Data-driven Detection and Isolation of Wear-induced Faults in Industrial Robots

by

Sathish V

Under the supervision of

Dr. Sachit Butail

Dr. Michal Orkisz

Indraprastha Institute of Information Technology Delhi

February, 2021

Data-driven Detection and Isolation of Wear-induced Faults in Industrial Robots

by

Sathish V

Submitted
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

to the
Indraprastha Institute of Information Technology Delhi
February, 2021

Certificate

This is to certify that the thesis titled "**Data-driven detection of wear-induced faults in industrial robots**" being submitted by **Sathish V** to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by him under our supervision. In our opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

February, 2021
Dr. Sachit Butail

February, 2021
Dr. Michal Orkisz

Indraprastha Institute of Information Technology Delhi
New Delhi 110020

Acknowledgment

I wish to express my sincere appreciation to my academic supervisor, Dr. Sachit Butail and thank him for his immense patience in reading, correcting and constructively criticizing the work with his deep and experienced insights. I am equally grateful to my industry supervisor Dr. Michal Orkisz for his insightful questions and suggestions which helped to strengthen my research contributions. I would like to extend my appreciation and gratitude to other members of my Ph.D. committee, Dr. Mikael Norrlof and Prof. Pankaj Jalote. I express my deepest gratitude to Dr. Mikael Norrlof for his unparalleled support towards completing the thesis. I indebted to him for his guidance in identifying research topic and scaffolding me with his ABB Robotics domain expertise. I am thankful to Prof. Pankaj Jalote who inspired me to do the research in Indraprastha Institute of Information Technology-Delhi (IIITD) and helped through initial phase of the research.

I am grateful to the faculties and administrative staff of the IIITD for providing wonderful academic environment for completing my coursework. My special thanks to Prof. Ashwin Srinivasan for inspiring me to learn and understand the concepts of intelligent systems.

I am acknowledging all my esteemed teachers who inspired me to do research, especially Prof. Rajagopalan Nair during my undergraduate studies and Prof. K Ramamurthy and Prof. Ambalavanan during my master's program.

The work in this thesis would not have been possible without the support of ABB Robotics division. I am thankful to Bernhard Wult who has supported me to set up the ABB simulation software used in the thesis. During my data collection and exploration phase, I received immense support from Gunilla Lonnemark and Roman Trampush to make me understand the data collection process and database structure. With their support, I could sail through the process of data wrangling. I am grateful to ABB GISL India, for sponsoring me and providing support towards my research work. Specifically I am thankful Dr. Srini Ramaswamy who motivated and guided me to join the PhD program and mentored me from industry side along with Prof. Jalote during the inital phase of my research. I would also like to extend my deepest gratitude to Ulf Moberg, who was then the center director of ABB GISL and approved the sponsorship of my candidature for doctoral

programme. The following people from ABB helped through various stages of my research work: Dr. Magnus Larsson, Boris Fiedler, Rajeshwar Datta, Dr. Sudarsan SD, Dr. Divya Sheel Sharma who listened patiently and provided constructive comments on my research work. I also express my sincere thanks to the connected service global management team consist of Cassio Brochado, Josep Reius. Also, thankful to Robotics India management, specifically Subrata Karmakar, Sumer Rathore, and Balaji V.V. I indebted to each one of my colleagues from my team - Service Intelligence Unit who encouraged me through out my research.

My special thanks my IIITD collegues: Dr. Anush Sankaran, Dr. Hemant Agarwal, Dr. Monika Gupta, and Dr. Damodaram Kamma, who supported me without hesitation when in need.

I thank anonymous reviewers and editors for the assistance in improving manuscripts of the research publications.

My family and friends deserve my deepest gratitude for always being there for me. Finally, heartfelt appreciation to my wife Sridevi for her continuous support, and encouragement during the research. It was a great comfort and relief to know that you were willing to provide management of our household activities while I completed my work. My sincere thanks.

Abstract

Industrial robots are complex systems that require technical expertise for condition monitoring and diagnostics. In small and medium scale industries, adequate skilled resources may not be available to monitor the robots consistently. In such cases, domain experts can provide maintenance advisory based on data collected through remote monitoring. The goal of this thesis is to develop and validate data-driven approaches to detect and identify the source of mechanical degradation in industrial robots using data typically collected through remote monitoring.

Performance of data-driven methods is influenced by both data and algorithms. The factors that influence the data include robot application, tasks, and environment. Algorithms differ in terms of the extent to which they rely on an underlying model, and how they weight bias versus variance to isolate anomalous situations. With a remote monitoring solution, knowledge about all the factors may not be available and thus create uncertainty about the data generation process. Therefore, we adopted a two-pronged approach for the study: a) evaluate data-driven methods on simulated data, b) apply the algorithms that worked well with simulated data on real data enhanced with preprocessing methods. For evaluating data-driven methods, we identified wear through significant changes in torque values from normal operating conditions using principal component analysis and studied the effect of source and type of training data on detecting failures in industrial robots.

Towards application of these results on real data, we next formulated strategies to detect the occurrence of wear-induced fault using supervised learning algorithm in a systematic hypothesis-driven study. That study sought to identify effective combinations of four preprocessing techniques on data collected from twenty-six industrial robots. Our results show that preprocessing techniques improved the fault detection performance. Finally, we investigated the problem of isolating the axis of fault by inferring pairwise directional relationships between all axes using an information-theoretic approach called Transfer Entropy (TE). The approach was validated on simulated data generated with an in-house robotic simulation tool. The axis responsible for wear was always detected when the wear was 10% above nominal value. The results of these two studies form the basis for informed data-driven strategies for fault detection and isolation in industrial robots and sets the stage for advanced adaptive detection approaches.

THIS PAGE INTENTIONALLY LEFT BLANK

Research Dissemination

Archival journal articles

J1 **Vallachira, Sathish**, Michal Orkisz, Mikael Norrlöf, and Sachit Butail. "Data-driven gearbox failure detection in industrial robots." *IEEE Transactions on Industrial Informatics* 16, no. 1 (2019): 193-201
Impact Factor - 9.112.

J2 **Vallachira, Sathish**, Michal Orkisz, Mikael Norrlöf, and Sachit Butail. "A Transfer Entropy Based Approach for Fault Isolation in Industrial Robots." *ASME Letters in Dynamic Systems and Control* (2021): 1-23.

