Data-Efficient Process Monitoring and Failure Detection for Robust Robotic Screwdriving

Xianyi Cheng, Zhenzhong Jia, and Matthew T. Mason *Fellow, IEEE*

***Abstract*— Screwdriving is one of the most prevalent assembly methods, yet its full automation is still challenging, especially for small screws. A critical reason is that existing techniques perform poorly in process monitoring and failure prediction. 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 stage transition graph, enabling our system to generalize across different experimental settings and largely reduce the required data and labeling. In our experiments, a system trained on M1.4x4 screws adapted with very little non-labeled data to three other screws (M1.2x3, M2.5x5, and M1.4x4) with widely varying tightening current, motor velocity, insertion force, and tightening force.**

I. INTRODUCTION

Screwdriving is one of the most common assembly methods [1]. In the consumer electronics industry, hundreds of billions of tiny screws are assembled every year; however, fully automating this huge-volume assembly remains challenging, especially for smartphones [1], [2], [3]. Smaller screws require tighter tolerance and higher alignment accuracy [4]. Moreover, compared to the well-studied peg-in-hole problem (e.g., in [5], jamming diagrams for flexible dual peg-in-hole task were well-studied through large/small deformation stages, providing theoretical basis for control strategy design), screwdriving has more process stages and failure modes; most stages have complicated mechanics involving multiple contacts with highly variant discontinuous surfaces [6]. A system capable of online process monitoring, failure prediction and recovery is necessary for highly automated solutions [3]. 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 [7] or unknown failures. Moreover, previous methods are not guaranteed to work when experimental conditions (e.g., screw sizes) change.

Due to difficulties in screwdriving process modeling, most failure detection systems are essentially data-driven. Directly learning from data yields good results. However, this approach often needs lots of training data and laborious labeling that requires expert knowledge [7]. Getting tons 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" effect [8]: most failure rates are less than $1\% - 2\%$. Hence, can we minimize human effort needed for data labeling and avoid recollecting data when experiment setup changes?

To address the robustness and data-efficiency issues, we propose a system (Fig. 1) that combines process knowledge with a learning method for stage and result classifications in an unsupervised and data-adapting manner. With the sensor signals, we construct knowledge-based process models and feed them into a hidden Markov model (HMM), through which we do stage estimation and rule-based result prediction. The HMM automatically adapts and improves as data feed in. This framework is built on the following facts: the mechanics (Section IV-A) that dominate individual stages and the structure of stage transition graph (Fig. 4) does not change over experiment conditions. Process models (Section IV-A) encapsulate local invariant characteristics (e.g., mechanics for each stage) in their constraint equations, while an HMM incorporates a globally invariant stage transition graph in its transition model. For environment-dependents, process models treat them as identifiable model parameters, while 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 accumulate, the system can adapt 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 distinguish some 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.

II. RELATED WORK

A. 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

The authors are (were) with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: xianyic@cmu.edu, zhenzhong.jia@gmail.com, matt.mason@cs.cmu.edu. Foxconn provides the financial support.

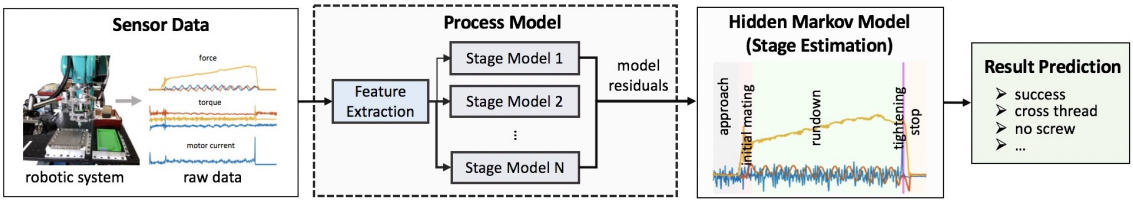

Fig. 1. our online stage estimation and result prediction pipeline

torque-angle curve [9]: *initial mating*, *rundown*, and *tightening*. Recently, [7] presents a more complete stage transition graph, covering multiple process stages and result classes. 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 [10], [6] and [11]. A force spike can be used as an indicator [11] [7]. In *rundown*, quasi-static analysis shows that the oscillation phenomenon is an important signature [12]. In *tightening*, torque, rotation angle and torque-angle gradient [13] are commonly used for failure detection. In Section IV-A, we combine these features with our own analysis to develop screwdriving process models.

