Towards Automatic Assembly of Small Screws: Failure Detection and Stage Classification

Xianyi Cheng

CMU-RI-TR-19-69

August 2019

The Robotics Institute

Carnegie Mellon University

Pittsburgh, PA 15213

Thesis Committee:

Matthew T. Mason

Ralph Hollis

Ankit Bhatia

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Hundreds of billions of small screws are assembled in consumer electronics industry every year, yet reliable screwdriving automation remains one of the most challenging tasks. Barriers that prevent further adoption of robotic threaded fastening systems are system cost and technical challenges, especially those pertinent to small screws. Smaller screws require tighter tolerance and higher alignment accuracy, which increases the system cost and failure rate. A solution is to close the loop. An affordable intelligent screwdriving system that can support online stage classification and failure prediction is the first step to automating the assembly of small screws. However, existing failure prediction techniques are simple and they perform poorly. In addition, most solutions are essentially data-driven, thereby requiring lots of training data and laborious labeling. Moreover, they are not robust against varying environment conditions and suffer from generalization issues. To this end, we propose a stage and result prediction framework that combines knowledge-based process models with a hidden Markov model. The novelty of this work is the incorporation of operation-invariant characteristics such as screwdriving mechanics and a stage transition graph, enabling our system to generalize across a range of experimental settings and largely reduce the requirements on data and labeling. In our experiments, a system trained on M1.4x4 screws adapts with very little non-labeled data to three other screw sizes (M1.2x3, M2.5x5, and M1.4x4) with varying tightening current, motor velocity, insertion force, and tightening force.

We also discuss the role of sensor reduction and compliance in screwdriving automation. To that end, we present our preliminary work on sensor reduction. We use statistical methods to select proper sensors and produce affordable intelligent screwdrivers that can be deployed in industry. Our preliminary experiments also show that compliance allows screws to be inserted with larger misalignment errors.

Acknowledgments

I would like to first thank my advisor, Matt Mason, for giving me the chance to start my research in robotics in such a cool lab. I learned a lot from him and my labmates. I would also like to thank Zhengzhong Jia and Ankit Bhatia, my collaborators in this Foxconn Screwdriving project, for always being there for me when I have technical problems. I also want to thank Ralph Hollis for being in my committee and giving me helpful feedbacks. Finally, I am grateful to my family for their love and support.

Funding

The authors would like to thank Foxconn for providing financial support. The views and opinions expressed in this paper do not necessarily state or reflect those of Foxconn.

Contents

1 Introduction **1**

2 Related Work **5**

3 Robotic Screwdriving System **11**

3.1 Robotic Screwdriving System 11

3.2 Robotic Screwdriving Operation Procedure 11

4 Screwdriving Dataset **15**

4.1 Overall Description 15

4.2 Screwdriving Operation Parameters 16

4.3 Dataset Structure 18

4.4 Downloads 18

5 Stage Classification and Result Prediction **19**

5.1 Modeling of the Screwdriving Process 19

5.1.1 State Transition Graph 19

5.1.2 Feature Extraction 20

5.1.3 Stage Models 21

5.2 Stage Prediction by Hidden Markov Model 23

5.2.1 HMM Representation 24

5.2.2 Stage Classification 24

5.2.3 Data Adaptation 25

5.2.4 Anomaly Detection 26

5.3 Rule-Based Result Prediction 26
5.4 Generalization 27
5.5 Experiments 27
5.5.1 Dataset and Implementation 27
5.5.2 Results 28
5.5.3 Limitations 31

6 Discussion **33**
6.1 Sensor Reduction 33
6.1.1 Data for Sensor Reduction 33
6.1.2 Sensor Reduction based on Result Classification 34
6.2 The Role of Compliance in Screwdriving 39

7 Conclusion **43**

Bibliography **45**

List of Figures

1.1 Our online stage estimation and result prediction pipeline. 3

2.1 Torque-angle curve during the screwdriving process [1] 6

2.2 A stage transition graph summarizing possible stages and results in the screwdriving dataset from [2] 6

3.1 Intelligent robotic screwdriving system for data collection. 12

3.2 An example of collected raw sensor data. The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis. 12

4.1 Screwdriver bits, from left to right: PH#00-1.18, PH#00-1.38, and PH#1-1.95 . . 16

4.2 Left: rigid-bit, right: flex-bit. The red circle points out the cut slot that makes the screwdriver bit flexible when fixed with a small screw. 16

4.3 A cover plate on a screw hole plate 17

4.4 a screw hole plate installed on our compliant x-y plate 18

5.1 The stage transition graph summarizes all possible stages in the screwdriving process. 20

5.2 Illustrative samples of stage classification selected from our dataset. Comparison of the stage predictions before and after data adaptation on high precision operation data and error introduced operation data. Different color blocks correspond to different stages. Only the force sensor data are plotted due to space limitations. The actual result for each operation is shown on the top of the figure. 29

5.3 Anomalies detected by our system. The first row: the force profiles. The second row: the torque profiles. The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis. . . 30
5.4 The visualization of the generalization of our system on *dataset 2* (force and torque profiles), *dataset 3* and *dataset 4* (force profiles only). The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis. 32

6.1 The 6-axis force and torque signatures of (a) successful case with *hole finding* stage and (b) an unsuccessful (*crossthread*) case. 34
6.2 Left: The state transition graph summarizing all the stage and result classes. The vertices represent stages through which the screwdriving passes, with the terminal stages corresponding to the result classes. Colors and sizes are scaled logarithmically with the number of runs in each transition, stage, or result (modified from [2]). Right: A successful screwdriving run example including a *hole finding* stage. 35
6.3 Confusion matrix for the LDA model trained with 85 features. 36
6.4 Prediction accuracy for optimal feature subsets of different sizes. 38
6.5 Stacked bar graph during the feature reduction process. 38
6.6 Confusion matrices for LDA models trained with 20 (left figure) and 18 (right figure) global features, respectively. Result type numbering: 1.*success*; 2.*no screw*; 3.*no hole found*; 4.*crossthread*; 5.*stripped*; 6.*stripped no engage*; and 7.*partial*. 39
6.7 our two-axis variable stiffness actuator plate for active compliance control 41
6.8 Parallel force and position control scheme 42

List of Tables

4.1 Description of Operation Parameters 17

5.1 Data collection settings 27
5.2 Result Accuracy 31

6.1 Optimal features for $\{T_z, M_c, F_z, \theta\}$ signals 40
6.2 Optimal features for $\{M_c, F_z, \theta\}$ signals 40

Chapter 1

Introduction

Threaded fastening is one of the most commonly used methods in industrial assembly [3]. Over a quarter of typical assembly operations can be classified as bolt or nut insertions [4] [5] [6]. Even then, screwdriving remains a difficult task to automate despite substantial research in this field. Our incomplete understanding of the underlying process, particularly the initial mating step [7] is one of the reasons. Our survey paper [3] summaries various open problems and barriers that need to be conquered for automated screwdriving systems. First, fast and reliable ways to feed screws with smaller length-to-diameter aspect ratios need to be developed. Second, strategies for fast and reliable initial thread mating and early fault detection need to be developed. Third, interactions of multiple objects (screw, driver bit, vacuum adapter, and target) need to be investigated. Fourth, online failure prediction and fault recovery algorithms need to be developed to improve the overall success rate.

Automated screwdriving becomes even more challenging when it comes to the consumer electronics industry (for example laptops, tablets, and smartphones), where hundreds of billions of small screws ($\leq$ #4 or $\leq M3$) are assembled every year [8]. In fact, it is one of the most challenging operations that prevent manufacturing enterprises from further adopting automated robotic systems [8]. Errors in screwdriving can cause serious consequences. For example, a loose screw can damage the battery inside a laptop, causing overheating and posing a fire hazard [9].

Small screws introduce additional challenges and design requirements [10] [3]:

- Tighter tolerances for screw feeding and acquisition;

- Higher positioning accuracy and improved locating strategies (e.g., visual servoing) to correct position and angular misalignments;
- Fully functional (supporting online fault prediction and recovery), accurate yet affordable screwdrivers.

Automated screw fastening involves multiple steps, often including screw feeding and acquisition, alignment, screwdriving, and post-fastening steps [3]. Comprehensive reviews of threaded fastening, including theoretical fundamentals, tools, control strategies, failure detection and industrial applications, can be found in [11], [12], and [3]. Among various steps, most works focus on the screwdriving process, where a properly acquired screw is driven into the target hole. One way to understand the screwdriving process is by plotting applied torque against the total rotation angle. This *torque-angle curve* [3] is often used for ISO rotary tool evaluation standards [13], control strategies, and failure detection [2].

Unlike most existing work which focus on big screws, our work focuses on the miniature screwdriving process. As a summary of the above mentioned challenges, smaller screws require tighter tolerance and higher alignment accuracy [7].

The screwdriving process can be further divided into three major sub-steps: *initial thread mating*, *rundown*, and *tightening* with some variations or extra steps for self-tapping screws [3].

Moreover, compared to the well-studied peg-in-hole problem, screwdriving has more process stages and failure modes with complex mechanics involving multiple discrete contact events [14]. A system capable of online process monitoring, failure prediction and recovery is necessary for highly automated solutions [15]. However, existing work is still preliminary. Most of the previous work can only perform result classification given known failure modes, which alone cannot detect irreversible process failures [2] or unknown failures. Moreover, previous methods are not guaranteed to work when experimental conditions (for example screw sizes) change.

Due to difficulties in screwdriving process modeling, most failure detection systems are essentially data-driven. Learning from data directly yields good results. However, this approach often needs lots of training data and laborious expert labeling [2]. Acquiring large amounts of data is easy in industry. However, collecting a large dataset that covers various failure modes is extremely time-consuming and requires extra engineering effort, mainly due to the "long tail"

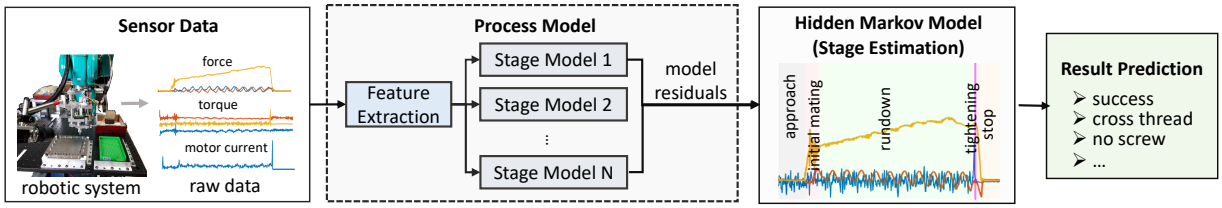

Figure 1.1: Our online stage estimation and result prediction pipeline.

effect [16]. Most failure rates are less than $1\% - 2\%$. So how can we minimize human effort needed for data labeling and avoid recollecting data when experiment setup changes?

To address these robustness and data-efficiency issues, we propose a system (Fig. 1.1) that combines process knowledge with a learning method for stage and result classifications in an unsupervised and data-adaptive manner. With sensor feedback, we construct knowledge-based process models and feed them into a hidden Markov model (HMM), to estimate stages and results during the screwdriving process. We also show that the HMM can automatically adapt and improve as it is fed more examples. This framework is built on the following facts: the mechanics (Chapter 5.1) that dominate individual stages and the structure of stage transition graph (Fig. 5.1) do not change over experiment conditions. Process models (Chapter 5.1) encapsulate local invariant characteristics (for example the mechanics for each stage) in their constraint equations, while an HMM incorporates a globally invariant stage transition graph in its transition model. For environment-dependence, process models treat them as identifiable model parameters, where as an HMM adapts to these variations by updating its observation models during training.

Our system shows robustness and data-efficiency in the following ways. First, it performs online stage classification and result prediction. Second, with simple and limited prior knowledge, our system can automatically label a large amount of data, freeing people from tedious labeling. As new data accumulates, the system adapts to minor changes without human intervention. Third, our system can generalize and adapt to new experiment setups with little new data. Fourth, our system can detect anomalies and unseen situations. To summarize, our main contributions are:

- The first attempt of unsupervised learning and automatic labeling in screwdriving, to our best knowledge.

- A fast generalization framework for industrial problems.
- Significant reduction in data collection and labeling by taking full advantage of screwdriving mechanics.