Peer Reviewed Conference Publications

C1 **Sathish, V.**, Sithu D. Sudarsan, and Srini Ramaswamy. "Event based robot prognostics using principal component analysis." *In 2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 14-17. IEEE, 2014.

C2 **Sathish, V.**, Srini Ramaswamy, and Sachit Butail. "A simulation based approach to detect wear in industrial robots." *In 2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1570-1575. IEEE, 2015.

C3 **Sathish, V.**, Srini Ramaswamy, and Sachit Butail. "Training data selection criteria for detecting failures in industrial robots." *In IFAC International Conference on Advances in Control and Optimization Of Dynamical Systems,* volume 49, pp. 385-390. IFAC, 2016.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Abbreviations

ANOVA	–	Analysis of Variance
AUC	–	Area Under Curve
CC	–	Cross Correlation
COV	–	Coefficient of Variation
DOF	–	Degree of Freedom
ES	–	Ensemble Stacking
FDI	–	Fault Detection and Isolation
FPR	–	False Positive Rate
KL	–	Kullback-Leibler
LR	–	Logistic Regression
OTM	–	One-to-many
OTO	–	One-to-one
PC	–	Principal Components
PCA	–	Principal Component Analysis
RF	–	Random Forest
ROC	–	Receiver Operating Characteristic
SVM	–	Support Vector Machine
TE	–	Transfer Entropy
TPR	–	True Positive Rate

List of Symbols

τ_k^i	Input torque for axis i at time step k
ω_k^i	Disturbance for axis i at time step k
ν_k^i	Measurement noise for axis i at time step k
$c(\delta t)_{\tau^j \tau^j}$	co-variance at lag δt for two time series τ^i and τ^j
$R(\delta t)_{\tau^j \tau^j}$	Cross correlation coefficient for the time series τ^i and τ^j
λ^{-1}	Diagonal matrix with inverse eigen values for k principal components
$p(x)$	Probability distribution function
$D(p\|\|q)$	Kullback-Leibler divergence between $p(x)$ and $q(x)$
P	Principal Component Coefficient matrix
T	Score matrix of principal components
E	Residual Error matrix
Λ	Inverse gear ratio matrix
$\boldsymbol{\phi}$	Vector of robot joint angles
$\dot{\boldsymbol{\phi}}$	Vector of robot joint velocity
$\ddot{\boldsymbol{\phi}}$	Vector of robot joint acceleration
$\boldsymbol{\tau}$	Vector of robot joint torque
$T_{i \to j}$	Transfer entropy from process i to j
$netT_{i \to j}$	net transfer entropy from process i to j

Part I

Introduction

Chapter 1

Introduction

Industrial robots are significant contributors to increased productivity, quality and safety in automated manufacturing. Therefore, ensuring continuous availability of the robots is a non-negotiable demand from the industry. Industrial robots are widely used in applications such as automotive, electrical and electronics, food and beverages, packaging and palletizing and welding.

This thesis deals with data-driven fault detection methods and the effect of preprocessing methods on detection performance. Simulated as well as real robot data are analyzed to detect faults caused by wear of the mechanical parts of the industrial robots. While such wear processes may take several years to affect the performance of robot, once they appear, wear-induced faults can be caused rapidly [1]. An early detection of excessive wear can help the industries to plan their maintenance activities so that the influence on the production is minimized. This chapter presents an introduction and motivation to the problem of detecting wear-induced faults, the main research contributions of the thesis and an outline of the rest of the thesis.

1.1 Motivation

Unexpected breakdown of robots can lead to loss of production time and result in considerable financial losses. Therefore, timely maintenance of robots is very crucial. In practice, robot monitoring is focused on protection and safety and not many commercial

solutions are available for monitoring the health of the robots [2].

The requirement of high availability of robots is typically attained through preventive maintenance schedules. These schedules are arrived on the basis of estimated life expectancy of various robot components - the actual condition of the robot is typically not taken into account [3], a scenario that leads to high maintenance costs. Condition-based maintenance is an alternative strategy that uses the data collected from various parts of the robots to assess the condition of the robot. ABB's Connected service offering is one of the available commercial approaches to monitor the robots remotely. In a remote service set-up for industrial robots, information such as events and measurements from important variables in the system are sent to a remote database for hundreds of robots located at customer sites [4]. This information is further processed to prepare advisory reports that assist field engineers or customers to prioritize their maintenance activities. Such arrangement is economically viable if the users of the robots are not experts in identifying problems with robots (Fig. 1-1). The value of remote monitoring can be enhanced by improving the capability of predicting faults across multiple robots.