B. Fault Detection for Threaded Fastening

In industrial screwdriving, the most common fault detection approach is the *teach method* [1]. Correct torque-angle fastening signatures are collected as reference, and then compared with actual signals using limit check or trend check [14]. The *teach method* is easy to implement, but it lacks flexibility and generality. To overcome these problems, intelligent screwdriving systems are developed, most of which fall into two categories: model-based and data-driven. Model-based methods [15] 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 ANNs [16], SVMs [12], GTC-DF [7], decision trees [3], and CNN [17], directly learn from labeled data. Data-driven methods do not require prior knowledge, but they requires a large amount of data.

C. Discrete State Estimation in Manipulation

A typical robot task often includes a series of discrete states [18], which can be formulated by Markov model or hybrid systems. Our approach is a combination of these two methods. Hybrid models are incorporated to augment the HMM, while the HMM keeps learning and modifying the hybrid model parameters.

The Markov model approach is probabilistic. HMMs are used to perform automatic action segmentation in [19] and [20]. In [21], 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. Hybrid system approach is more deterministic, which

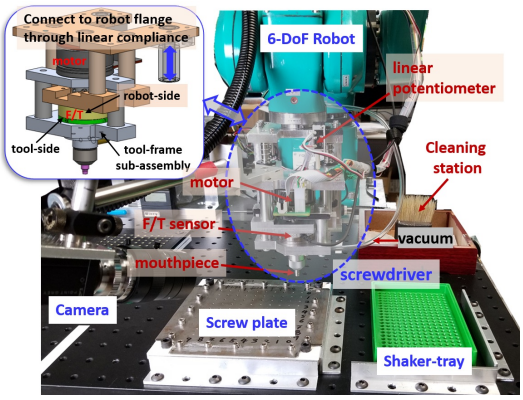

Fig. 2. Intelligent robotic screwdriving system for data collection.

incorporates simplified mechanical models to approximate complex robot behaviors, to reason about mechanisms and to perform feedback control [22] [23]. A discrete state is identified when the state variables fall into the corresponding domain [24].

III. DATA COLLECTION

Our data is collected using the robotic system in Fig. 2. By using a "floating structure" [3] design, the forces and torques exerted on the screwdriver tip can be measured by a 6-axis force/torque sensor. This system is an upgrade of our previous system used in [7] [8], where more detailed descriptions can be found. In each operation, a screw from the tray is first picked up by the screwdriver using vacuum suction. Then the robot moves above the calibrated screw holes on a plate. The screwdriver motor turns on and starts collecting data as the screw insertion starts. Each run terminates when the motor current or motor angle reaches the specified limit. 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 looks like Fig. 3. For this paper, we collected experiment data of different sized small screws and different experiment setups. The detailed information is given in Section V.

IV. METHODS

An overview of our system is as follows. First, all possible stages during the screwdriving process are specified and modeled as constraint equations. Second, the model residuals of constraint equations are used as inputs of HMM, which can be trained with or without labels. Once the stages are

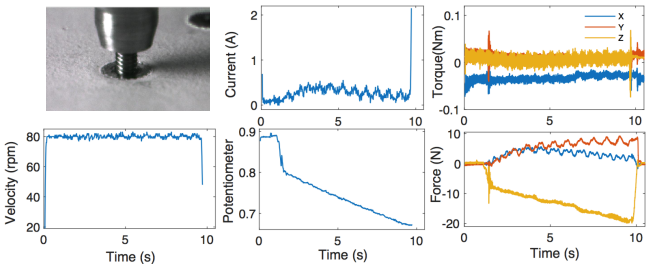

Fig. 3. An example of collected raw sensor data.