Beside having accurate stage classification and failure prediction, a robotic screwdriving system should also be affordable for industrial use. To reduce the system cost, we perform some preliminary study on sensor reduction. The goal is to select the minimal set of sensors and produce a low-cost screwdriving system. The discussion and preliminary experiments are presented in Chapter. 6.1.

We also discussed the role of compliance in screwdriving in Chapter. 6.2. In our preliminary experiments, we show that certain degree of compliance enables screwdriving to be successfully performed with larger misalignment.

The rest of this thesis is organized as follows: Chapter. 2 reviews existing work related to screwdriving automation. Chapter. 3 describes our robotic screwdriving system and the dataset we collected using this system. Chapter. 5 presents our stage classification and failure detection system, as well as evaluation results on a subset of the dataset. Chapter. 6 discusses some of our preliminary work: sensor reduction and compliant control for robotic screwdriving. Chapter. 7 concludes this thesis and brings up future work.

Chapter 2

Related Work

In the following, we briefly describe some work that are directly related to this thesis. A exhaustive review of automated threaded fastening can be found in [3].

Analysis of the Screwdriving Process

A thorough understanding of screwdriving process is critical for reliable and accurate fault detection. Typically, the process can be segmented into several stages through the torque-angle curve [12]. A typical torque-angle curve is plotted in Figure 2.1. Recently, [2] presents a comprehensive stage transition graph, covering multiple process stages and result classes, as shown in Figure 2.2 A standard operation often consists of *approach*, *initial thread mating*, *rundown* and *tightening*, while other stages like *hole finding* or *no screw spinning* occur with alignment error or pick-up failure. Each stage has different sensor signal signatures and process models. In *initial mating*, contact models are studied by [17], [14] and [18]. A force spike can be used as an indicator [18] [2]. In *rundown*, quasi-static analysis shows that the oscillation phenomenon is an important signature [19]. In *tightening*, torque, rotation angle and torque-angle gradient [20] are commonly used for failure detection. In Chapter 5.1, we combine these features with our own analysis to develop screwdriving process models.

Fault Detection and Quality Monitoring

Reliable fault detection and error recovery are required by autonomous screwdriving, because even well-engineering systems can be tripped up by factors like part tolerance issues, bad materials, and tool wearing. One of the most commonly used methods is the teaching method,

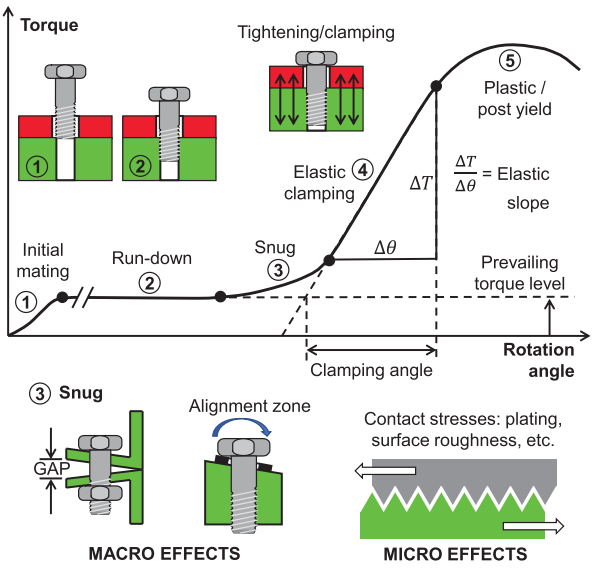

Figure 2.1: Torque-angle curve during the screwdriving process [1]

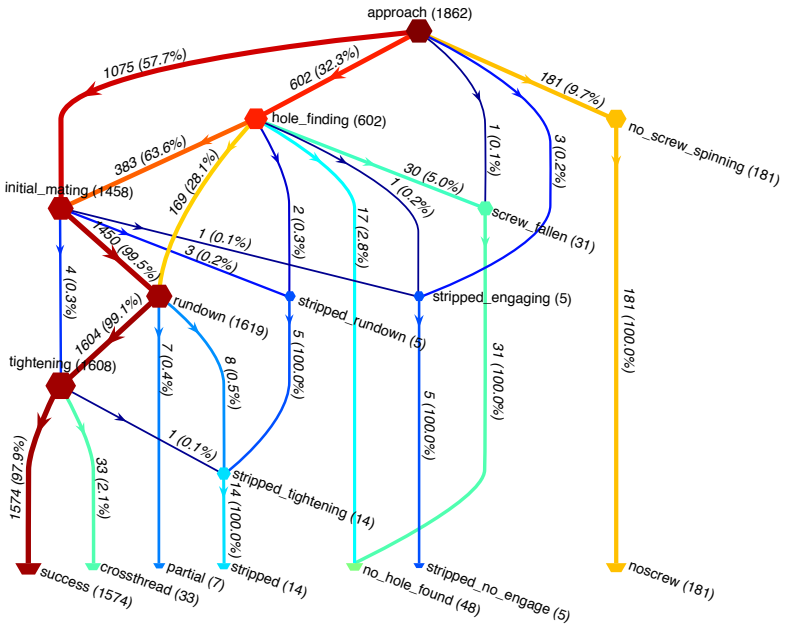

Figure 2.2: A stage transition graph summarizing possible stages and results in the screwdriving dataset from [2]

in which faults typically show up as major deviations from the correct torque-angle fastening signature curve. Correct torque-angle fastening signatures are collected as reference, and then compared with actual signals using limit check or trend check [21]. The teach method is easy to implement, but it lacks flexibility and generality.

To overcome the inflexibility and generality issues of the *teaching method*, intelligent screwdriving systems are developed, most of which fall into two categories: model-based and data-driven. Model-based methods [22] require analytic models and accurate system parameters. Model-based approaches are flexible, but accurate system models are hard to obtain. Data-driven methods, such as Artificial Neural Networks [23], Support Vector Machines [19], Graph of Temporal Constraint Decision Forests [2], Decision Trees [15], and Convolutional Neural Networks [24], directly learn from labeled data. Data-driven methods do not require prior knowledge, but they require a large amount of data.

Most methods mentioned above can achieve around 90% or better accuracy in detecting failure modes. However, many things need to be improved before deploying to assembly lines. Moreover, most of these are developed for hand-held screwdrivers, with no robot information incorporated. However, as shown in Chapter. 5.1, robot signals are strong indicators of system status.

Fault detection alone cannot satisfy the stringent requirements for high-volume production [3] in the consumer electronics industry. To improve the overall success rate, fault prediction and recovery algorithms that can detect preceding failures early and take proper corrective actions upon predicted failure types are desired for future assembly lines [3] [2]; unfortunately, they are still missing in the literature. Our stage classification algorithm developed in Chapter. 5 is the first step to bridging the gap.

Discrete State Estimation in Manipulation

A typical robot task often includes a series of discrete states [25] which can be formulated by Markov models or hybrid systems. Our approach is a combination of these two methods. Hybrid models are incorporated to augment an HMM, while the HMM keeps learning and modifying the hybrid model parameters.

Markov model approaches are probabilistic. HMMs are used to perform automatic action segmentation in [26] and [27]. In [28], which is very related to our work, contact states are predicted by combining contact model estimation with HMM. Contact states are modeled with unspecified parameters estimated from observations, while HMM acts as an acceptance test to predict the most likely state. Its major difference to our work is that we use HMMs to refine and adapt. The hybrid system approaches are more deterministic, which incorporates simplified mechanical models to approximate complex robot behaviors, to reason about mechanisms and to perform feedback control [29] [30]. A discrete state is identified when the state variables fall into the corresponding domain [31].

Sensor Selection for Screwdriving Systems

In previous screwdriving work, sensor selection and control strategies are mainly empirical. For example, many commercial off-the-shelf (COTS) screwdrivers [32] are equipped with sensors to measure or estimate the fastening torque and rotational angle, because these two parameters can provide valuable information for screwdriving, especially for fault detection. Many control strategies also involve monitoring the torque and/or angle information. These strategies can be divided into three broad categories: *torque-only control*, *torque-angle monitoring and control*, and *torque-rate control* [12] [3].

However, one problem associated with COTS systems is that their sensor signals (torque and angle) may not be rich enough for controls and fault detection in robotic screwdriving. For example, robust detection of *initial mating* is critical for *torque-angle control* because it triggers the clamping angle count — an important parameter for quality control. Many high end COTS screwdrivers (for example [32]) compare the driving torque against a user-specified threshold to detect the initial engagement. However, the driving torque (T_z) is not a robust feature because it is almost constant during *initial mating*. In comparison, the insertion force (F_z) might be a much better candidate. Another limitation is that fault detection algorithms based on torque-angle signals alone cannot detect uncommon failure modes as shown in [2] [16].

There are some robotic screwdriving systems that can provide richer sensor readings. In [33], the insertion force (F_z) can be calculated by measuring the spring displacement. In [19] and [2],

6-axis force/torque (F/T) sensors are used to provide much richer information, at the expense of significant increase in system cost. In fact, besides reliability issues, cost (especially the sensor cost) is another factor that prevents further adoption of robotic screwdriving. For example, a high-end screwdriver [32] can cost more than $10,000 while a typical 6-axis F/T sensor [34] cost around $7,000 or even more. In this paper, starting from a screwdriving dataset, we follow a systematic feature reduction approach to identify the optimal subset of sensord that are needed for both online stage and result classifications (the building blocks for a fault prediction and recovery system), while minimizing the system cost.

Chapter 3

Robotic Screwdriving System

3.1 Robotic Screwdriving System

Our robotic screwdriving system is shown in Fig. 3.1. This robotic system is built on a 6 degree-of-freedom industrial robot arm, FOXBOT A600. A screwdriver is installed as the end-effector of the robot arm, as shown in Fig. 3.1 A high speed camera is attached to the screwdriver to record videos during screwdriving operations. In the robot workspace, a shaker tray to hold screws and a replaceable plate with threaded screw holes are installed on a table.

The screwdriver is shown in Fig. 3.1, which has a direct drive Maxon EC 45 motor and an ATI mini-40 force-torque(F/T) sensor. The motor drives a screwdriver bit. The screwdriver bit passes through the centered hole on the F/T sensor. The F/T sensor is attached to the screwdriver bit with an independent structure so that it can sense force and torque exerted on the bit. By using this ”floating structure“ design, forces and torques exerted on the screwdriver tip can be measured by a 6-axis F/T sensor. The screwdriver is attached to the robot flange with linear compliance through springs. A linear potentiometer is used to measure the deformation of the linear springs. Combining robot position information and linear potentiometer readings, we can infer the actual position of the screwdriver bit.

3.2 Robotic Screwdriving Operation Procedure

In each screwdriving operation, a screw from the shaker tray is first picked up by the screwdriver using vacuum suction. Then the robot moves above the calibrated screw holes on a plate. The

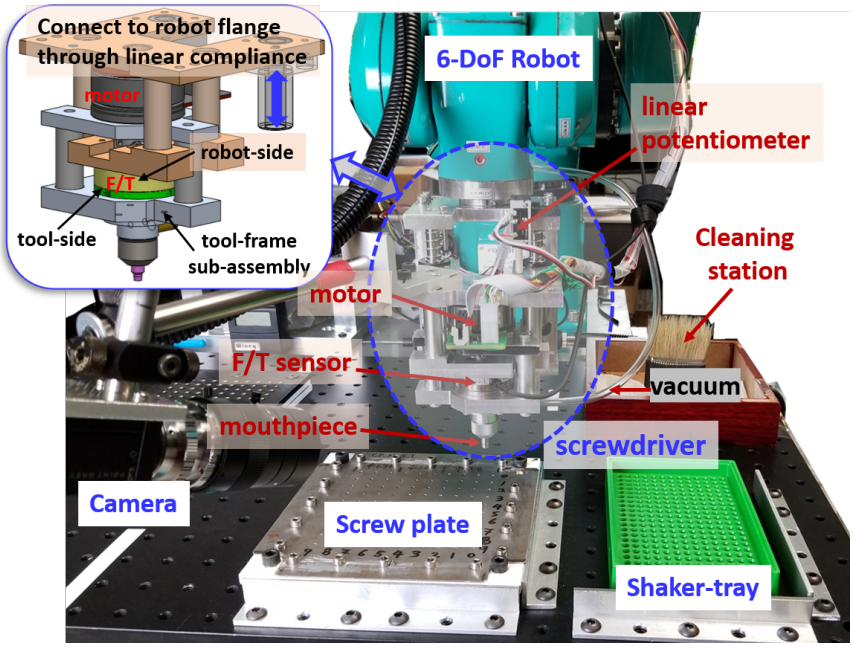

Figure 3.1: Intelligent robotic screwdriving system for data collection.