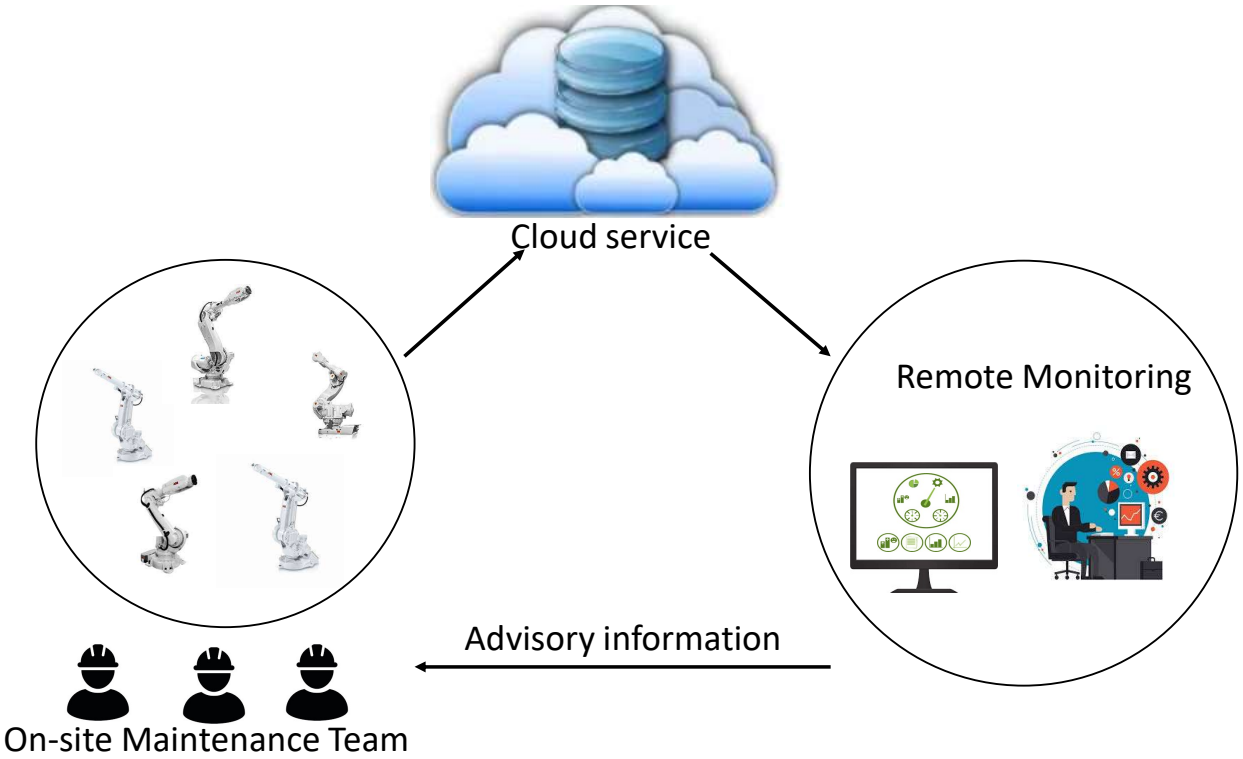


Figure 1-1: Remote monitoring of industrial robots.

1.2 Problem Statement

Robot manipulators are composed of links connected by joints into a kinematic chain (Fig. 1-2). Joints are typically rotary (revolute) or linear (prismatic). Robot manipulators can be classified by several criteria, such as their power source, or way in which the joints are actuated, their geometry, or kinematic structure, their intended application area, or their method of control [5].

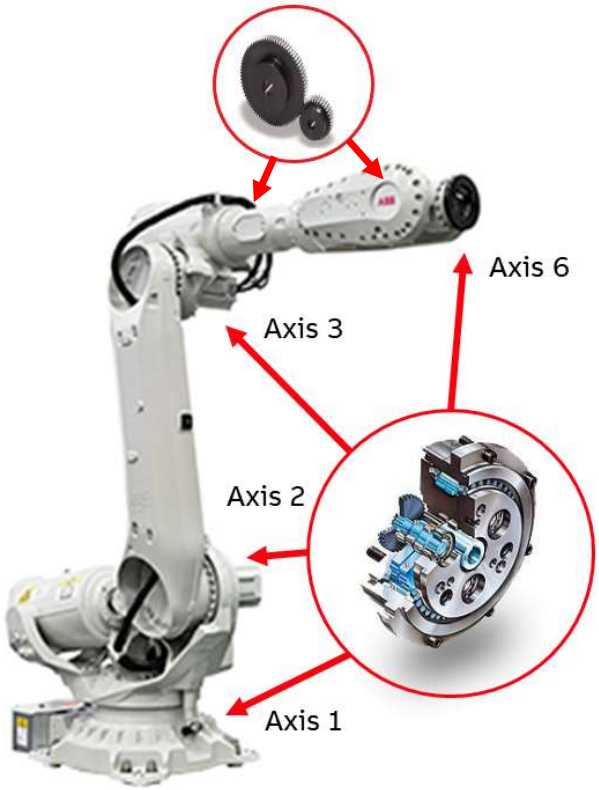


Figure 1-2: An ABB industrial robot (IRB 6700)

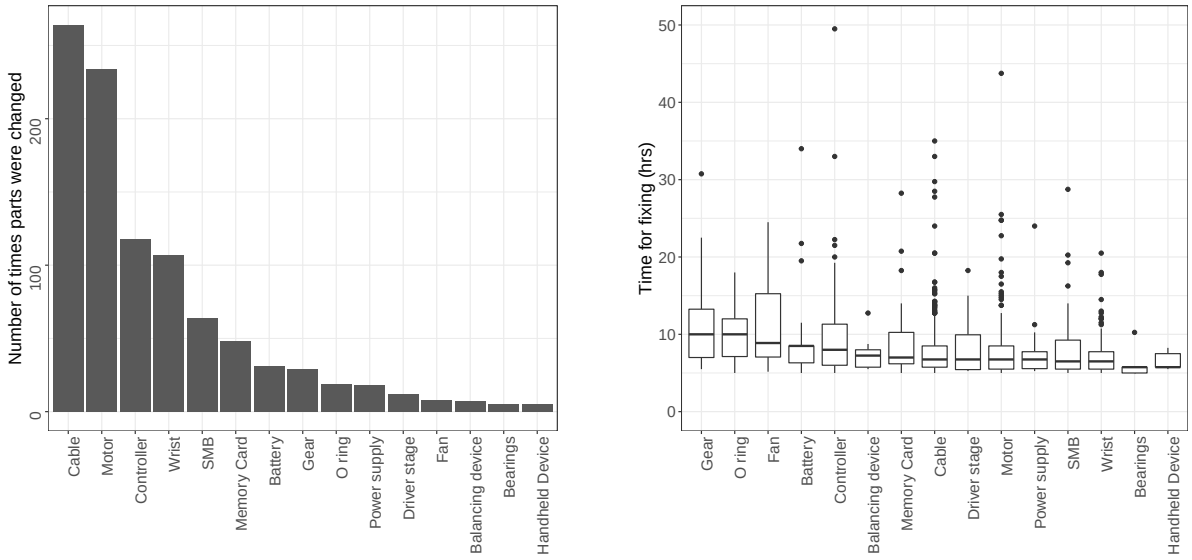
A typical robot drive chain consist of bearings, gears and ball-spindles. The regular operation of the robots can lead to wear and tear in these components. Similar wearing effects are present in various peripherals that include moving parts such as motors, transmission systems, and measurement systems [6]. A typical industrial robot consist of 6 gearboxes in each of the six axes (Fig. 1-2). The characteristics of link joint strongly influence the dynamic behavior of robots. In order to perform a task by robot, each joint requires a transmission through geared or ball-screws to drive the end-effector at a required speed. The work load

can impose severe stress on the transmission system that results in wear, lead to increased backlash, and eventually degrade robot performance. The transmission system is the weakest link in the robot reliability model [6] and most robot failures in manufacturing plants are due to wear in the gears of transmission system [7]. The most common failures in gearbox are: a) bearing failure - due to excessive loads on bearing and improper lubrication, b) gear failure - contaminated oil within gearbox causing micro pitting or fretting, misalignment of gears, excessive heat, and c) sealing failure.

