the screw for insertion before the process even begins, goes only to the *no screw spinning* stage before completion. This abnormally high pick-up failure occurs because the vacuum adapter in our screwdriver is not optimized

Table 1. The result classes used when classifying each screwdriving run, along with descriptions and percentage of total runs with the corresponding label.

Result	**Description**	**Freq**
Success	Screw successfully driven into hole and tightened	84.4%
No screw	Failed to acquire screw	9.7%
No hole found	Screw acquired but never dropped into hole	2.6%
Crossthreaded	Screw entered hole but threads crossed so current limit hit before rundown completed	1.8%
Stripped	Screw successfully run down into hole but bit slippage prevented full tightening	0.8%
Partial	Screw driven partly into hole but time limit reached before operation completed	0.3%
Stripped (no engage)	Screw was so stripped it never engaged hole	0.4%

Table 2. The stage labels used when classifying each screwdriving run, along with descriptions and percentage of total runs with the corresponding stage. As each run passes through multiple stages, stage percentages do not sum to 100%.

Stage	**Description**	**Freq**
Approach	Screwdriver approaches and touches plate	100.0%
Hole finding	Screw has touched plate but not yet fallen into hole.	45.1%
Initial mating	Driver moving down and fully engaging screw	87.0%
Rundown	Screw is engaged and spinning into hole	86.4%
Tightening	Driver applies torque to tighten screw against plate	78.3%
No screw spinning	Bit is spinning with no screw present	9.8%
Screw fallen	Screw has fallen off of bit	1.6%
Stripped engaging	Screwdriver attempts to engage screw but bit slips against screw head.	0.3%
Stripped rundown	Screwdriver attempts to run down screw but bit slips against screw head.	0.4%
Stripped tightening	Screwdriver attempts to tighten screw but bit slips against screw head	0.8%

for the screws used in the experiments. Another important feature to note is that, except for the *crossthread* failure, all result cases pass through distinct stages before completion. Thus, a complete list of the stages that the operation undergoes is almost completely sufficient to predict the result, suggesting that it is possible to build a failure prediction and avoidance system to anticipate impending failures and take appropriate behavior to avoid them.

In addition, the stages provide a deeper understanding of the underlying operation, which can be applied to identify process failures that would be missed through simple result classification. For example, the *hole finding* stage, in which the screw walks on the part searching for the hole, may be problematic since it damages the finish near the hole; however, a simple result classifier such as that described in the next section is unable to differentiate between successful runs that include a *hole finding* stage and those that do not. Furthermore, examining this stage can provide information about the mechanical mating procedure

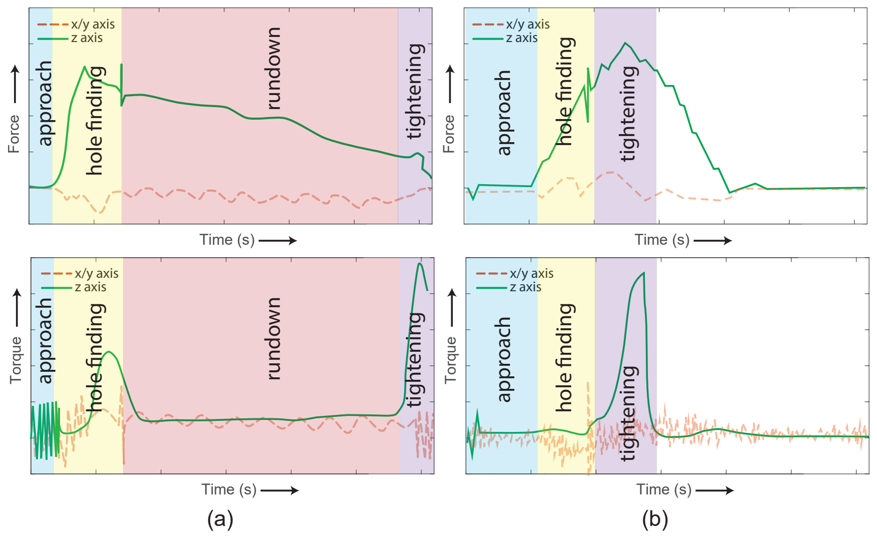


Fig. 2. Conceptual force-torque signatures of (a) successful and (b) unsuccessful (cross-threaded) screwdriving runs.

between the screw and the hole, such as the effect that the compliance of the screwdriver has in the screw mating procedure; additional work is required to explore this relationship.