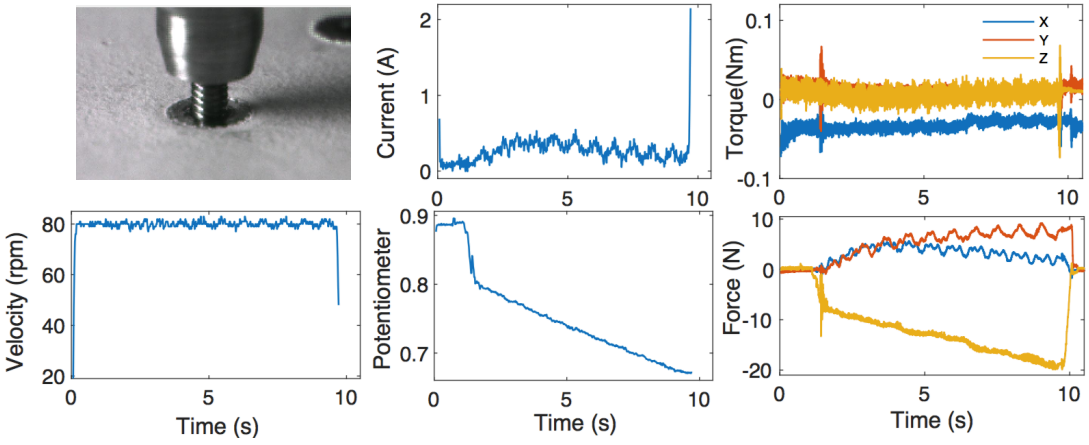

Figure 3.2: An example of collected raw sensor data. The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis.

screwdriver motor turns on and starts collecting data as the screw insertion starts. During the insertion, the robot speed is coordinated with screwdriver motor speed so that the insertion force (the force that the screwdriver exerts vertically on the screw) is roughly maintained to be constant. Each run terminates when the motor current reaches a specified limit or the motor encoder reading exceeds a large threshold.

The system records robot positions, 6-axis F/T data, motor current, motor encoder readings, linear potentiometer values that measure the spring displacement (to trace the screwdriver tip), and videos (as the ground truth) from a high-speed camera. A typical successful run is shown Fig. 3.2.

Chapter 4

Screwdriving Dataset

4.1 Overall Description

Using the procedure decribed in Section 3.2, we collected a large dataset of experiment data of different sizes of small screws under different experiment setups.

For each screwdriving operation, we collect sensor data including: 6-axis forces and torques, motor currents, motor speeds, motor encoder readings, robot positions, linear potentiometer readings, and a video.

In this dataset, all the screws are Phillips-head screws. All the screw holes are threaded holes. The screw sizes vary from M1.0x3 to M2.5x5. Screw head types include both pan-head and flat-head. Some screws have a blue nylon material on the threads (NYLOK) for the purpose of thread locking and sealing.

The screwdriving operation conditions also vary. A detailed description can be found in Chapter. 4.2. For each operation condition, we collect data with both high precision alignment ($1/4$ of total) and random alignment errors ($3/4$ of total). When collecting high precision alignment data, accurate positions of screw holes are given to the robot. To obtain more failure data, we also add alignment errors of the screw central axis to the screw hole central axis, including lateral errors and angular errors. The lateral errors are uniformly sampled in the range of 0% to 40% of screw diameters and the angular errors are uniformly sampled in the range of 0° to 6°.

Compared to the industrial data, our data have much more unexpected and deviated patterns, which largely increase the difficulty for prediction. The actual average rate of successful operations in our dataset is around 75%.

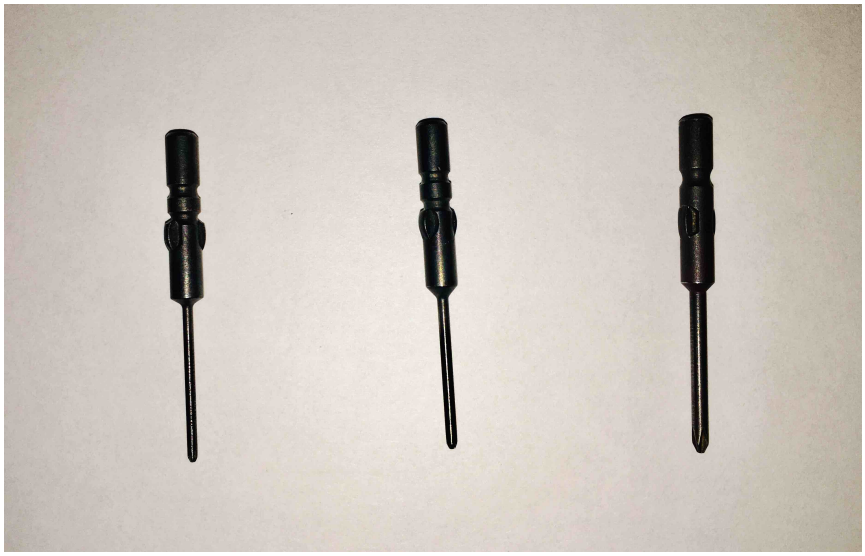

Figure 4.1: Screwdriver bits, from left to right: PH#00-1.18, PH#00-1.38, and PH#1-1.95

Figure 4.2: Left: rigid-bit, right: flex-bit. The red circle points out the cut slot that makes the screwdriver bit flexible when fixed with a small screw.

4.2 Screwdriving Operation Parameters

Please refer to Table. 4.1 for a description of operation parameters. For detailed information on the operation parameters for each operation condition, please refer to this description form on our dataset website.

Figure 4.1 shows the screwdriver bits. In the description form, “rigid-bit” means that the screwdriver bit is tightly installed on the screwdriver. “flex-bit” means that the screwdriver bit is loosely installed (has a cut slot at the fixing point) so that it has some flexibility to move vertically (about ± 0.2 mm). See Figure 4.2 for the comparison of “rigid-bit“ and “flex-bit”.

Table 4.1: Description of Operation Parameters

Operation Parameter	Description
Screwdriver Bit	the size and type of the bit of the screwdriver
Tighting Current	the current threshold for the screw to be tightened
Rundown RPM	the screwdriver motor spinning speed during insertion
Insertion Force	the force vertically exert on the screw at the start of insertion
Compression Force	the force vertically exert on the screw at the end of insertion
Position Error	whether lateral alignment errors are added
Angular Error	whether angular alignment errors are added

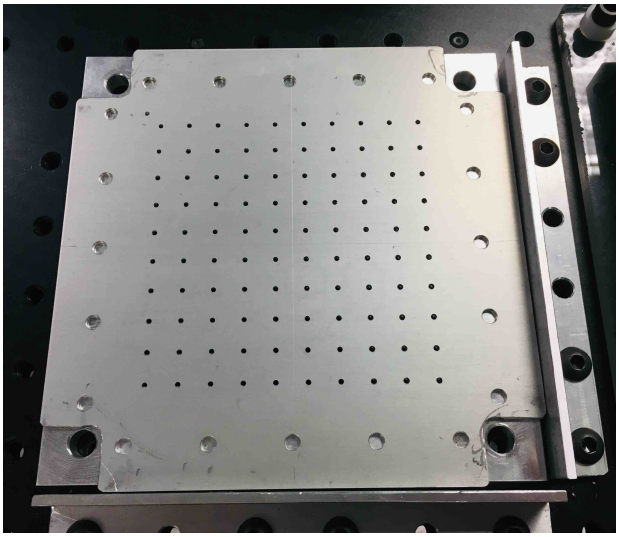

Figure 4.3: A cover plate on a screw hole plate

"Notes" in the description form informs whether there are other added operation settings. For some operations, we add metal cover plates on the screw plate aligned with the screw holes, as shown in Figure 4.3. To simulate the case that the assembled parts are not well aligned with the screw holes, for some operations we add alignment offsets of the cover plates and the screw holes.

While most operations are performed with fixed screw plates, a small portion of operations are completed with a compliant screw plate. This screw plate is installed on a two-axis variable stiffness actuator plate (compliant x-y plate), see Figure. 4.4. This is also noted in "Notes".

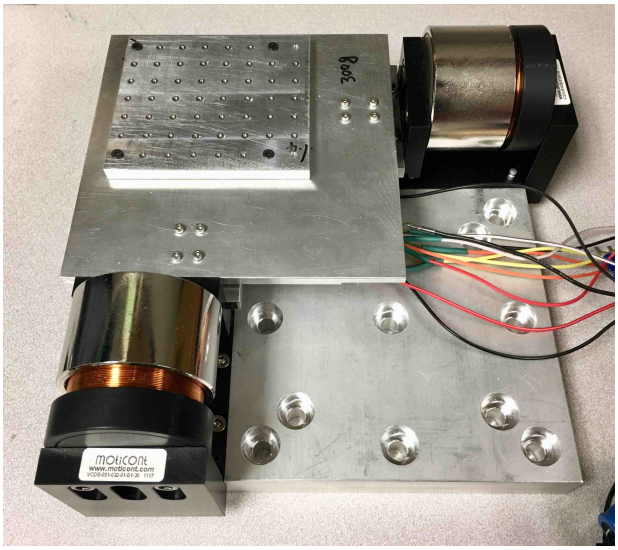

Figure 4.4: a screw hole plate installed on our compliant x-y plate

4.3 Dataset Structure

For each operation, the data and video are stored in 5 files: `*screwdriver.csv`, `*ft.csv`, `*info.json`, `*foxbot.csv`, and `*.avi`. For each operation, the symbol `*` is replaced with its numbered file name.

`*foxbot.csv`: robot positions in Cartesian space, including positions and quaternions

`*ft.csv`: force and torque data from an ATI-mini 40

`*screwdriver.csv`: data from screwdriver, including: position (motor encoder reading), velocity (motor velocity readings), current (motor current), and potentiometer (linear potentiometer readings)

`*info.json`: basic information about this screwdriving operation, including operation duration, screwdriver hole location and the value of the introduced alignment errors.

`*.avi`: a video recorded by a high speed camera.

4.4 Downloads

Please visit this site for download instructions.

Chapter 5

Stage Classification and Result Prediction

An overview of our system is as follows. First, all possible stages during the screwdriving process are specified and modeled as constraint equations. Second, model residuals of the constraint equations are used as inputs of an HMM, which can be trained with or without labels. Once the stages are predicted by the HMM, results (successes or failures) are inferred from stages using constructed rules from the stage transition graph. Third, given new data from similar experimental setups, a well-trained system generalizes to new data by parameter specification and HMM learning — our system quickly adapts to new experiment setups. Fig. 1.1 shows the pipeline of our online stage estimation and result classification.

5.1 Modeling of the Screwdriving Process

The screwdriving process can be modeled through mechanical knowledge and empirical observations. Our modeling includes building a stage transition graph, designing a feature set, and modeling stages as constraint equations.