Spare part information from failure cases reported from ABB robots from a specific region over a period of one year are analyzed for this study. The critical components are classified in terms of frequency of change and cost of downtime (information proprietary to ABB). The distribution of spare parts with respect to frequency of change and working time for replacement is shown in Fig. 1-3a. The median working time for replacing gearbox is highest among all spare-parts of an industrial robot (around 10 hrs) (Fig. 1-3b). The most frequently changed spare parts include cables, motor and controller. The controller has many sub components like main board, i/o card, controller module etc., these sub components are aggregated as a single count in the chart for a concise representation. Since many of the failures in electrical/electronic components are abrupt in nature and difficult to predict, only mechanical failures of the robots are considered in this study. Therefore, with focus on avoiding failures, data-driven approaches are used to predict excessive degradation caused by wear of the mechanical parts and preceded by warning in terms of performance degradation. In this study, wear induced mechanical degradation such as gearbox failures are considered.

Once the type of failures - wear induced failures - were identified for the investigation, all the measurements and event logs, minimum 30 days prior to failure for all the failed robots were extracted from connected service database. Significant effort is spent on understanding the data generation process, familiarizing database structure and evaluating preprocessing methods.

The research objective is to enhance the capability of data-driven fault detection with actionable lead time using data collected with the remote monitoring set-up. The following questions are formulated to address the objective along with the peer-reviewed publications that describe the associated studies (C# refers to peer-reviewed conference paper and J#



(a) (b)

Figure 1-3: Sample spare part change information: a) Frequency distribution most commonly changed spare parts, b) Box plot of repair time for replacement of spare-part

refers to archival journal paper):

Q1: What is the effectiveness of principal component analysis (PCA) in terms of detecting the onset of wear induced failures in industrial robots? (C1, C2)

Q2: Can the effectiveness of failure identification using standard classification methods be improved by curating the data using pre-processing methods? (C3, J1)

Q3: Can correlation and information theoretic methods be used to isolate the source axis of the failure? (C2, J2)

The above questions are addressed by analyzing data using time-series methods, applying preprocessing techniques to develop machine learning based fault prediction models and evaluating the effectiveness of the preprocessing techniques on model performance. An information theoretic approach is further developed to locate the source axis of failure.

1.3 Contributions

The contributions of this thesis are:

- PCA results are evaluated in terms of their ability to detect the onset of failure on real and simulated data (C1, C2, C3) and also as an effective pre-processing approach for developing data-driven models (J1).

- The effect of different preprocessing techniques on fault detection using standard machine learning methods is evaluated. In a systematic hypothesis-driven study four different preprocessing techniques are applied to training data , collected from 26 industrial robots from the field were found to affect the performance of data-driven fault-detection methods (Fig.1-4) (C3, J1).

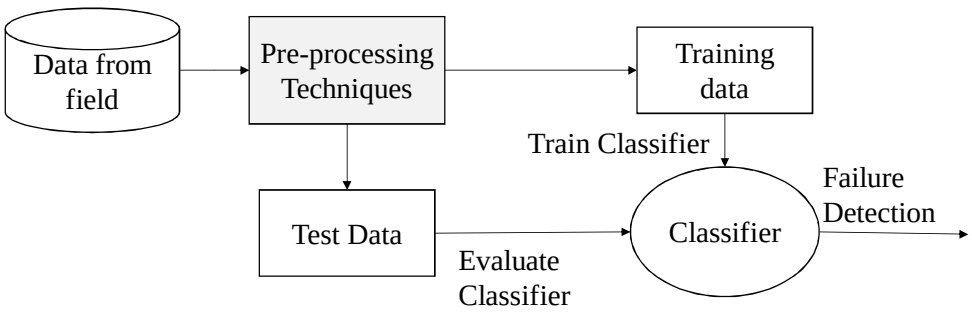


Figure 1-4: Research objective - evaluate the effect of data preprocessing techniques on failure detection in industrial robots.

- Correlation-based techniques are found to be less effective in isolating source axis of fault. An information theoretic based approach is developed to detect source axis of the failure in multi axes industrial robots (Fig. 1-5). Specifically, wear induced failures are simulated in an in-house robotic simulation tool, and the source axis of failure is identified as one which has the highest net transfer entropy (TE) across all pairs of axes (J2).

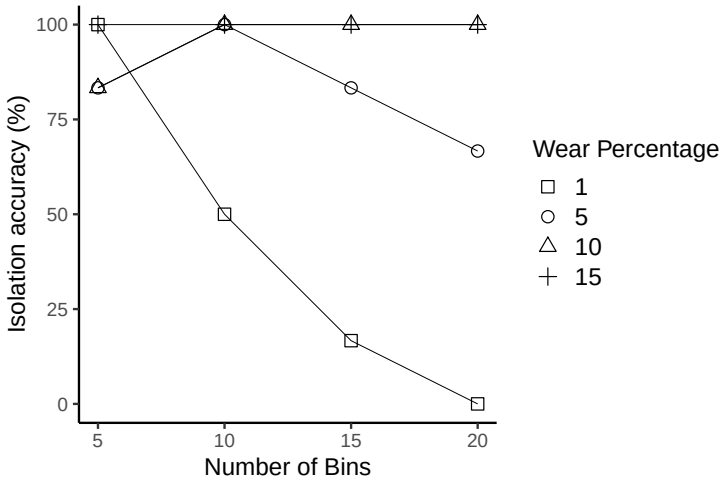


Figure 1-5: Accuracy of isolating source axis of wear using TE approach with varying wear percentages for a specific simulated task.

1.4 Outline

This thesis comprises five self-contained research articles with relevant literature reviews and it is organized in four parts, each with one or more chapters. Part 1 consists of Introduction, part 2 is based on material from three peer reviewed conference publications updated to address comments from thesis examiners, part 3 is based on material from two archived journal articles, and finally part 4 concludes the thesis with a summary of work and future directions.

Data such as event logs and measurements from robot controllers were collected using the existing remote data collection solution. Many events are getting generated with respect to predefined thresholds on various measurements and analyzing individual events may not be an optimal way to detect the failure. In the first paper $C1$, a Principal Component Analysis (PCA) based approach is adapted on event logs to detect a gearbox fault. The proposed method reduce the dimensionality of the original data which consist of correlated events while retaining the variation present in the data. An abnormal pattern was observed within 30 days before failure by using multivariate statistics: Hotelling T^2, Q Residuals and Q contribution chart [8]. However, this approach of using event logs could not be generalized as there exist variation in configured events across robots.