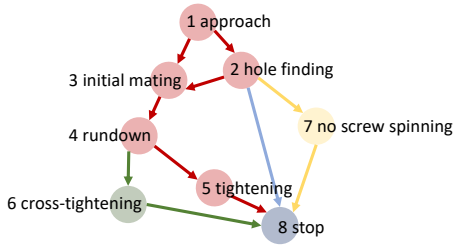

Fig. 4. The stage transition graph in the screwdriving process

predicted by HMM, results (success or different failure types) are inferred from stages using rules constructed 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 adapt to new experiment setups. Fig. 1 shows the pipeline of our online stage estimation and result classification.

A. Modeling of the Screwdriving Process

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.

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. 4) 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. Different types of failures are indicated by their process stages, such as *no screw spinning* and *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 screwdriving process, because we assume each stage has only one dynamic model. Moreover, this graph is used as the HMM state transition model. Note that this graph is modified from our work in [7]. Compared to the previous graph, by exerting proper insertion force during operation, stages and results related to *stripped* happen very rarely, thus can be removed from the stage transition graph and better to be treated as anomaly.

2) Feature Extraction: Feature extraction pre-processes the raw sensor data before stage modeling. We design a feature set based on previous work and our observations. These features enable complicated process models to be transformed into mostly 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 fix-sized interval of time before t:

ν_x, ν_y, ν_z: frequency of the maximum amplitude of forces by discrete Fourier transform. The oscillation phenomenon of the forces on x and y axis is only found in the rundown stage of robotic screwdriving task [12]. Under small alignment error assumption, the frequency approximately equals the screwdriver motor rotational speed.

d_{tip}: the distance from screwdriver tip to the screw hole. It is a strong indicator of the process status. It is calculated as: $d_{tip} = r_z - h_z - l_z$, where r_z is the screwdriver end position on 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, $\Delta_{d_{tip}} = max(d_{tip}) - min(d_{tip})$.

k_{tip}: the gradient of screwdriver tip distance. It is computed by least square estimation for $d_{tip} = k_{tip}t$. 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, $\Delta_{fz} = max(f_z) - min(f_z)$, 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 [11].

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

3) Stage Models: Each stage is modeled as a set of constraint equations using physics or 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: $\mu_{\mathbf{f}} = 0$; $\sigma_{\mathbf{f}} = 0$.

hole finding: the screw moves around without insertion, while the change of screwdriver tip distance is zero: $\Delta_{fz} = 0$.

initial mating: We consider a smooth mating, where the screw is inserted with a constant velocity $C_{1k_{tip}}$ and a constant gradient of force C_{1k_f}: $k_f - C_{1k_f} = 0$; $k_{tip} - C_{1k_{tip}} = 0$.

rundown: vibration phenomenon occurs in this stage, indicating the rotation of the center axis of screw [12].

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}: $v_x - V_m = 0$; $v_y - V_m = 0$; $k_f - C_{2k_f} = 0$; $k_{tip} - C_{2k_{tip}} = 0$

tightening: for a successful tightening, the torque-angle gradient must satisfy a preset value C_{1kta} to ensure correct tension. Since the motor current is proportional to 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: $k_{ta} - C_{1kta} = 0$; $k_c - C_{1kc} = 0$; $d_{tip} - H_1 = 0$.

cross tightening: the same model as *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 [10], thus the screwdriver tip distance can be computed as $H_2 = l \cdot cos(\theta)$, where l is the screw length and θ is the cross-thread angle.

no screw spinning: a 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 being periodically pushed back: $v_z - V_m = 0$.

stop: the screwdriver motor stops, thus the motor current m_c and motor velocity m_v are zero: $m_c = 0$; $m_v = 0$.

B. 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.

1) HMM Representation: An HMM λ is composed of states S, initial state distributions π, state transition distribution A, and observation probability distributions B [25]. When the states are given, 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. 4. 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 of state S_j occurs after S_i, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \le i, j \le N$. If there is no edge goes 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 in Section III. In our method, the $P(O_t | q_t = S_i)$, written as $b_i(O_t)$, is represented in the form of multivariate Gaussian distribution:

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

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

2) Stage Classification: Stage classification is, at the latest timestep T, to find the state sequence that maximize its joint probability $P(q_1, q_2 \ldots q_T \wedge O_1, O_2 \ldots O_T | \lambda)$ with the observation. Viterbi algorithm, a dynamic programming algorithm, can solve this problem efficiently [25]. 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: $\alpha_1(i) = \pi_i b_i(O_1)$; $\alpha_{t+1}(i) = \sum_{j=1}^{N} \alpha_t(j) a_{ji} b_i(O_{t+1})$. 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: $\beta_T(i) = 1$; $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$.