5.1.1 State Transition Graph

A screwdriving operation can be modeled as a sequence of discrete states, which are denoted as *stages*. The stage transition graph (Fig. 5.1) describes all possible transitions among the stages of our system. All operations begin at *approach*. A standard successful operation is followed by *initial mating*, *rundown*, *tightening* and *stop*, while *hole finding* stage could appear when there are alignment errors. Failures are indicated by stages, such as *no screw spinning* and

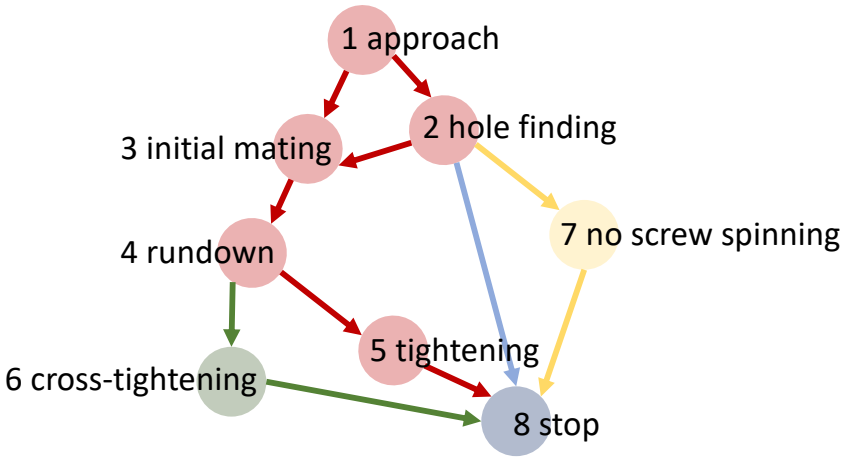

Figure 5.1: The stage transition graph summarizes all possible stages in the screwdriving process.

cross-tightening correspond to results *no screw* and *cross thread*. All operations end at *stop*. A well-defined graph enables us to clearly and separately model the screwdriving process, because we assume each stage has only one dynamic model. Moreover, this graph is used as the state transition model of the HMM. Note that this graph is modified based on [2]. Compared to the previous graph in Figure. 2.2, by exerting proper insertion force, stages and results related to *stripped* happen rarely, thus can be removed from the stage transition graph and treated as anomalies.

5.1.2 Feature Extraction

Feature extraction pre-processes raw sensor data before stage modeling. We design a feature set based on the previous work and our observations. These features enable complicated processes to be modeled as linear equations of features and parameters, making parameter estimation and generalization easier. At timestep t, a set of features is extracted from the signal data in a fixed-size interval of time before t:

v_x, v_y, v_z: frequencies of the maximum amplitude of forces computed by discrete Fourier transform. The oscillation phenomenon of the forces on x axis and y axis is only found in the rundown stage of the robotic screwdriving task [19]. With small alignment errors, the frequency approximately equals the screwdriver motor rotational speed.

d_{tip}: the distance from the screwdriver tip to the screw hole. It is a strong indicator of the

process status. It is calculated as:

$$d_{tip} = r_z - h_z - l_z \tag{5.1}$$

where r_z is the screwdriver end position on the z axis, computed from the robot positions. l_z is the compression of linear potentiometer, and h_z is the z-value of the calibrated hole location.

$\Delta_{d_{tip}}$: the change of screwdriver tip distance in the time window, we have

$$\Delta_{d_{tip}} = max(d_{tip}) - min(d_{tip}) \tag{5.2}$$

k_{tip}: the gradient of the screwdriver tip distance. It is computed by least square estimation for

$$d_{tip} = k_{tip} t \tag{5.3}$$

The other gradients in the following are also computed in the same way.

k_{ta}: the torque-angle gradient. This feature is widely used in torque control threaded fastening for indicating the status of tightening, whether in elastic zone or yield zone.

k_t: the z axis torque-time gradient.

k_f: the z axis force-time gradient.

k_c: the motor current-time gradient. Our screwdriver employs a direct-drive motor; the motor torque is proportional to the motor current. It is used as a redundant feature to provide more robust information against relatively large noise on the torque sensor.

Δ_{fz}: the change of force on the z axis, written as

$$\Delta_{fz} = max(f_z) - min(f_z) \tag{5.4}$$

where f_z is the z axis force in the time window. A drop of z axis force often indicates the alignment or mating of screw threads [18].

$\mu_{\mathbf{f}}$, $\sigma_{\mathbf{f}}$: the means and variances of forces.

5.1.3 Stage Models

Each stage is modeled as a set of constraint equations using physics or our process knowledge. After stage models (constraint equations) are obtained as one large set of constraints, the model

errors are used as the input to the HMM to perform stage estimation. Some stage model parameters can be specified from the experiment setup (setup parameters): V_m, H_1, H_2, C_{1kta}, and C_{1kc}. Other parameters are learned from the labeled data before HMM training (learned parameters): $C_{1k_{tip}}$, C_{1k_f}, $C_{2k_{tip}}$, C_{2k_f}, C_{2kta}, and C_{2kc}. The meanings of these parameters are explained later in this part.

Our stage models are constructed as follows:

approach: no contact between the screw and target hole, thus zero forces are assumed:

$$\mu_{\mathbf{f}} = 0 \tag{5.5}$$

$$\sigma_{\mathbf{f}} = 0 \tag{5.6}$$

hole finding: the screw moves around without insertion, while the change of the screwdriver tip distance is zero:

$$\Delta_{fz} = 0 \tag{5.7}$$

initial mating: We consider a smooth mating, where the screw is inserted at a constant velocity $C_{1k_{tip}}$ and a constant gradient of force C_{1k_f}:

$$k_f - C_{1k_f} = 0 \tag{5.8}$$

$$k_{tip} - C_{1k_{tip}} = 0 \tag{5.9}$$

rundown: the vibrations of forces on the x axis and y axis occur in this stage, indicating the precession of the screw axis [19]. The vibration frequency can be approximated by motor velocity V_m. At the same time, the screw goes down with a constant velocity $C_{2k_{tip}}$ and a constant gradient of force C_{2k_f}:

$$v_x - V_m = 0 \tag{5.10}$$

$$v_y - V_m = 0 \tag{5.11}$$

$$k_f - C_{2k_f} = 0 \tag{5.12}$$

$$k_{tip} - C_{2k_{tip}} = 0 \tag{5.13}$$

tightening: for the successful tightening, the torque-angle gradient must satisfy a preset value C_{1kta} to ensure that the screw is correctly preloaded. Since the motor current is proportional

to the screwdriver torque, the gradient of current should also approximately equal a constant C_{1kc}. The distance of screwdriver tip to the screw hole surface is the thickness of screw head H_1:

$$k_{ta} - C_{1kta} = 0 \tag{5.14}$$

$$k_c - C_{1kc} = 0 \tag{5.15}$$

$$d_{tip} - H_1 = 0 \tag{5.16}$$

cross tightening: the model is the same as that of *tightening* but with different model parameters C_{2kta}, C_{2kc} and H_2. The torque-angle gradient is much smaller than that of *tightening* due to angular errors. When cross threaded, only the first external thread is mated at the cross thread angle [17], thus the screwdriver tip distance can be computed as

$$H_2 = l \cdot cos(\theta) \tag{5.17}$$

where l is the screw length and θ is the cross thread angle.

no screw spinning: the vibration of z axis force at the frequency of V_m occurs when the spinning screwdriver tip contacts with the screw plate or the screw hole, and is periodically pushed back:

$$v_z - V_m = 0 \tag{5.18}$$

stop: the screwdriver motor stops, thus the motor current m_c and motor velocity m_v are zeros:

$$m_c = 0 \tag{5.19}$$

$$m_v = 0 \tag{5.20}$$

5.2 Stage Prediction by Hidden Markov Model

Given data from a screwdriving operation, features are first extracted. For each stage, the residuals of its constraint equations are computed. Then these stage model residuals are passed to a Hidden Markov model to make stage prediction. Given a time series of observations, an HMM can estimate the sequence of states which generate these observations.

5.2.1 HMM Representation

An HMM λ is composed of states S, initial state distributions π, state transition distribution A, and observation probability distributions B [35]. Given states, a compact notation of the HMM parameters is $\lambda = (A, B, \pi)$. In our system, the hidden states $S = \{S_1, S_2, \ldots, S_N\}$ are the stages in Fig. 5.1. The state at time t is denoted as q_t. The initial state distribution is the probability of each state that appears at the start, denoted as $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$, where $\pi_i = P(q_1 = S_i)$.

The state transition probability matrix $A = \{a_{ij}\}$ represents the probability that state S_j occurs after S_i, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N$. If there is no edge going from S_i to S_j in the stage transition graph, we have $a_{ij} = 0$; otherwise $a_{ij} > 0$.

The observation probability distribution $P(O_t | q_t = S_i)$ provides the probability density that the observation at time t (O_t) emitted by hidden state S_i. O_t includes robot positions, 6-axis F/T data, motor current, motor encoder readings, and linear potentiometer values as described in Chapter 4. In our method, the $P(O_t | q_t = S_i)$, written as $b_i(O_t)$, is represented in the form of a multivariate Gaussian distribution:

$$b_i(O_t) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp(-\frac{1}{2}(x_{it} - \mu_i)^T \Sigma^{-1} (x_{it} - \mu_i) \tag{5.21}$$

where x_t is the residuals of all constraint equations computed from O_t as in Section 5.1; μ_i and Σ_i are the mean vector and covariance matrix of the model residuals for state S_i.

5.2.2 Stage Classification

Stage classification is, at the latest timestep T, to find the state sequence that maximizes its joint probability $P(q_1, q_2 \ldots q_T \wedge O_1, O_2 \ldots O_T | \lambda)$ with the observation. The Viterbi algorithm, a dynamic programming algorithm, can solve this problem efficiently [35]. The forward-backward procedure is performed firstly. The forward variable is defined as $\alpha_t(i) = P(O_1, O_2 \ldots O_t, q_t = S_i | \lambda)$. It can be solved inductively as:

$$\alpha_1(i) = \pi_i b_i(O_1) \tag{5.22}$$

$$\alpha_{t+1}(i) = \sum_{j=1}^{N} \alpha_t(j) a_{ji} b_i(O_{t+1}) \tag{5.23}$$

Similarly, the backward variable $\beta_t(i) = P(O_{t+1}, O_{t+2} \ldots O_T | q_t = S_i, \lambda)$ can also be inductively solved as:

$$\beta_T(i) = 1 \tag{5.24}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \tag{5.25}$$

We define the most probable state sequence ended with S_i at timestep t as $\text{mpp}_i(t)$, and its probability as

$$\delta_i(t) = \max_{q_1 \cdots q_{t-1}} P(q_1, q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 \ldots O_t) \tag{5.26}$$

The most probable path computation is:

$$\delta_1(i) = \pi_i b_i(O_1) \tag{5.27}$$

$$\delta_{t+1}(j) = \delta_t(i^*) a_{i^* j} b_j(O_{t+1})) \tag{5.28}$$

$$\text{mpp}_j(t+1) = [\text{mpp}_{i^*}(t), S_{i^*}] \tag{5.29}$$

where

$$i^* = \arg\max_i \delta_t(i) a_{ij} b_j(O_{t+1}) \tag{5.30}$$

During the screwdriving process, at each timestep t, the current state sequence is predicted as $mmp_j^*(t+1)$, where

$$j^* = \arg\max_j \delta_t(j) \tag{5.31}$$

5.2.3 Data Adaptation

An HMM can iteratively adjust to better models as data accumulate using the Expectation-Maximization algorithm [35]. In our case, data adaptation is achieved by updating the mean μ and variance Σ of the multivariate Gaussian observation model.

The probability of observing state S_i for all $i = 1, \ldots, N$ can be computed as follows:

$$\gamma_t(i) = P(q_t = S_i | O_1 \ldots O_t, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)} \tag{5.32}$$

Thus the mean and variance can be updated similar to the weighted sum:

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) x_t}{\sum_{t=1}^{T} \gamma_t(i)}, \hat{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(x_t - \hat{\mu}_i)(x_t - \hat{\mu}_i)^T}{\sum_{t=1}^{T} \gamma_t(i)} \tag{5.33}$$

Since the starting states can only be S_1, *approach*, we have $\pi = \{1,0,0,0,0,0,0,0\}$. The state transition matrix A is empirically initialized through expert knowledge. If there exists an edge from S_i to S_j, a_{ij} should be constrained as nonzero during training. For the observation probability $b_i(O_t)$, all the Gaussian means are initialized as zeros. The covariance matrix, as a diagonal matrix, is the key to differentiate states. For state S_i, the elements in Σ_i that correspond to its own constraint equations in all equations are initialized to be small. The elements in Σ_i that correspond to constraint equations of the states other than S_i are set large.