In second paper $C2$, a method is proposed to detect the onset of wear and subsequently locate the source axis of wear using the simulated data. Wear is simulated by increasing friction coefficients of different axes in a PUMA 560 robot model available in MATLAB Robotics toolbox on three separate tasks. Load disturbances and measurement noise are added in the simulation to evaluate the robustness of the wear detection. Wear in robot is detected using principal component analysis, and the source axis is located by cross-correlating the output data from all axes through time.

In third paper $C3$, the effect of source and type of training data are studied with Principal Component Analysis (PCA) on detecting field failures in industrial robots. Specifically, using field data across multiple robots performing different tasks, two scenarios are compared: first, where training data obtained from a single robot is used to evaluate multiple robots (one-to-many), and second, where each robot is evaluated on the basis of its own training data (one-to-one). The effect of data preprocessing prior to running PCA on ability to detect and predict failures is investigated. To reduce task dependence of the raw signal, the signal is preprocessed by computing the absolute difference between successive measurements and the results are compared with a PCA model that is built using raw signal alone and another that is built from a combined signal having both raw measurements and their absolute difference. The effectiveness of detecting failures is quantified in terms of three measures: coefficient of variation of the Q-residual obtained by projecting the test data on the PCA model, number of samples above a data-driven confidence threshold, and lead time, measured as the number of days prior to failure when the residual error rises above a given threshold. The preprocessing of signal has been found to have significant influence on detecting failures.

In the fourth paper $J1$, data from 26 robots are used to study the effect of preprocessing techniques on gearbox fault detection performance using machine learning models. The performance data is collected from robots with gearbox faults occurred over a period of 4 years. Standard machine learning algorithms are limited in their ability to detect gearbox failures, mainly due to task variability arises from robot-specific data. To improve detection performance of machine learning approaches, pre-processing techniques are proposed to curate the data prior to building a classification model. In a systematic hypothesis-driven study exploring the effect of different pre-processing techniques, training data augmentation

with estimated measurements, data differencing to suppress task dependence, inclusion of local variation, and selection of principal components from the robots are evaluated. The pre-processing techniques have resulted in improving the failure detection performance.

In the fifth paper $J2$, a method is proposed to isolate the axis of failure in industrial robots. With implementation of such method, the service time and field visits are expected to reduce further. The problem of failure isolation was cast as that of causal analysis within coupled dynamical processes and hence, the efficacy of the information theoretic approach of transfer entropy is evaluated. Wear induced failure is simulated in an in-house robotic simulation tool, and source axis of failure is identified as one which has the highest net transfer entropy across all pairs of axes. To compare these results with an existing standard approach, the torque signal is correlated for each axis across time to identify the source axis of the wear. Compared with cross-correlation, the average source axis detection across the three tasks with TE is twice as effective at friction coefficient 10% above the nominal value.

Part II

Peer reviewed conference proceedings

2

Event Based Robot Prognostics using Principal Component Analysis [1]

Abstract: As industrial systems are getting complicated, challenges in coming up with efficient maintenance strategies which include predicting failures in the system become important industry specific research topic. Traditionally, research focuses on developing failure prediction models based on physical understanding of the system. But, development of such models are often time consuming and labor intensive for complex systems. In recent past, due to advent of cheaper data collection mechanisms and efficient algorithms, data driven approaches for predicting failures are gaining significant interest in industrial research community. In this paper, we provide a Principal component Analysis (PCA) based approach of failure prediction in industrial robots using event log information. The event logs are collected through remote service set-up from a robot controller. The proposed method will reduce the dimensionality of the original data which consist of interrelated events while retaining the variation present in the data. Using PCA and multivariate statistics such as Hotelling T^2, Q Residuals and Q contributions charts, we are able to detect abnormal behavior of event pattern within 30 days before failure.

[1] This chapter is a slightly modified version of paper published in 2014 IEEE International Symposium on Software Reliability Engineering Workshops and has been reproduced here with the permission of the copyright holder

2.1 Introduction

High reliability and availability are essential for industrial products. Intelligent prognostics and diagnosis methods are very important in maximizing productivity, ensuring high product quality and reducing the maintenance cost. Positive impact of this approach will be reduction in total cost of ownership of the product[9]. The main functions of these systems include detection, isolation prediction and correction. From the end-user point of view, the distinction of prognostic with diagnostic is the availability of lead time to useful life or failure. Identification of the remaining useful life will depend on the system complexity and application domain [10] and this study focus on the failure progression time-line. The ability to predict failure helps customers to plan their downtime for maintenance. This avoids (at best) or reduces (at worst) production losses. This strategy is more efficient compared to the traditional periodic maintenance where maintenance is performed whether it is really needed or not and help us to move up the value chain by performing need based maintenance as against the current practice of periodic maintenance.

2.2 Related Work

There are various approaches discussed in literature about prognostics and diagnostics of industrial systems. Techniques for condition based maintenance can be classified into two categories: diagnostics and prognostics [11]. Diagnosis deals with fault detection, isolation, and identification of faults when they occur. Prognostics, however, attempts to predict faults or failures before they occur. Simply stated, while prognostics is proactive, diagnostics is post-mortem. There are various prognostic approaches based on physical, data driven and experience models.Roemer et.al.[12] summarizes the range of possible prognostic approaches and their cost implications. As our current study is based on data driven approaches to predict faults in an industrial robot, we will discuss the related work in this section.

Data driven methods use a wide variety of data processing and analysis techniques such as multivariate statistical methods, cluster analysis, spectral analysis etc. One of the basic multivariate method is Principal Component Analysis(PCA) and PCA is simple and easy

to implement. PCA has been widely used in the area of signal processing, fault diagnosis, pattern recognition in various fields and provides good results [13].