4 Failure Detection

4.1 Basic Failure Detection

The simplest method of classification, and the most common method, uses two features: the final torque applied to and angle traveled by the screw at the time of cutoff [4] (see also [8], which uses the maximum insertion force rather than the driving torque). To provide a benchmark for advanced strategies, we started by classifying the data using only these features. The maximum angle was determined by the encoder counts from the end of the "approach" phase to the cutoff point, and the corresponding torque was determined by linearly interpolating the torque curve at the time of the cutoff.

We classified the data using multivariate logistic regression. With 10-fold cross-validation, this method produced an overall error of $e = 0.03$; with a training error $e_{train} = 0.03$, the overfitting is negligible. A scatter plot, with decision bounds marked, and the confusion matrix for this classifier appear in Figure 4. This classifier works well for identifying several important result classes. It identifies *success* and *crossthread* nearly perfectly, and it also does moderately well at *noscrew*. Since the method so accurately detects the success case, it is an

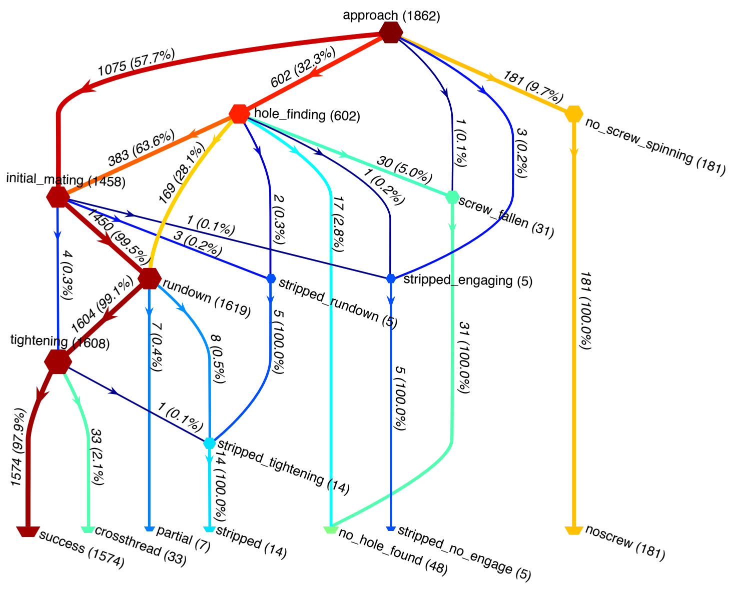
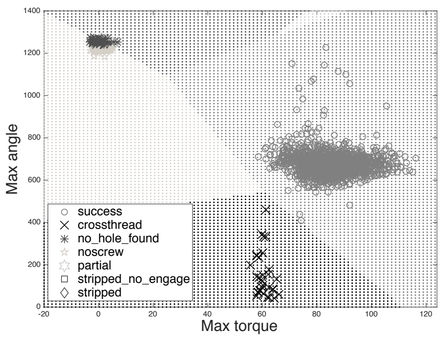


Fig. 3. State transition graph summarizing the stages and results of all data. Colors and sizes are scaled logarithmically with the number of runs in each transition, stage, or result.