5.2.4 Anomaly Detection

We detect anomalies for unknown stages or impossible state sequence. Unknown stages are those with observation probabilities very close to zero for all the states. Impossible path patterns signify no possible transition between the the possible states corresponding to the observations. The two cases mean that the probability of all the observations is zero, written as

$$P(O_1 \dots O_T|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = 0 \tag{5.34}$$

5.3 Rule-Based Result Prediction

There are five types of results, which can be inferred from stage sequences [2]: *success*, *cross thread*, *no screw*, *no hole found* and *partial*. The rules for result prediction are constructed as follows:

1) If *cross tightening* or *no screw spinning* are detected, the corresponding failure types (*cross thread* or *no screw*) are assigned.

2) If the time of *hole finding* or *initial mating* (with large forces) stage exceeds a predetermined threshold, this indicates a large non-correctable alignment error that might damage the screw plate. The system will predict *no hole found*.

3) If the operation follows the red path on Fig. 5.1, then the system will proceed to a condition check, which will examine the final insertion length, tightening torque and tightening torque-

Table 5.1: Data collection settings

dataset	*1*	*2*	*3*	*4*
screw size	M1.4x4	M1.2x3	M2.5x5	M1.4x4
number of samples	396	386	196	35
*tightening current (*mA*)*	1600	1300	3200	2400
motor velocity (rpm)	80	96	80	320
*insertion force (*N*)*	10	6	18	10
*tightening force (*N*)*	20	14	30	20

angle gradient. The result is predicted as *success* if the condition check is passed, otherwise *partial*.

5.4 Generalization

Previously trained HMM cannot be directly used when the operating conditions change. However, our system can quickly generalize to new experiment conditions. First, we change the setup parameters to match the new experiment setup as in Section.2.3. Then, learned parameters are estimated from previous data in the old setup. For example, in *cross tightening* and *tightening*, we often have $5C_{2kf} = C_{1kf}$. This relationship is estimated from old process models and can be used as a good initialization for the new dataset to generalize. These estimated parameters do not need to be accurate because we can always use data adaptation to refine. The new HMM keeps improving as new experiment data feeds in.

5.5 Experiments

5.5.1 Dataset and Implementation

We evaluate our method on four datasets from the large datast mentioned in chapter 4 under different operation conditions as in Table 5.1. The operation condition parameters are explained in Chapter. 4.2, Table. 4.1. We uniformly sample all data at 100 Hz.

We implemented our algorithms in MATLAB. In the process modeling, the setup parameters are manually modified according to our operation settings. The learned parameters are estimated from the 10 labeled well-aligned operation samples from *dataset 1*. The parameter estimation

is solved as a least square problem. We implement the HMM based on [35]. The proper initial values of covariances corresponding to their state constraint equations can be set as $1/10$ of corresponding constant terms, if not available in parameter estimation. The other large error variances are empirically initialized as 100 for small screws. From our experience, as long as these values are set to be large, the system will be able to distinguish different states. During training, to ensure convergence, the mean and covariance in observation models are constrained within $\pm 15\%$ change of the initialization. The HMM is considered converged when the change in the log-likelihood of observations is less than a set threshold of 30. Our HMM training often converges after 8-10 batches in less than 3 minutes with 50 samples in each batch. Rules of result prediction are constructed as a state machine.

5.5.2 Results

Some selected operations are firstly visualized. The actual performances of our method are then evaluated by the failure prediction accuracy.

Stage and Result Classifications

Fig. 5.2 visualizes the stage classification of our HMM on high accuracy alignment data and random alignment error data before and after training. The HMM before adaptation (*HMM before*) is only initialized on 10 accurately aligned samples, while the HMM after adaptation (*HMM after*) are trained on 50 unlabeled samples of both alignment types. All HMMs are tested on new samples that have never been seen by our system. This comparison emulates the HMM adapting to new data as the experiment conditions gradually change. *HMM before* makes reasonable predictions on accurate alignment data for all types of failures. With data adaptation, *HMM after* learns better stage models. For example, in Fig. 5.2 (a), *HMM after* predicts to extend the period of *initial mating* to exactly where an expert label would be. In Fig. 5.2 (c) and (d), the predictions for *cross-tightening* are narrowed down to the right period that cross thread occurs. As errors are introduced, *HMM before* makes more mistakes because of the deviation of the signal data. *HMM after* adjusts to errors while maintaining performance on accurate alignment data.

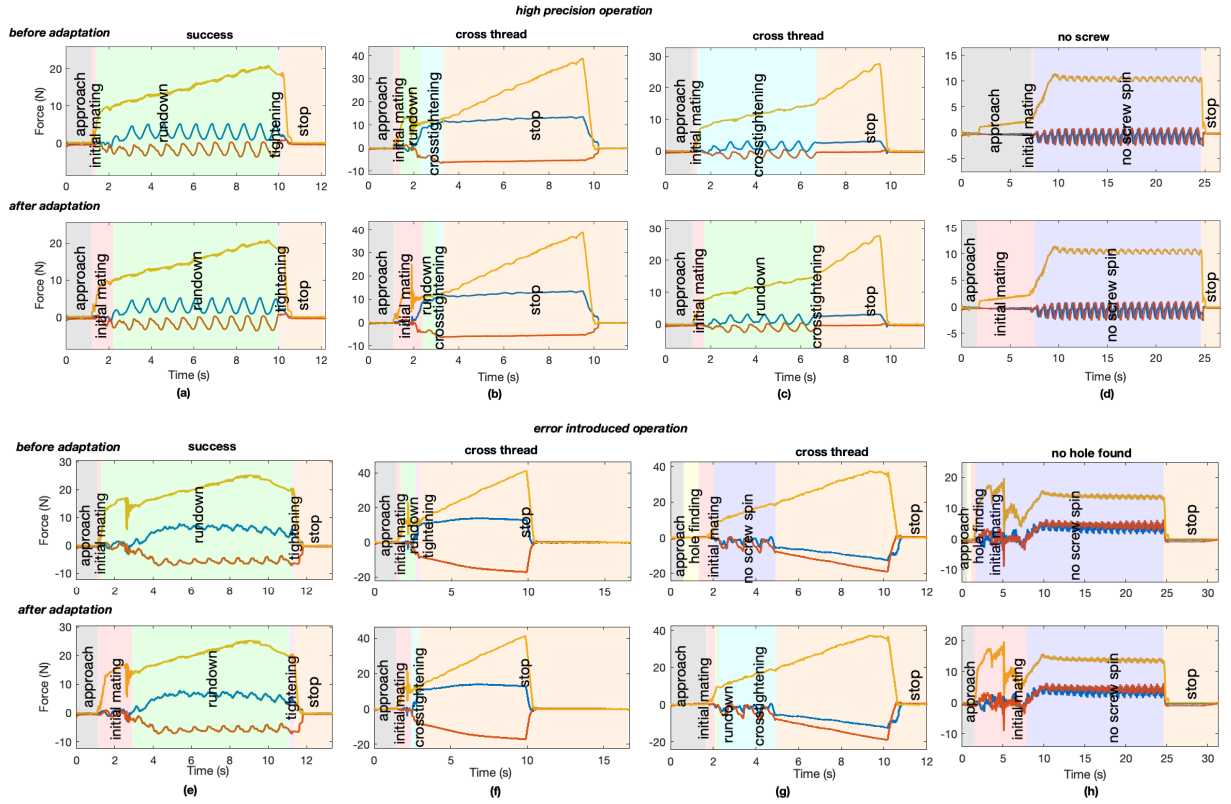

Figure 5.2: Illustrative samples of stage classification selected from our dataset. Comparison of the stage predictions before and after data adaptation on high precision operation data and error introduced operation data. Different color blocks correspond to different stages. Only the force sensor data are plotted due to space limitations. The actual result for each operation is shown on the top of the figure.

To show the above mentioned improvement, we visualize the cases that HMM before adaptation makes mistakes while HMM after adaptation improves. The performance of the HMM before adaptation, evaluated in the latter part of this section, is actually better than shown in our visualization.

Fig. 5.2 covers most of the result types. The operation *partial* has the same signal profiles as *success*, it is the quality check based on actual assembly requirements that distinguishes them. The profiles of *no hole found* vary largely; Fig. 5.2(h) visualizes one example: the screw tip fails to match the screw hole during *initial mating*, when the screwdriver goes down, the screw is pressed away and thus results in stage *no screw spinning*.

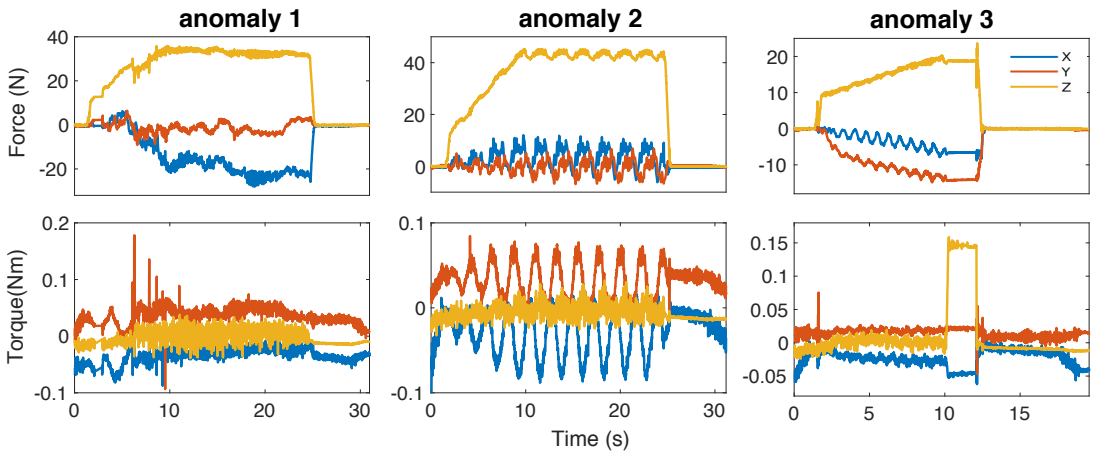

Figure 5.3: Anomalies detected by our system. The first row: the force profiles. The second row: the torque profiles. The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis.

Anomaly Detection

Fig. 5.3 shows three anomaly cases our system detected. It is even hard for an expert to tell what happened simply from the signals. By checking the videos, we found the reasons for these anomalies: for *anomaly 1*, the screws failed matching the screw hole, and the screwdriver was stuck by the head of the fallen screw. For *anomaly 2*, the screw was cross threaded and accidentally stripped due to the misalignment. In *anomaly 3*, the data connection was lost for two seconds.

Generalization

As shown in Fig. 5.4, without any label, our method generalizes to *dataset 2*, *3* and *4*. In *dataset 2*, we use tiny screws (M1.2x2). The tightening torque is much smaller, making the noise-signal ratio of the torque sensor as high as 50%. Our method can reduce the influence of noise by exploiting known meaningful models. In *dataset 3*, the screw diameter (2.5 mm) is almost 2 times of that of our original dataset (1.4 mm). This verified that our model can be adapted to much larger screw sizes and tightening torques. In *dataset 4*, the system generalizes to a motor speed (320 rpm) four times faster than the original dataset (80 rpm). In real factory scenarios, screwdriving motor speed varies and can be really fast (over 1,000 rpm).

Table 5.2: Result Accuracy

dataset		*1*(accurate)	*1*(error)	*2*	*3*	*4*
HMM type		original		generalized		
adapt	before	97.47%	77.27%	89.90%	91.33%	88.57%
	after	98.48 %	84.85%	91.71%	94.38%	91.43%

Evaluation

We evaluate our method using the accuracy of result prediction (Table.5.2), the proportion of correctly predicted results in all the samples. The correctness of stage classification can be told through visualization. The classification visualization can be found in *our GitHub repository*. We do not directly measure the performance of stage classification because manually labeling the stages is extremely time consuming. The result prediction, inferred by predicted stages, can indirectly show the effectiveness of stage classification.

Our models obtained over 97% result prediction accuracy for the original ones with parameter estimation from labeled samples. The generalized HMMs, initialized with known experiment setup parameters and trained on unlabeled data, still make reasonable predictions. By data adaptation, our model can automatically fit to the actual data and make more accurate predictions. Note that we intentionally increase the difficulty for prediction, thus our result accuracy is not comparable to the desired 99.9% for industry use. For example, for *dataset 1*, we introduce alignment errors, producing many largely deviated signal patterns. The HMM is trained with limited data that cannot cover these deviations. This is not a problem for industrial use, where operations are highly accurate and repeatable.