We define the most probable state sequence ended with S_i at timestep t as $\text{mpp}_i(t)$, and its probability $\delta_i(t) = \max_{q_1 \ldots q_{t-1}} P(q_1, q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 \ldots O_t)$. The most probable path computation is:

$$\delta_1(i) = \pi_i b_i(O_1)$$
$$\delta_{t+1}(j) = \delta_t(i^*) a_{i^* j} b_j(O_{t+1}))$$
$$\text{mpp}_j(t+1) = [\text{mpp}_{i^*}(t), S_{i^*}]$$

where $i^* = \arg\max_i \delta_t(i) a_{ij} b_j(O_{t+1})$. During the screwdriving process, at each timestep t, the current state sequence is predicted as $mmp_j^*(t+1)$, where $j^* = \arg\max_j \delta_t(j)$.

3) Data Adaptation: HMM can iteratively adjust to better models as data accumulate using Expectation-Maximization algorithm [25]. 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:

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

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

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

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 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, while the variances corresponding to other state constraint equations are set large.

4) Anomaly Detection: We detect anomaly for unknown stages or impossible path patterns. Unknown stages are those have observation probability very close to zero for each state. Impossible path pattern means no possible transition between the the possible states corresponding to the observations. The two cases both mean that the probability of all the observations is zero, written as $P(O_1 \ldots O_T | \lambda) = \sum_{i=1}^{N} \alpha_T(i) = 0$.

C. Rule-Based Result Prediction

There are five types of results, which can be inferred from stage sequences [7]: *success*, *cross thread*, *no screw*, *no hole found* and *partial*. The rules for result prediction are constructed as follows:

1) If certain failure stages are detected, such as *cross tightening* and *no screw spinning*, the system will predict the corresponding failure types, *cross thread* and *no screw*.

2) If the time of *hole finding* or *initial mating* (with large forces) stage exceeds certain threshold, this indicates 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. 4, then the system will proceed to condition check, which will examine the final insertion length, tightening torque and tightening torque-angle gradient. The result is predicted as *success* if condition check is passed, otherwise *partial*.

D. Generalization

Previously trained HMM cannot be directly applied when the operation conditions change. However, our system can quickly generalize to new experiment conditions. First, we change the setup parameters according to new experiment setup as in Section.II-A.3. Then, learned parameters are estimated from previous data. 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.

V. EXPERIMENTS

A. Dataset

We collected four datasets under different operation conditions as in Table I. The *tightening current* is the motor stop threshold. The *insertion force* is the z-axis force exerted on the screw when screwdriving operation starts. The *tightening force* is the z-axis force at the time when screw tightening finishes. We uniformly sample all data at 100 Hz.

To further evaluate the robustness of our system, for all the dataset, we collect data with both high precision alignment ($1/4$ of total) and random alignment errors ($3/4$ of total). For high precision alignment data, accurate positions of screw holes are given to the robot. The random alignment errors include translational errors (in the range of 0 to 40% of screw diameters) and angular errors (0 to 6° of the screwdriver axis to the screw hole). 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%.

TABLE I

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

B. Implementation

We implemented our system in MATLAB. In 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 [25]. 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 can be initialized as 100 for small screws; these values are empirical. 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 of 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 *if-else* statements.