Some of the related work in which fault detection is based on usage of PCA for predicting faults in reciprocating compressor[14]. Features are extracted from vibration signals collected from the compressor and used for model building and consequently for fault prognosis. In another study [9], PCA is used for gearbox fault diagnosis. On-line monitoring is difficult as this approach required pre-processing of vibration signals. But in case of process industries, measurements collected are directly used for model development [15]. The PCA model is applied to new data and then, associated T^2 and Q statistics are calculated and used for monitoring. Any deviation of these values from the confidence limits indicate an aberration in process performance and alarm is raised to process operator [16]. Lin et.al. [17] recommends the usage of PCA for fault diagnosis as the characteristic parameter of traditional methods cannot effectively reflect the trend of mechanical fault when the equipment load changes significantly. As the demand of practical use of PCA is growing, researchers have studied and discussed many improvements on PCA [13],[9],[18]. However, in our current study we use the PCA without any modifications. Most of the related work is based on the data collected on a continuous basis and for a continuous process. In our study that we conducted on failure prognosis of industrial robots by analyzing discrete event logs. In earlier studies, researchers [19] have developed model-driven methods for fault isolation in robots using discrete event logs. However, application of data-driven methods such as PCA on discrete event logs from real world robot systems makes the study unique compared to previous work.

2.3 Approach

2.3.1 Problem Definition

Industrial robots are widely used to automate repetitive tasks and thereby increasing productivity. In order to maximize the productivity, minimizing the downtime of robots due to maintenance or repair is essential. Robot manufactures provide various maintenance strategies to help their customers. In ABB, one of the important service offering to our

customers is remote service which involve monitoring the robots remotely and provide advisory and assistance to minimize the downtime of robots. For a complex system like robots, rule based alarms and warnings may not be enough to predict failures or planned maintenance. With focus on service, it is important to detect the change in condition before the critical degradation occurs and subsequently perform timely maintenance. Wear in robot joint can lead to a degradation of performance and to an eventual failure. Wear typically develops slowly with usage and it can be detectable in an early stage using data-driven methods. However, faults such as electronic faults and broken cables cannot be detected with the data that is collected using remote service solution. In the present study, we attempt to use data driven approaches to predict wear induced failures which can occur through the life of an industrial robot.

2.3.2 Data Collection

In this study, we collected data from two sources. One source is related to *Field failure information* about robot controllers reported by service engineers and the second source is the event logs which are collected automatically in a remote server from robot controllers. Field failure information is gathered at the time of failure. This information includes, serial number of robot which failed, date and time of failure and a brief description of the failure. These details are reported manually by service engineers while addressing the issue at the customer site. On contrary, events are specific occurrence of a thing which generate an item in the log. For instance, if the manipulator collides with an obstacle, this will cause a message to be sent to the log. A typical event log consist of event number, event title, event time, description of event, consequence and probable actions [20]. In this study, we have identified a specific failure of Robot and studied how the event pattern changes in normal and failure condition. The failure date is super imposed with event log data in time domain so that we can observe the pattern based on events before failure. One month data of event log is considered for assessing the pattern of normal working condition of robot. Events are getting collected in the remote service database based on the event of interest configured by service engineers in agreement with customers. When an event of interest (alarm) happens, a collection of events along with the alarm are sent to remote database. In this study, about

7 months event data are used for analysis. The frequency of event types are calculated on a daily basis. The data set consist of daily count for each of the sixty-seven events. It is noted that if we analyze the frequency of event types on an hourly basis, the resulting data leads to sparse matrix. We also observed that not all event types generated during normal and failure condition are same. This was solved by introducing zeros for those variables which do not appear in either before or after failure. The number of distinct event types considered for this study is sixty seven.

2.3.3 Analysis

Initially, we set a hypothesis that the events pattern near failure and normal condition are different. But, it was a challenge for us to identify the number of days before the failure where the event log pattern changes. We tried clustering analysis assuming that pattern near to failure will be different from that away from failure. But, k-mean clustering did not reveal any pattern near failure compared that of away from failure. We used multivariate monitoring technique such as PCA which accounts for correlation between parameters and therefore alert when relationship between parameters change as well as mean level of individual parameter change. Since we do not know exactly the number of days before the changes happen in the event log distribution, we decided to consider all six month data before failure as failure data. In the following section, we discuss about the PCA approach and our results.

2.3.3.1 Model development using PCA

The objective of PCA is to capture variability of the data for identifying components which explain maximum variability of the data. This is done by transforming the original data into a set of uncorrelated variables called Principal Components(PCs)[21]. This helps to reduce the noise in the data. PCA is also used as a pattern finding technique.

Let the variables or events in our case, be m dimensional data set: X $=x_1, x_2, x_3, ...x_m$, PCA decomposes the event vector X, into new direction P as [22]

$$X = TP^T = \sum_{i=1}^{m} t_i P_i^T \tag{2.1}$$

Where P_i is an eigen vector of the covariance matrix of X. P is known as principal component coefficient matrix and T is defied as score matrix of PCs. The matrix P gives information about the contribution of each variable to a specific PC. Score matrix represents the data X in PC space and reflects the relationship among the samples(rows in X). In PCA, it is expected that relatively few components will account for maximum variance in the data. Therefore, there will be minimal information loss by choosing on few PCs expressed as equation 2.2 [15],[14].

$$X = TP^T + E = \sum_{i=1}^{k} t_i p_i^T + E \tag{2.2}$$

where E denotes residual error matrix. Cumulative variance of first 5 PCs explain 85% of variability [17], and this means that the subspace consisting of 5 PCs contains enough variation of the original features (event types). Here, k is 5 as we selected only 5 PCs.

Events data collected during normal working condition is used for developing a PCA model and test is performed with the event data before failure.

2.3.3.2 Fault Detection using PCA Model

In PCA based anomaly detection, Hotelling T^2 and Q Statistics are used. The use of Q statistic and T^2 statistics are complimentary. Hotelling T^2 provides the overall variation from the center of data set and Q statistic explains how well the PCA model fits to the given data[16].

For new measurement feature vector X, the T^2 statistic can be found from equation 2.3[15]:

$$T^2 = x^T P \lambda^{-1} P^T x \leq T_\alpha^2 \tag{2.3}$$

where $100(1-\alpha)\%$ control limit for T_α^2 is calculated by means of F distribution as in equation 2.4. λ^{-1} is a diagonal matrix with inverse eigen values for k PCs used for the

model [22]:

$$T_\alpha{}^2 = \frac{(k(m-1)}{m-k} F(k, m-1; \alpha) \tag{2.4}$$

where $F(k, m-1; \alpha)$ represents F distribution with k and m-1 degrees of freedom, α is the level significance α, k is the number of PCs retained in the model, m is the number of samples used to develop the model.