5.5.3 Limitations

At data adaptation, we found one stage model might fit to another stage, because prediction errors can accumulate and reinforce when training without labels. To prevent this, we add constraints on the means and covariances. However, to actually solve this problem, we need observation models with higher capacity, but complicated models require more data and labeling to train. One possible solution to this trade-off is introducing human correction.

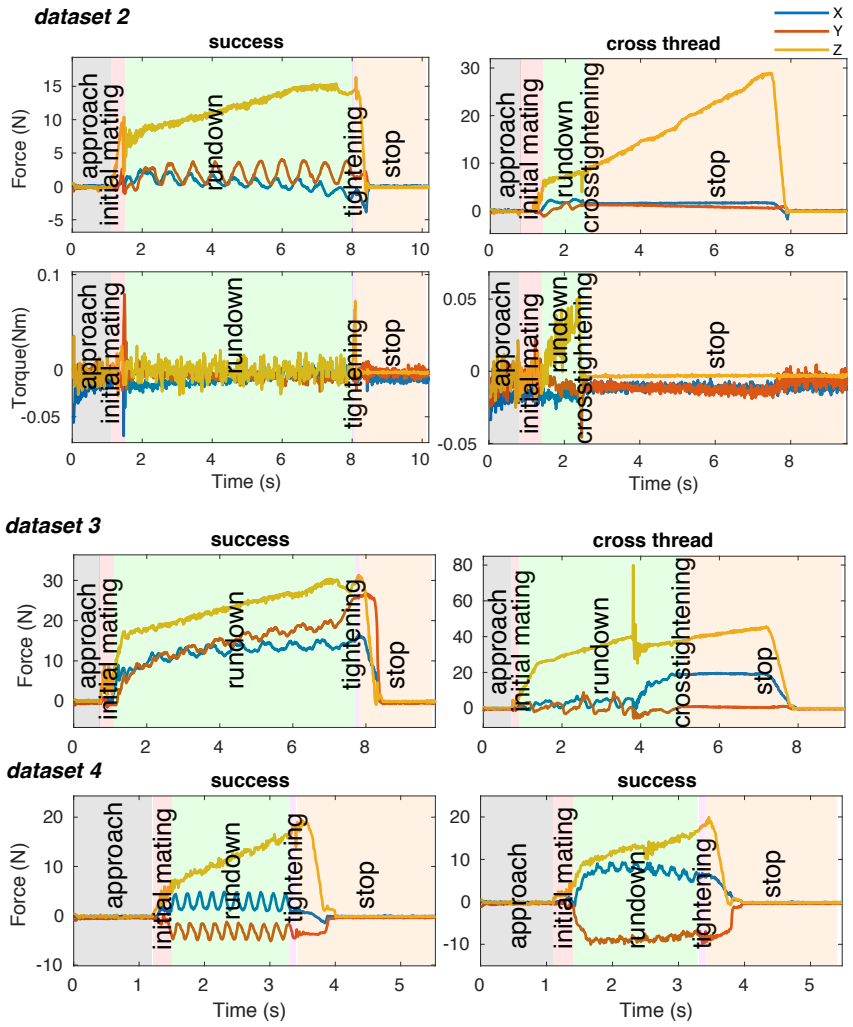

Figure 5.4: The visualization of the generalization of our system on *dataset 2* (force and torque profiles), *dataset 3* and *dataset 4* (force profiles only). The yellow colored signals are from the z axis; the blue colored signals are from the x axis; the red colored signals are from the y axis.

Chapter 6

Discussion

6.1 Sensor Reduction

One factor that prevents further adoptions of automated screwdriving systems is cost. For example, six-axis F/T sensors provide much richer information for failure detection, at the expense of significant increase in system cost. Hence, one goal in the automation of screwdriving is to select proper sensor signals to produce affordable intelligent screwdrivers for consumer electronics products.

In screwdriving literature, the sensor selection and control strategies are mainly empirical, our preliminary study presents a systematic way to identify optimal sensor readings to produce low-cost intelligent screwdriving systems. Please refer to our paper [15] for more detailed discussions.

6.1.1 Data for Sensor Reduction

The data we used for sensor reduction is slightly different from the dataset in chapter. 4. The data is collected by our previous version of robotic screwdriving system in chapter. 3. Detailed descriptions and discussions of the screwdriving dataset used for sensor reduction here can be found in [2] [16].

A brief description about this screwdriving dataset is as follows. In [2], the authors collected 1862 screwdrving runs, each of which consists of 6-axis force and torque (see Fig. 6.1), motor current and speed, and video data (see Fig. 6.2), all sampled at 100 Hz. After data analysis, [2] empirically came up with a list of stages (through which the screwdriving operation progresses)

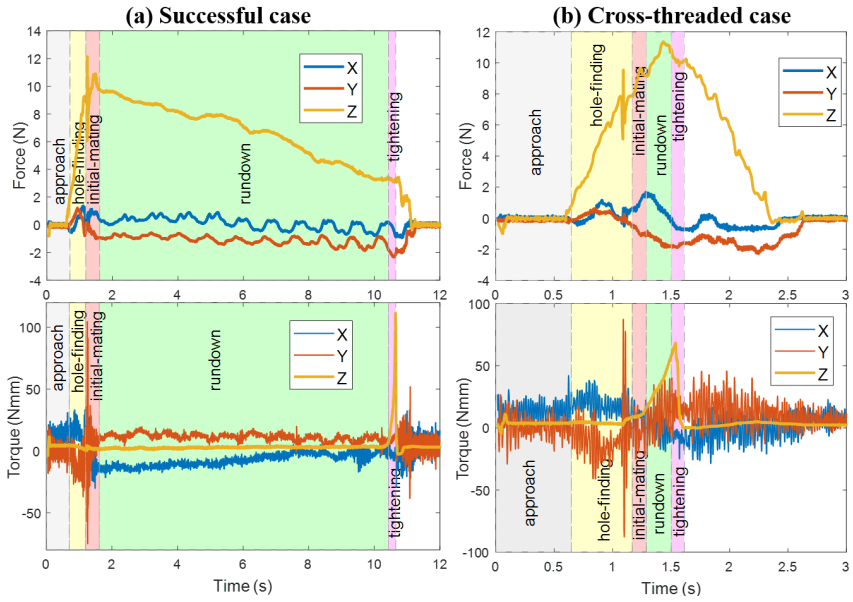

Figure 6.1: The 6-axis force and torque signatures of (a) successful case with *hole finding* stage and (b) an unsuccessful (*crossthread*) case.

and result classes, as shown in Fig. 6.2. A complete description of the stages and result classes can be found in [2]. This dataset has also been hand-labeled with corresponding stage (see Fig. 6.1) and result information for each run, forming the largest labeled screwdrving dataset we know of [2]. In sensor reduction, we only use the result labels from this dataset.

6.1.2 Sensor Reduction based on Result Classification

Our sensor reduction is performed based on the accuracy of result classification. That is, we hope to choose the best and minimal sensor set that can produce most accurate result classification.

Fast and accurate result classification is performed by linear discriminant analysis (LDA) models [36]. A wrapper method for feature subset selection [37] is used to reduce the number of required features and sensor signals. Reduced feature subsets (see Table 6.1 and Table 6.2) from less sensor signals are selected to produce a highly accurate yet affordable robotic screwdriving system.

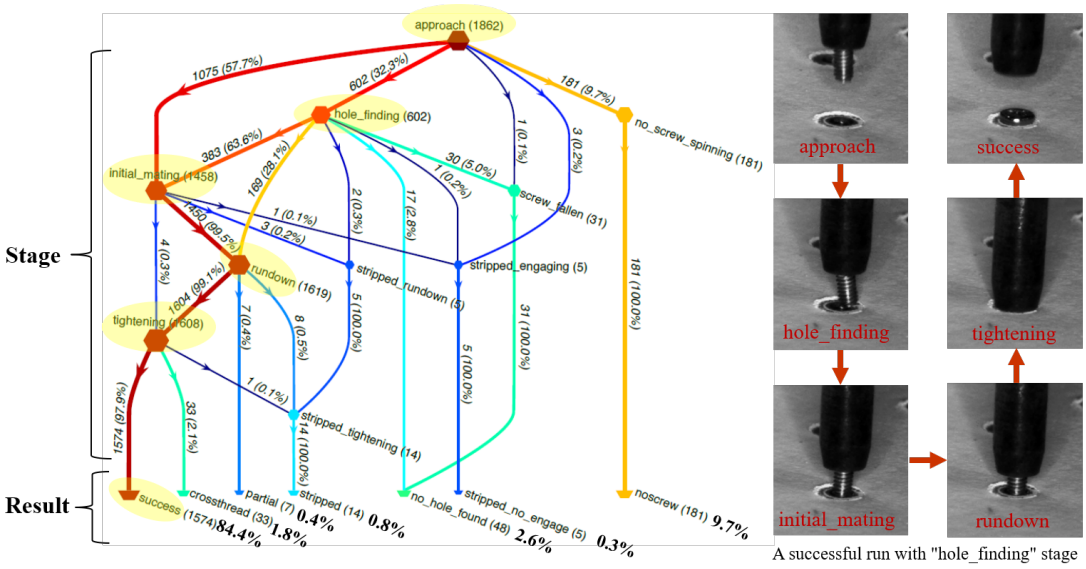

Figure 6.2: Left: The state transition graph summarizing all the stage and result classes. The vertices represent stages through which the screwdriving passes, with the terminal stages corresponding to the result classes. Colors and sizes are scaled logarithmically with the number of runs in each transition, stage, or result (modified from [2]). Right: A successful screwdriving run example including a *hole finding* stage.

Result Classification

First, from sensor signal data, global features are extracted. For each screwdriving run, 85 global features are extracted from the whole length of time series data from full sensor signals (8 channels in total). The angle feature (θ) is provided by the last motor encoder reading, which is commonly used in industry as an important criterion for failure detection, as discussed in Chapter. 2. The other 84 features are based on signals from the 6-axis F/T sensor and motor current, i.e., 7 channels in total. For each channel, 12 statistic features are extracted, including the range, mean and standard deviation of the following time series data: the original data, its first order differences (Δ), its second order differences (Δ^2), and its successive ratio (Δdd). Finally, each feature is normalized to have a standard normal distribution.

Then, given 1862 labeled samples and 85 normalized features, a multi-class linear discriminant analysis (LDA) model is applied to predict the result classes shown in Fig. 6.2. Multi-class LDA is a simple but powerful classification algorithm that can be used to separate multiple classes. The LDA has the advantages of giving linear decision boundaries and requiring less

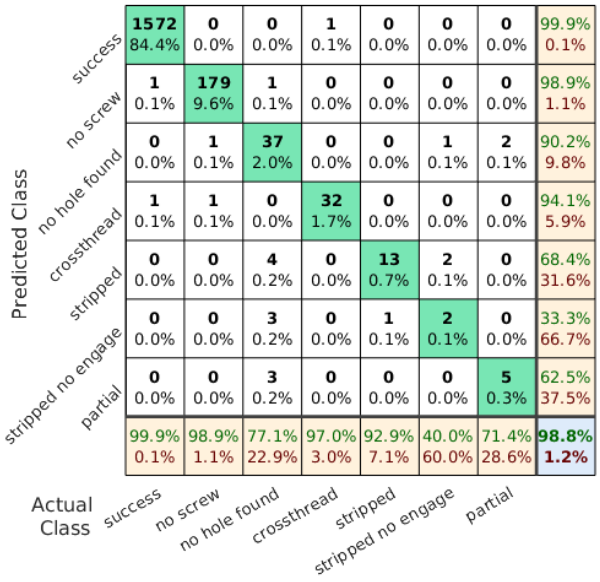

Figure 6.3: Confusion matrix for the LDA model trained with 85 features.

computations. In this method, a set of observation x is classified as the class $\hat{y}$ that has the largest posterior probability $P(Y = y|X = x)$ among all K classes. By Bayes' theorem, this can be written as

$$\hat{y} = \underset{y=1,2,\ldots,K}{\arg\max}\ P(X = x|Y = y)P(Y = y) \tag{6.1}$$

where $P(X = x|Y = y)$ is the probability density function of x and $P(Y = y)$ is the prior probability of class y. For density functions of all the classes, LDA assumes that they are normally distributed with the same covariance Σ but different means $\mu_1, \ldots, \mu_K$, where $\mu_1, \ldots, \mu_K$, and Σ can be directly estimated from the training data. By maximizing the posterior probability, a linear decision boundary can be found for each pair of the classes.