C. Results

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

1) Stage and Result Classifications: Fig. 5 visualizes the stage classification of our HMM on high accurate alignment data and random alignment error data before and after training. The HMM before adaptation (*HMM before*) only initialized on 10 accurately aligned samples, while the HMM after adaptation (*HMM after*) makes adaptation 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 our HMM adapting to new data as the experiment conditions gradually change. *HMM before* makes reasonable prediction on accurate alignment data for all types of failures. With data adaptation, *HMM after* learns better stage models. For example, in Fig. 5 (a), *HMM after* predicts to extend the period of *initial mating* to exactly where an expert label would be. In Fig. 5 (c) and (d), the predictions for *crossthightening* 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 maintains performance on accurate alignment data.

2) Anomaly Detection: Fig. 6 shows three anomaly cases our system detected. It is even hard for human to tell what happened simply from the signals. By checking the videos, we found that: 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 misalignment. In *anomaly 3*, the data connection was lost for two seconds.

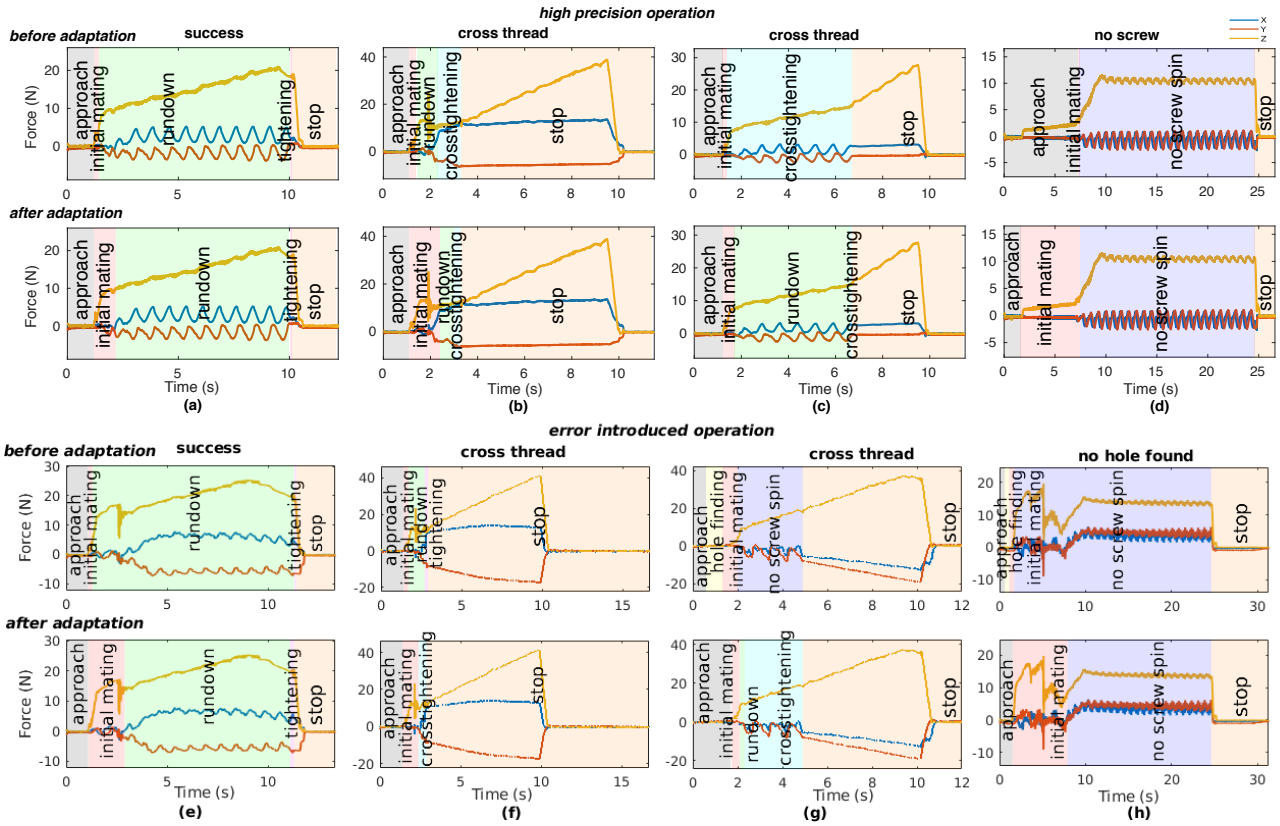

Fig. 5. Illustrative samples of stage classification selected from our dataset. Comparison of the stage predictions before and after data adaptation on high precision operation data and error introduced operation data. Different color blocks correspond to different stages. Only the force sensor data are plotted due to space limitations. The actual result for each operation is shown on the top of the figure.