Lack of model fit can be calculated with Q statistic and is expressed as sum of squares of each row of E (refer equation 2.2) as

$$Q = X(I - P_k P_k^T)X^T \leq Q_\alpha \tag{2.5}$$

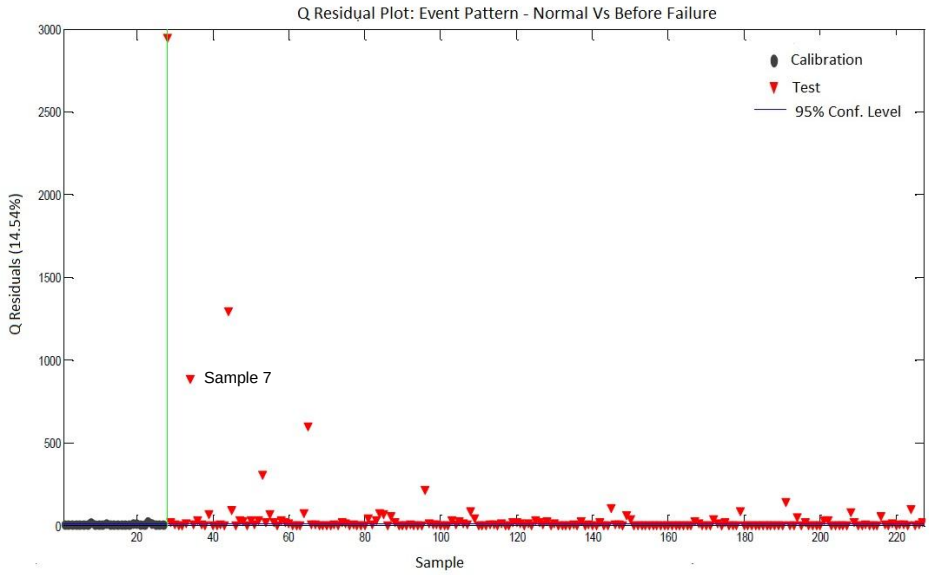
Let the eigenvalues λ_i of the covariance matrix of X and with $100(1-\alpha)\%$ confidence level, upper control Q_α is calculated as in equation 2.6 [15] :

$$Q_\alpha = \theta_1 [\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}]^{\frac{1}{h_0}} \tag{2.6}$$

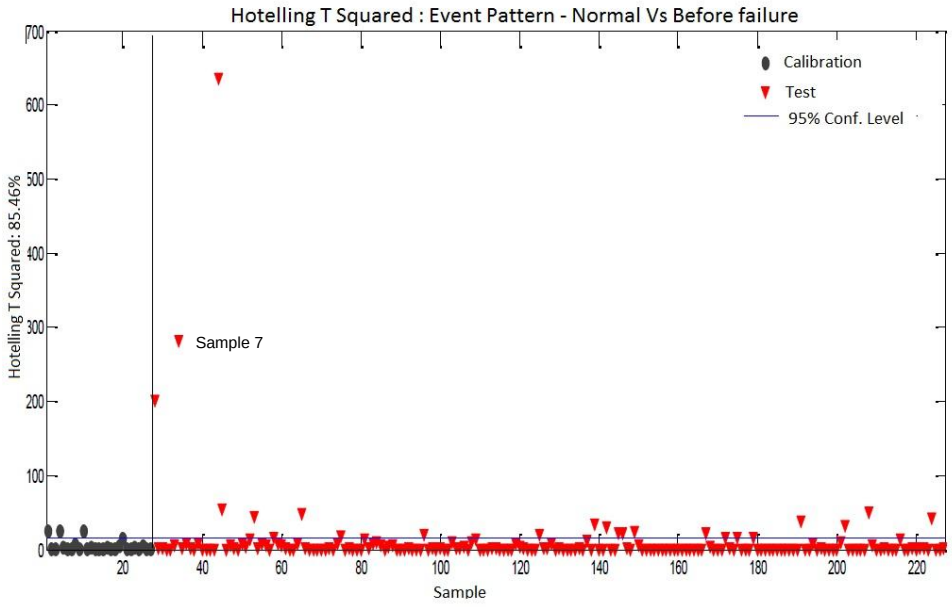
where :

$\theta_i = \sum_{j=k+1}^{m} \lambda_j^i$, $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$, and c_α represent standard normal deviate corresponding to upper (1-α) percentile.

The new event pattern can be detected using Q or T^2. When an unusual event pattern occurs and it results in change in structure of covariance matrix which will be detected by high value of Q. The plot has first twenty eight samples that belong to normal working condition of robot and event frequency is ordered from the failure date to seven months before failure. In Fig. 2-1a, significant increase in value of Q residual was observed between samples twenty eight to sixty where twenty-eighth sample represent the failure date and sixtieth sample represent thirty two days before failure (right side of the vertical line). Similar behavior is also observed with Hotelling T^2 plot as shown in 2-1b. This information indicates that event pattern indeed can be used for predicting the failure.



(a)



(b)

Figure 2-1: Fault Detection using PCA Model: a) Q Residual plot, b) Hotelling T Squared plot.

Once we identified the samples which caused the abnormal behavior using Q- residual and Hotelling T^2 plots, next step is to identify what are the contribution levels of each events(attributes) to this abnormal phenomena. This can be done with the help of Q and Hotelling T^2 contribution plots.

Q contribution show how much each attribute contribute to overall Q statistic while retaining the sign. T^2 contribution explains the contribution of each attribute to the Hotelling T^2 value. These contributions are calculated for a given sample.