Wit 85 global features, this multi-class LDA method achieves an average classificaton accuracy of 98.93% in 10-fold cross-validation. The confusion matrix of the LDA model of full features is shown in Fig. 6.3.

Sensor Reduction through Feature Selection

Sensor reduction for a more economical screwdriving system is achieved by feature reduction. We follow a standard feature selection approach to reduce the number of required features and sensor signals. Feature selection requires a search algorithm to select candidate feature subsets

and an objective function to evaluate these candidates [38]. In our study, the sequential backward selection (SBS) [37] is chosen as the search algorithm. Starting from the full feature set, SBS sequentially removes the feature that least reduces the value of the objective function. SBS provides a systematic way to sort the original features according to their importance. Note that alternatives such as bidirectional search might give better results; SBS is chosen here due to its simplicity. For subset evaluation, we use the wrapper approach. In the wrapper approach, a feature subset is evaluated by the performance of the chosen learning algorithm. In this paper, for a given feature subset, a multi-class LDA model is trained. Based on the LDA model, the average 10-fold cross-validation accuracy is used as the criterion for evaluation. The overall feature reduction algorithm is as follows:

1. Start with the full feature set $X_0 = \{x_1, x_2, ..., x_N\}$.
2. For every feature $x_i \in X_k$, train a LDA model on $X_k - \{x_i\}$.
3. Remove the feature $\hat{x} = \underset{x}{\operatorname{argmax}} J(X_k - \{x\})$, where J is the 10-fold classification accuracy.
4. Update the feature set as $X_{k+1} = X_k - \{\hat{x}\}$.
5. Goto step 2 and repeat until meeting a certain stop condition (the accuracy and the number of remaining features meet user-specified requirements).

This algorithm helps to determine the optimal feature subset and provides a sensor selection guideline to reduce cost for the final implementation. In addition, fewer features result in reduced complexity and faster algorithm.

To show the whole picture, we set the stopping condition to be $k = N - 1$, i.e., only one feature is left in the final subset. Fig. 6.4 shows the 10-fold cross-validation accuracy of result classification for each optimal subset. The curve is non-monotonic; the accuracy initially increases when some features are removed and drops significantly when very few features are left. Fig. 6.5 shows the variation of the optimal feature set (grouped according to the sensor signals) during the reduction process. We see that most features from F_x and F_y are removed at the first few steps, while features from M_c, F_z and T_z can still achieve quite good accuracy when most features from other signals are removed. The angle feature θ remains during the entire reduction process (not shown in the figure); this explains the importance of angle measurement.

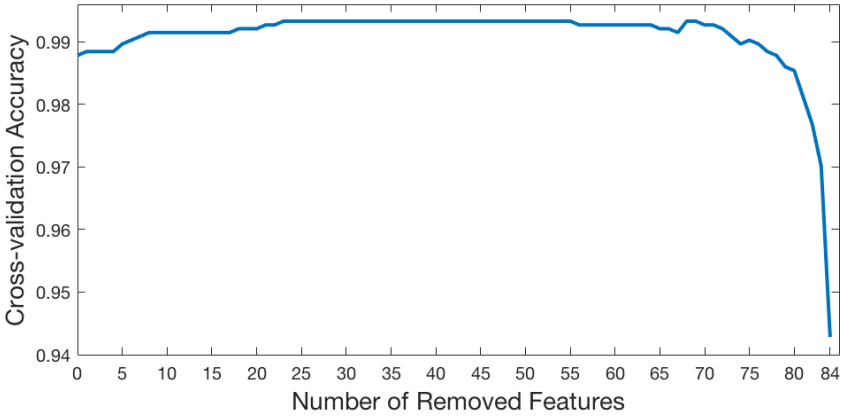

Figure 6.4: Prediction accuracy for optimal feature subsets of different sizes.

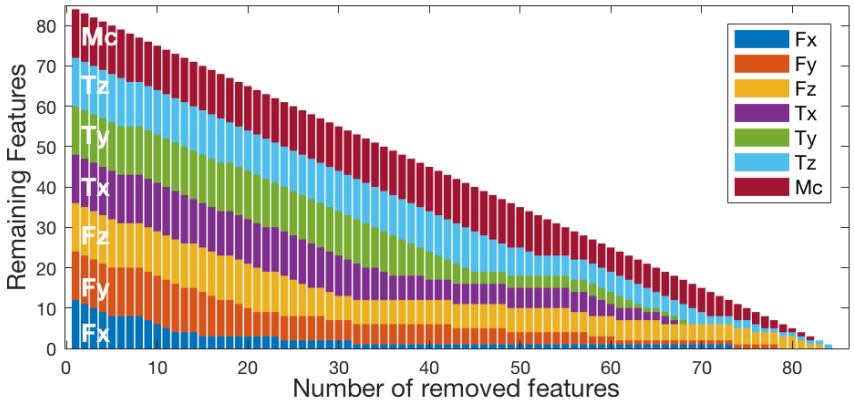

Figure 6.5: Stacked bar graph during the feature reduction process.

This feature selection process indicates that, instead of using all the signals, accurate result type classification can be achieved with only four signals: M_c, F_z, T_z, and θ (angle). Note that we do not need both M_c and T_z. In fact, we can further reduce required sensor signals and cost if the system has lower gear ratio. For low gear ratios (5.8:1 in our system), M_c is almost proportional to T_z, thus we can select M_c (motor current) because it costs less to measure. For high gear ratios, it is necessary to measure T_z because the fastening torque cannot be estimated from M_c due to high gear loss.

For our system, we can remove T_z and reduce the signals to $\{M_c, F_z, \theta\}$. To evaluate the performance, we train two LDA classifiers on the selected signals with and without T_z. We obtain an optimal feature subset consisting of 20 features for $\{T_z, M_c, F_z, \theta\}$ signals and 18 features for $\{M_c, F_z, \theta\}$ signals, as shown in Table 6.1 and Table 6.2, respectively. As shown in Fig. 6.6, even

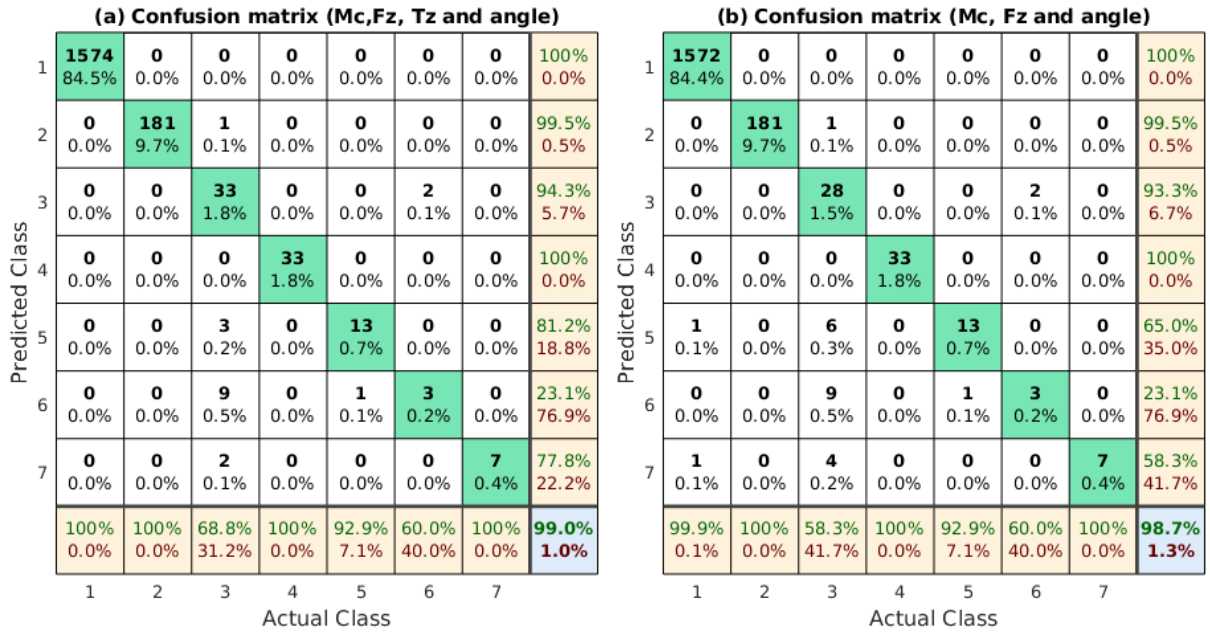

Figure 6.6: Confusion matrices for LDA models trained with 20 (left figure) and 18 (right figure) global features, respectively. Result type numbering: 1.*success*; 2.*no screw*; 3.*no hole found*; 4.*crossthread*; 5.*stripped*; 6.*stripped no engage*; and 7.*partial*.

with fewer sensor signals and features, both classifiers achieve very high accuracy (99.03% and 98.66%, respectively). Their performance is quite similar to the baseline case shown in Fig. 6.3, where 85 global features are used. Compared with the baseline, the reduced LDA models perform better in predicting the *partial* and *stripped no engage* classes; they perform slightly worse for the *no hole found* class.

We see that a linear classifier based on reduced sensor signals can successfully predict the results of screwdriving tasks with very high accuracy. This method can be viewed as an extension of the current industrial method, which only uses the maximum torque and angle to predict success or failure.

6.2 The Role of Compliance in Screwdriving

Screws in most of the electronic devices are usually at very small scales. For example, the dimensions of the smallest screw in an Iphone are less than 1 mm. At such scale, even a tiny misalignment between the screw and the screw hole can cause failure. Therefore, the automa-

Table 6.1: Optimal features for $\{T_z, M_c, F_z, \theta\}$ signals

Order	Feature	Order	Feature	Order	Feature
1	T_z std	8	$\Delta^2 T_z$ std	15	ΔT_z std
2	ΔF_z std	9	M_c std	16	T_z range
3	$\Delta^2 F_z$ range	10	$\Delta^2 F_z$ std	17	ΔF_z mean
4	F_z mean	11	F_z range	18	$\Delta^2 M_c$ mean
5	F_z std	12	$\Delta^2 M_c$ range	19	$\Delta^2 T_z$ mean
6	ΔT_z mean	13	M_c range	20	T_z mean
7	θ	14	$\Delta^2 M_c$ std		

Table 6.2: Optimal features for $\{M_c, F_z, \theta\}$ signals

Order	Feature	Order	Feature	Order	Feature
1	F_z std	7	$\Delta^2 F_z$ range	13	ΔF_z range
2	$\Delta^2 M_c$ range	8	F_z range	14	ΔF_z mean
3	F_z mean	9	ΔM_c range	15	ΔddM_c std
4	$\Delta^2 F_z$ std	10	$\Delta^2 M_c$ std	16	ΔddM_c range
5	ΔF_z std	11	θ	17	ΔM_c mean
6	M_c mean	11	M_c range	18	ΔddM_c mean

tion of the screwdriving process requires robustness and high accuracy to deal with all kinds of exceptions and displacements. Human beings are good at such tasks, because we have dexterous hands. Most importantly, we can rely on our sensing feedback so that when driving the screw, we can adjust the insertion position and force. In this section, we seek to study the role of compliance in the screwdriving process, which could help relax the requirement on the amount of allowable misalignment between the screw and the hole. We use active compliance in x axis and y axis, to help correct the misalignment of the screw and the hole, and thus decrease the error rate in the mininature screwdriving process.

To achieve active compliance, we use a two-axis variable stiffness actuator plate (compliant x-y plate), see Figure. 6.2. Our design idea is from the IBM fine positioner [39] [40], which has lateral compliance in the x axis and y axis and angular compliance in the z axis, supported by an air bearing. Hollis et al. [41] [42] also developed a more advanced compliant magnetically levitated fine-motion wrist, which can provide 6D compliance.