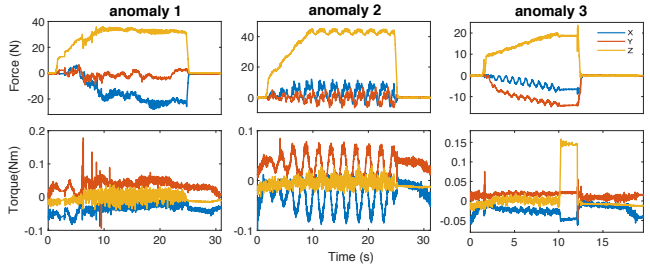

Fig. 6. Anomalies detected by our system. The first row: the force profiles. The second row: the torque profiles.

3) Generalization: As shown in Fig. 7, without any label, our method generalize to *dataset 2*, *3* and *4*. In *dataset 2*, we use tiny screws. 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 size is almost 2 times of that in our original dataset. 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 over four times faster than the original dataset. In real factory scenarios, screwdriving motor speed varies and can be really fast.

D. Evaluation

We evaluate our method using the accuracy of result prediction (Table.II), the proportion of correctly predicted results in all the samples. The correctness of stage classification can be told through visualization. Our 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.

TABLE II
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%

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.

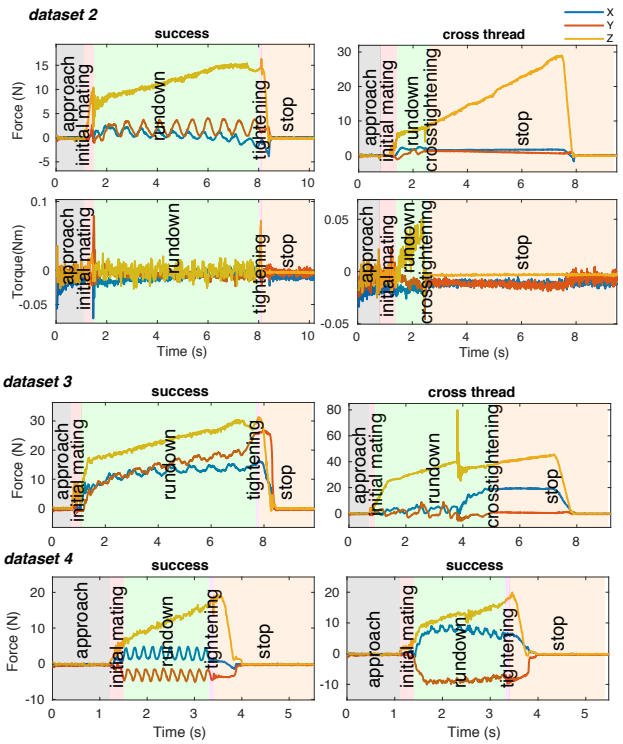

Fig. 7. The visualization of the generalization of our system on *dataset 2* (force and torque profiles), *dataset 3* and *dataset 4* (force profiles only).

E. 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 mean and covariance. 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.

VI. CONCLUSION AND FUTURE WORK

In this paper, we develop a failure detection system which can perform stage and result prediction under limited data resource. This system combines known process models with a 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 work includes 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.

REFERENCES

[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.

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

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

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

[5] K. Zhang, J. Xu, H. Chen, J. Zhao, and K. Chen, "Jamming analysis and force control for flexible dual peg-in-hole assembly," *IEEE Transactions on Industrial Electronics*, 2018.

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

[7] 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.

[8] 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.

[9] J. H. Bickford, *Introduction to the design and behavior of bolted joints: non-gasketed joints*. CRC Press, 2007.

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

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

[12] 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*.

[13] 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.

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

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

[16] 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.

[17] 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)*.

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

[19] 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*.

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

[21] 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.

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

[23] 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.

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

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