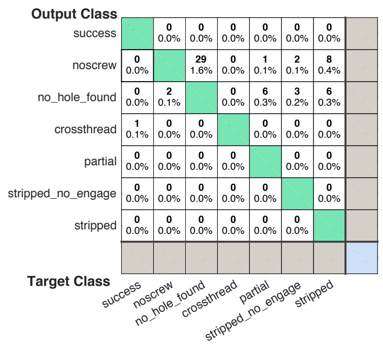
appropriate method to use when the mode of failure is irrelevant. However, this method works poorly when distinguishing among failure cases, especially those that are not *crossthread*. The scatter plot indicates that this problem is not the fault of the classifier choice; while *success*, *crossthread*, and to a lesser extent *noscrew* are all linearly separable, the rest of the failure cases are all closely grouped together in this feature space. In fact, the classifier has a 0% success rate when classifying the more obscure result categories. In order to distinguish between these failure cases, we must use more sophisticated features.

4.2 Multivariate Temporal Models for Failure Detection

We have also tested the utility of a more sophisticated machine learning approach to identify and predict failures. For this, we relied on eight time series (3-axis measurements of each of the force and torque, as well as 1D measurements of motor current and motor speed), each sampled at 100Hz. We first preprocessed the data using the Honey [13] tool. For each time series, we derived time series of simple statistics including simple moving average (sma), moving standard deviation (sd) and moving range of values (range) using time windows of aggregation 0.1, 0.2, and 0.3 second wide. We also extracted "peak events" from each of the raw series, as well as their corresponding moving averages, moving standard deviations, and moving ranges. A peak is defined as an instantaneous event occurring when the first order time derivative of a signal (estimated with trian-



(a) Scatter plot of data overlaid on logistic regression classification regions



Output Class	success	noscrew	no_hole_found	crossthread	partial	stripped_no_engage	stripped
success	🟩	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
noscrew	0 0.0%	🟩	29 1.6%	0 0.0%	1 0.1%	2 0.1%	8 0.4%
no_hole_found	0 0.0%	2 0.1%	🟩	0 0.0%	6 0.3%	3 0.2%	6 0.3%
crossthread	1 0.1%	0 0.0%	0 0.0%	🟩	0 0.0%	0 0.0%	0 0.0%
partial	0 0.0%	0 0.0%	0 0.0%	0 0.0%	🟩	0 0.0%	0 0.0%
stripped_no_engage	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	🟩	0 0.0%
stripped	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	🟩

Target Class

(b) Confusion matrix for classifier results

Fig. 4. Results of logistic regression using maximum torque and total rotation angle

gular kernel) changes its sign. Depending on the sign change direction, a peak can be "up" (change of trend from positive to negative) or "down" (negative to positive). Each peak event is labeled with a time stamp, its direction, and the value of the second order time derivative of a signal to reflect the rate of trend change (informally reflecting the "strength" of a peak). Such peak features are simple but often effective at representing dynamic aspects of temporal data. The resulting featurization includes the raw input time series, their moving averages, standard deviations, moving ranges, and two types of peak events, combining into 144 temporal features in total.

We use the Graph of Temporal Constraint Decision Forest (GTC-DF) algorithm [14] for the classification of the screwdriving runs. GTC-DF is a machine learning algorithm designed for classification of Symbolic and Scalar Time Sequences (SSTS), an extension of the familiar multivariate time series paradigm to include asynchronous discrete events in the representation of data. Unlike conventional machine algorithms that require temporal data to be featurized and transformed into a transactional form of a flat table with rows representing the subsequent discrete time stamps and columns representing various features derived from data at the corresponding points in time, GTC-DF can consume time sequence data directly and learn models that reflect the temporal structure of data in a more natural fashion. At its core, GTC-DF infers a decision-tree–like structure from data, each node of which represents a specific Graph of Temporal Constraints [14].

To ensure robustness of the reported results, and for consistency with the basic classification above, we used 10-fold cross-validation to score performance of GTC-DF. Of the 1862 runs, it was able to correctly classify 1844 runs as success or failure (average error rate of 0.0097), and 1841 when predicting the exact type of failure (0.0113 average error rate). The obtained AUC (Area Under the ROC [Receiver Operating Characteristic] Curve) ranges between 0.99920 and 0.99992 with 95%-ile confidence. These results yield a 3-fold improvement when compared to the basic classification performance reported above, and illustrate the value of using a natively temporal approach to model temporal sequence data. Training the GTC-DF model took on average 15m4s on an 8-core i7-3770 3.40GHz computer with 16 GB of main memory. Once GCT-DF is trained, applying it takes an average of 7ms per screwdriving run.