We implemented two control schemes to achieve active compliance. The first one is PD position control with a low stiffness. This system can be seen as a spring and a damper connected

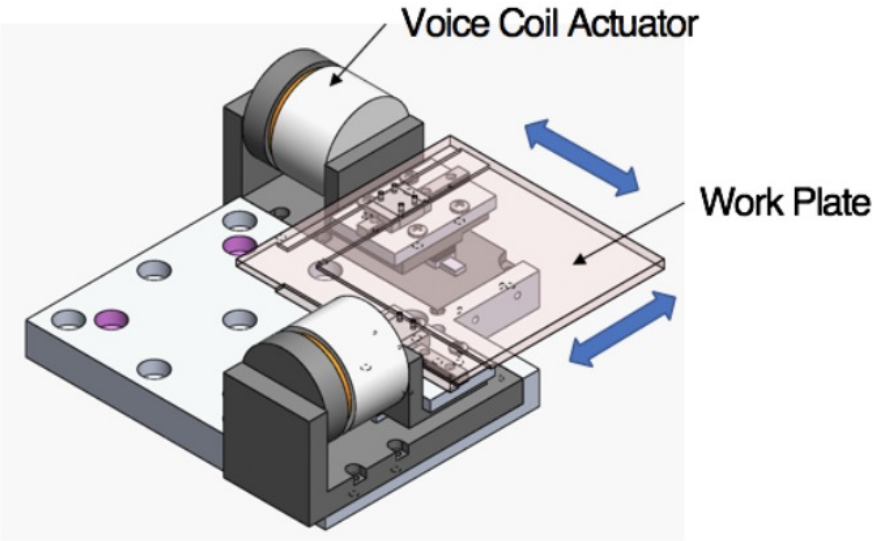

Figure 6.7: our two-axis variable stiffness actuator plate for active compliance control

to the plate on each axis. The videos for our experiments can be found at the following links: video 1, and video 2.

Video 1 is our test for the PD position control. The upper left plot shows the encoder readings from both motors, which are smooth and accurate. And the movement of the plate is compliant. We have conducted experiments on a real screw driving system. As shown in video 2, the plate can compensate for alignment errors by moving compliantly. In our experiments, it can eliminate maximum 0.7 mm misalignment for a screw with the diameter of 1.4 mm. However, unwanted forces still remain if the plate only moves passively.

So, there comes the second control scheme, hybrid force and position control. We aim at minimizing the contact force as well as the position displacement. As shown in Figure. 6.2, we have two parallel controllers, the PI force control and PD position control. The force control loop dominates over the position control loop, thus the position error is allowed while eliminating the contact force. This system can deal with larger misalignments than pure PD position control.

Our force estimation is based on current measurement. The videos for our experiments can be found at the following links: video 3, and video 4.

As shown in video 3, the current reading can roughly reflect the force applied to it. However, the reading is noisy even after filtering. With this kind of setting, the system is not robust enough.

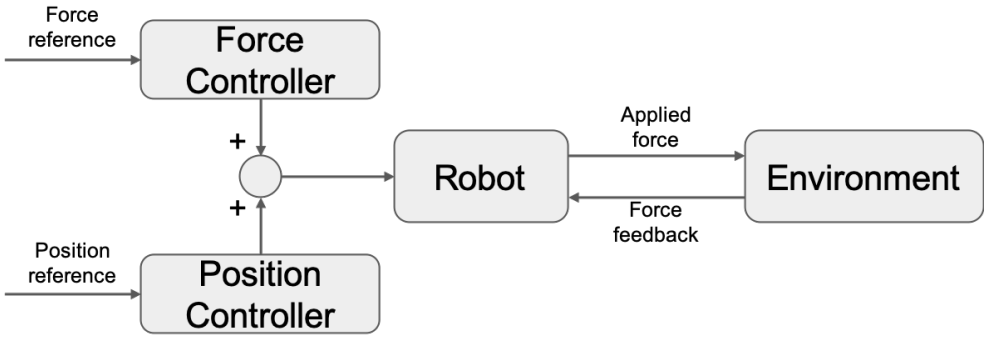

Figure 6.8: Parallel force and position control scheme

But even with this drawback, this system can still outperform the pure position control. As we can clearly see from video 4, the robot can successfully drive a screw in with very large position error.

To conclude, we propose and implement two control schemes to achieve compliance in the autonomous screw driving task. We conduct the experiments on hardware and verified that these two control schemes could fulfill their design goals to ensure successful screw insertions with large misalignments. Comparing these two different control strategies, we found that position control is more stable but less flexible, because of the contact force. On the other hand, current-based force control is more flexible but less stable, because the current measurement is too noisy to be taken as the force estimation.

Chapter 7

Conclusion

We developed a failure detection system which performs stage and result prediction under limited data resources. This system combines known process models with an HMM. With a good estimation of process models, our system can perform unsupervised learning as unlabeled data accumulate, as well as generalize to similar but different screwdriving operations. We show that even with simple prior process knowledge and very limited data, we can develop a robust failure detection system. This method can be extended to processes similar to screwdriving (complex underlying models but limited human knowledge).

Our future works for failure detection include accuracy improvement through human correction, alignment error estimation and recovery strategy development. Data adaptation can be improved if an expert spends a little effort telling whether the predictions are correct. Moreover, predicting real-time stages is not enough. The system should also be able to specify the magnitude of failures, such as alignment error values.

We discuss sensor reduction and compliance control for robotic screwdriving. Our preliminary experiments show that: 1) we are able to use less sensors to achieve even better result prediction than using the full sensor set; 2) compliance control is helpful to relax allowable misalignment in mininature screwdriving.

In the future, towards a fully automated screwdriving system for mini-screws, a low-cost screwdriver will be designed based on more mature sensor reduction analysis. An online stage and result prediction system will be developed through faster and improved implementation of our algorithms. Recovering strategies corresponding to different failure types and stages will

also be designed to improve the overall success rate.

Bibliography

[1] Z. Jia, A. Bhatia, R. M. Aronson, D. Bourne, and M. T. Mason, "A survey of automated threaded fastening," *IEEE Transactions on Automation Science and Engineering*, 2018. (document), 2.1

[2] R. M. Aronson, A. Bhatia, Z. Jia, M. Guillame-Bert, D. Bourne, A. Dubrawski, and M. T. Mason, "Data-driven classification of screwdriving operations," in *International Symposium on Experimental Robotics*, 2016. (document), 1, 1, 2, 2.2, 2, 2, 5.1.1, 5.3, 6.1.1, 6.2

[3] Z. Jia, A. Bhatia, R. Aronson, D. Bourne, and M. T. Mason, "A survey of automated threaded fastening," *IEEE Transactions on Automation Science and Engineering*, Conditionally accepted, 2018. 1, 2, 2, 2

[4] J. L. Nevins and D. E. Whitney, "Computer-controlled assembly," *Scientific American*, vol. 238, pp. 62–74, 1978. 1

[5] J. Nevins and D. Whitney, "Assembly research," *Automatica*, vol. 16, no. 6, pp. 595–613, 1980. 1

[6] L. A. Martin-Vega, H. K. Brown, W. H. Shaw, and T. J. Sanders, "Industrial perspective on research needs and opportunities in manufacturing assembly," *Journal of manufacturing systems*, vol. 14, no. 1, pp. 45–58, 1995. 1

[7] D. E. Whitney, *Mechanical assemblies: their design, manufacture, and role in product development*. Oxford university press, 2004. 1

[8] Z. Li. Robotics research for 3c assembly automation. [Online]. Available: https://app.box.com/s/zcg8qqxt6fw6v4xz22h6 1

[9] (2018) Lenovo recalls thinkpad laptops due to fire hazard. [Online]. Available: https://www.cpsc.gov/Recalls/2018/lenovo-recalls-thinkpad-laptops-due-to-fire-hazard 1

[10] A. Weber, "Automation for small screws," *Assembly*, vol. 55, no. 2, pp. 40–43, 2012. 1

[11] J. H. Bickford, *Handbook of bolts and bolted joints*. CRC press, 1998. 1

[12] ——, *Introduction to the design and behavior of bolted joints: non-gasketed joints*. CRC Press, 2007. 1, 2, 2

[13] ISO, "ISO 5393: Rotary tools for threaded fasteners – performance test method," ISO, Tech. Rep., 2013. 1

[14] S. Wiedmann and B. Sturges, "Spatial kinematic analysis of threaded fastener assembly," *Journal of Mechanical Design*, 2006. 1, 2

[15] X. Cheng, Z. Jia, A. Bhatia, R. M. Aronson, and M. T. Mason, "Sensor selection and stage & result classifications for automated miniature screwdriving." 1, 2, 6.1

[16] R. M. Aronson, A. Bhatia, Z. Jia, and M. T. Mason, "Data collection for screwdriving," in *Robotics Science and Systems, Workshop on (Empirically) Data-Driven Manipulation*, 2017. 1, 2, 6.1.1

[17] E. J. Nicolson, "Grasp stiffness solutions for threaded insertion," Master's thesis, University of California, Berkeley, 1990. 2, 5.1.3

[18] M. A. Diftler, "Alignment of threaded parts using a robot hand: Theory and experiments," Ph.D. dissertation, Rice University, 1998. 2, 5.1.2

[19] 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*. 2, 2, 2, 5.1.2, 5.1.3

[20] S. Smith, "Use of microprocessor in the control and monitoring of air tools while tightening thread fasteners," *Eaton Corporation, Autofact West, Proc. Society of Manufacturing Engineers, Dearborn, MI*, 1980. 2

[21] R. S. Shoberg, "Engineering fundamentals of threaded fastener design and analysis. i," *Fastening*, 2000. 2

[22] R. Isermann, *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*, 2011. 2

[23] K. Althoefer, B. Lara, Y. Zweiri, and L. Seneviratne, "Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks," *Proceedings of the Institution of Mechanical Engineers*, 2008. 2

[24] G. R. Moreira, G. J. G. Lahr, T. Boaventura, J. O. Savazzi, and G. A. P. Caurin, "Online prediction of threading task failure using convolutional neural networks," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2

[25] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Current opinion in neurobiology*, 2006. 2

[26] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Modeling manipulation interactions by hidden markov models," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. 2

[27] L. Rozo, P. Jiménez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intelligent service robotics*, 2013. 2

[28] T. J. Debus, P. E. Dupont, and R. D. Howe, "Contact state estimation using multiple model estimation and hidden markov models," *The International Journal of Robotics Research*, 2004. 2

[29] W. Heemels, D. Lehmann, J. Lunze, and B. De Schutter, "Introduction to hybrid systems," *Handbook of Hybrid Systems Control–Theory, Tools, Applications*, 2009. 2

[30] A. M. Johnson, S. A. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *The International Journal of Robotics Research*, 2016. 2

[31] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *European journal of control*, 2007. 2

[32] *MicroTorque-ToolsTalk MT User Guide*, Atlas Copco. 2

[33] J.-Y. Hwang, D.-H. Jung, Y.-J. Roh, K.-J. Nam, and D.-Y. Hwang, "Low-cost automatic

screw machine using a commercial electric screwdriver," in *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*. IEEE, 2012, pp. 1055–1060. 2

[34] A. I. Automation, "Multi-axis force/torque sensors." [Online]. Available: http://www.ati-ia.com/products/ft/sensors.aspx 2

[35] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 1989. 5.2.1, 5.2.2, 5.2.3, 5.5.1

[36] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112. 6.1.2

[37] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media, 2012, vol. 454. 6.1.2, 6.1.2

[38] V. Kumar and S. Minz, "Feature selection," *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014. 6.1.2

[39] R. Hammer, R. L. Hollis, C. H. An, and F. Henriks, "Design and control of an air-bearing supported three degree-of-freedom fine positioner," *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 677–684 vol.1, 1992. 6.2

[40] R. L. Hollis. Ibm robotic fine positioner. [Online]. Available: http://www.msl.ri.cmu.edu/resources/fine_positioner/ 6.2

[41] R. L. Hollis, S. E. Salcudean, and A. P. Allan, "A six-degree-of-freedom magnetically levitated variable compliance fine-motion wrist: design, modeling, and control," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 320–332, 1991. 6.2

[42] S.-R. Oh, R. L. Hollis, and S. Salcudean, "Precision assembly with a magnetically levitated wrist," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 127–134. 6.2