Trained GTC-DF models may contain hundreds of decision tree-like structures and thousands of temporal constraints graphs, making them impossible to fully interpret by a human user. But in practice, we can often heavily constrain the complexity and select only a few representative components of the GTC-DF collection inferred from data to avoid overwhelming the end users, while maintaining useful model accuracy.

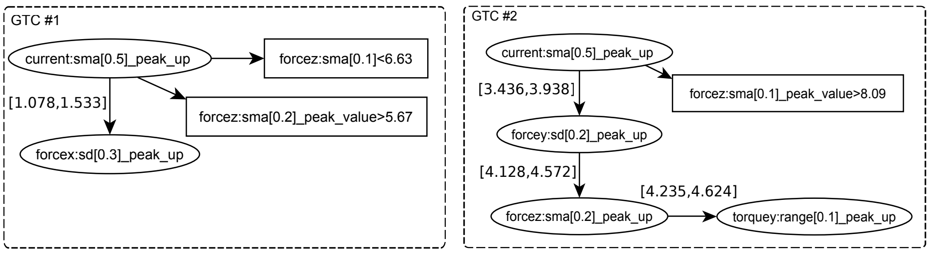


Fig. 5. Two examples of GTC models inferred from data under limited complexity. The specific model design and parameters indicated here were determined algorithmically to be a simplified model that aligns most closely to the results of the full model.

Table 3. Classification results on test data using the pair of GTCs shown in Figure 5. Cells summarize distribution of failures and successes (x%:y%) and percentage of runs covered (in italics).

	Not GTC #2	**GTC #2**
GTC #1	100%:0%	0%:100%
	7.487%	*2.674%*
Not GTC #1	0%:100%	50%:50%
	88.77%	*1.067%*

Figure 5 shows two example GTC models of purposely restricted complexity. Each of them defines a Boolean condition on time series: it is true if the model matches current data, and false otherwise. Oval nodes in the diagrams represent existential conditions on the events. Edges between oval nodes represent temporal constrains between them. Rectangular nodes reflect inequality conditions on time series values at the time of their parent oval node application. Label X:Y of each node denotes a raw or a derived signal, where X denotes the sensor type (current, speed, force or torque), and Y reflects the type of preprocessing applied to it. For example, "current:sma[0.3]_peak[up]" refers to a peak in the 0.3s simple moving average of the "current" time series. More explicitly, GTC #1 in

Figure 5 evaluates as true if at time t during a run "*the 0.5s moving average of motor current peaks up at time t, followed, between 1.078s and 1.53s, by a peak up of the 0.3s standard deviation of the force measured along the x-axis, and if the moving average of the force along the z-axis over the past 0.1s stays below 6.63 at time t, and if the strength of the last peak of the 0.2s moving average of the force along the z-axis measured at t is greater than 5.67.*" Note that all the numeric parameters represented in the graph have been automatically inferred from data by the GTC-DF algorithm. Note also that the resulting model is truly multivariate: the example graph simultaneously uses motor current and two vectors of force, via their specific statistics that have been found informative by the learning algorithm. Finally, note that the model can be read in plain English and it represents a fully interpretable logical statement of reasonable complexity.

Table 3 shows classification of screwdriving runs represented by the four possible combinations of the two example GTC models. This table only reflects the test data runs (training data not included in statistics). If the GTC #1 is matched by the current data but the GTC #2 is not (upper left cell of Table 3), this always yields a successful screwdiving run. This logical combination of applicability of GTCs #1 and #2 covers almost 7.5% of runs. Similarly, we can read that if both GTCs match the data, all such runs are failures and this reflects 2.674% of all data (upper right cell in Table 3). Whenever neither of the GTCs is matched, we get another clean cut determination of the run outcome (lower left cell), but when #2 is matched while #1 is not, we observe a 50-50 split of possible outcomes (lower right cell). Note that even though the only confusing combination of applicability of the two example GTCs applies to a very small fraction of all data, this data can be further disambiguated by including additional GTCs into the set. Empirical 10-fold cross validation performance of the simplified model yields an error rate of 0.0118 which is slightly higher than that of the full model (0.0097) but still substantially lower than the simple classification model presented in Section 4.1 (0.03).

5 Conclusions and Future Work

In this research, we collected a large data set of screwdriving operations to identify failure cases and stages of the process. Using this data, we have demonstrated that the collected signatures can be used to classify the result a run with high accuracy using a sophisticated learning strategy. Accurately classification of failures enables us to pursue a fully integrated failure detection and recovery system, with recovery strategies dependent on the specific failure case. Our overall classification accuracy was near 99%, significantly improving upon the baseline success rate of 84.4% for our setup and suggesting that failure detection and recovery can indeed improve industry manufacturing systems.

The work here also suggests additional strategies for improving the failure detection and recovery system. The state diagram in Figure 3 shows that it may be possible to predict incipient failures and prevent them before they occur. Augmenting the classification system to model prediction confidence along with failure type may improve the overall accuracy of the system. Additionally, the understanding of the underlying screwdriving process provided by the stages and failures identified here can motivate future development of the mechanical screwdriving device. We also intend to test the robustness of the stage and result lists and classification strategies presented here by collecting additional data sets with different setups, varying screw size, compliance, and other factors. All of these results will enable enhanced robustness of automated screwdriving systems for industrial use.

6 Acknowledgments

The authors gratefully acknowledge the financial support from Foxconn.

References

1. Z. Jia, A. Bhatia, R. Aronson, D. Bourne, and M. T. Mason, "A survey of automated threaded fastening," in preparation.
2. E. J. Nicolson, "Grasp stiffness solutions for threaded insertion," Master's thesis, University of California, Berkeley, 1990.
3. D. E. Whitney, *Mechanical assemblies: their design, manufacture, and role in product development.* Oxford University Press, 2004, vol. 1.
4. J. H. Bickford, *Handbook of bolts and bolted joints.* CRC press, 1998.
5. ——, *Introduction to the design and behavior of bolted joints: non-gasketed joints.* CRC Press, 2007.
6. ISO, "ISO 5393: Rotary tools for threaded fasteners – performance test method," ISO, Tech. Rep., 2013.
7. J. Boys and P. Wallace, "Design and performance of an automatic control system for fastener tightening," *Proceedings of the Institution of Mechanical Engineers*, vol. 191, no. 1, pp. 371–380, 1977.
8. T. Matsuno, J. Huang, and T. Fukuda, "Fault detection algorithm for external thread fastening by robotic manipulator using linear support vector machine classifier," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 3443–3450.
9. R. Chumakov, "An artificial neural network for fault detection in the assembly of thread-forming screws," *Journal of Intelligent Manufacturing*, vol. 19, pp. 327–333, 2008.
10. 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, pp. 1081–1095, 2008.
11. N. I. Giannoccaro and M. Klingajay, "Online Identification for the Automated Threaded Fastening Using GUI Format," in *Cutting Edge Robotics*, V. Kordic, A. Lazinica, and M. Merdan, Eds. Germany: Pro Literatur Verlag, Jul. 2005, pp. 727–745.
12. *MicroTorque-ToolsTalk MT User Guide*, Atlas Copco.
13. M. Guillame-Bert. (2016) Official home of the Honey programming language. http://framework.mathieu.guillame-bert.com/. Last accessed 06 June 2016.
14. M. Guillame-Bert and A. Dubrawski, "Classification of time sequences using graphs of temporal constraints," *Journal of Machine Learning Research, in review*, 2016.