The T contribution for i^{th} sample, $t_{con,i}$, is calculated as given in equation 8 [8]:

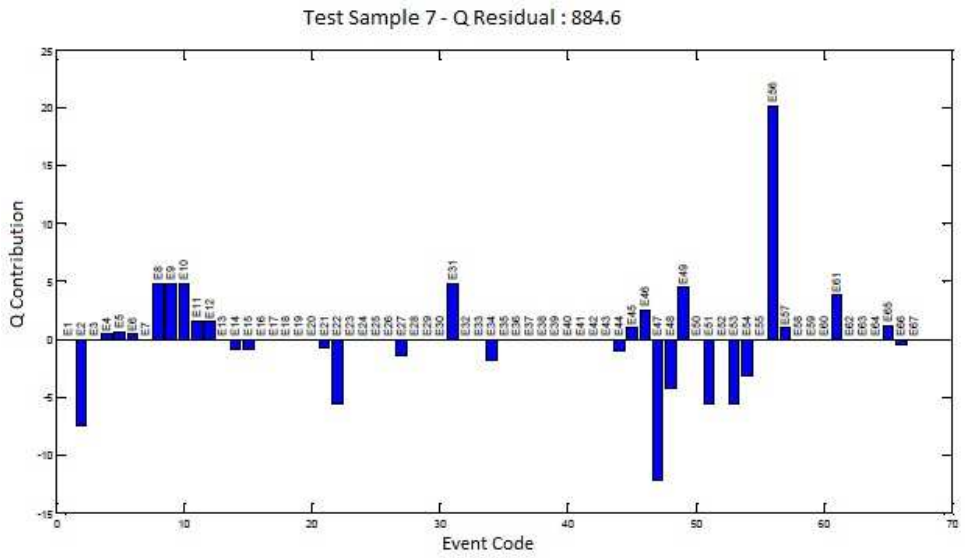
$$t_{con,i} = x_i P_k \lambda^{1/2} P_k^T \tag{8}$$

In the study that we conducted, the Q-Contribution plot is given in Fig 2-2a and T-Contribution plot Fig 2-2b for given sample which was outside the control limit(sample 7 of test data as in figures 2-1a and 2-1b). From this plot we can find that some of the events (eg. E56, E47) have higher influence on Q-Residual and Hotelling T^2. This analysis helps to build causal models which is beyond the scope this paper.

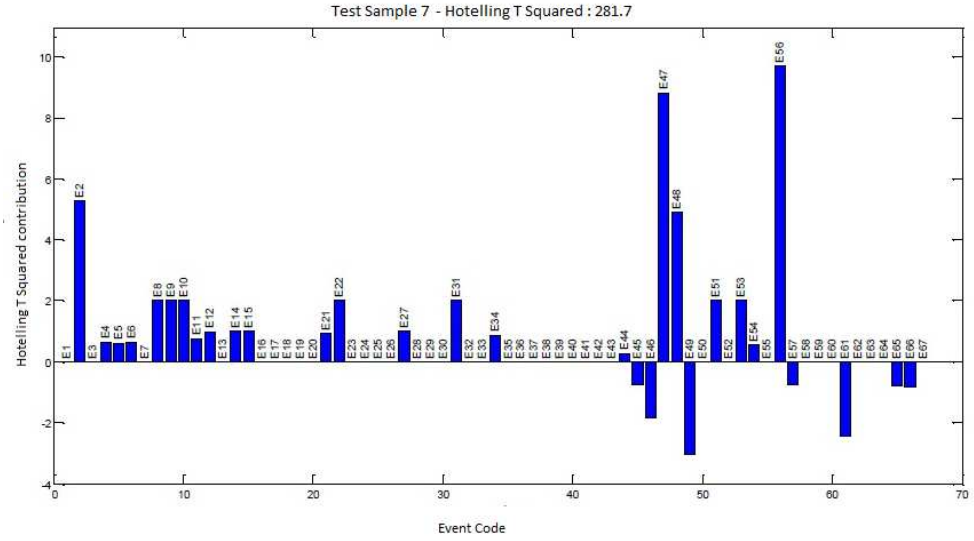
It is very important from service perspective to predict the failures of robots, especially when factories are located in remote locations. It may take days or weeks to reach the location and fix the failure. These predictions will also equip service engineers to arrange spare parts so that the delays can be avoided and thus improve customer experience. Predicting failures or repairs have significant benefits by offering service level agreements with high confidence level.

2.4 Conclusion

Our results show that using PCA for detecting the abnormal condition in event log pattern is a promising approach and can be used for giving sufficient indication to service engineers about the impending failure. Model is developed based on 5 PCs which explain 85% variability of the calibration data with 67 event types. The presence of fault or abnormal behavior can be identified with help of Q Residual or Hotelling T^2 statistics by comparing even log frequency pattern in time domain. The study results shows that it is possible identify the anomaly in event pattern around 30 days before failure.



(a) Q Contribution plot



(b) T Contribution plot

Figure 2-2: Q Contribution and T Contribution plot for identifying the contributing events for the anomaly/fault

2.5 Challenges and Future work

In this study, we attempted to predict failures with help of event log patterns using PCA. However, this study is limited to a specific type of failure- wear induced fault such as gearbox faults and generalization of this method to other types of failures need to be verified. The results show there is a possibility to predict failures with event pattern. This approach can

detect the changes in behavior of the equipment from the normal working condition. The event pattern may vary depending on the type of failure. For complex systems, there can be failure modes and types of failures. Our objective is to incorporate these aspects and enhance our approach for predicting failures. It is also observed that setting a threshold for number of type of event across different robot controllers for a specific failure will not be meaningful. This is because remote service data center will receive a bunch of event data from the controller at the time of alarm generation and configuration of the alarms for robot controllers is customer specific. As future work, we will be examining more field failures and corresponding event pattern. We are also exploring the possibility to simulate failures and events so that we can cover wide range of scenarios. In addition, we will evaluate the possibility to use t-SNE(t-distributed stochastic neighbor embedding), a non-linear alternative to PCA [23].

THIS PAGE INTENTIONALLY LEFT BLANK

3

Simulation based approach to detect wear in industrial robots[1]

Abstract: In this paper we simulate, detect, and locate wear in industrial robots operating over long time-periods on repetitive tasks. We simulate wear using the MATLAB robotics toolbox by increasing the Coulomb friction coefficient on select robot axes. Disturbance and measurement noise are included in each axis while computing the inverse and forward kinematics. To compare change across all configurations within a repetitive cycle, the output torque data is transformed into a normalized distribution sampled once every hour. We identify wear as significant change in torque values from normal operating conditions, and detect wear using principal component analysis; the source axis is located by cross-correlating the output data from all axes through time. Simulation results with a range of friction values tested for three different tasks across all axes show that wear detection on the basis of comparison with normal operating condition depends on the manipulator task, whereby onset of wear was detected in five out of six axes for one task and two out of six axes in the rest. Wear location was correctly identified in all successful detections except one.

[1]This chapter is a slightly modified version of paper published in IEEE International Conference on Automation Science and Engineering (CASE) 2015 and has been reproduced here with the permission of the copyright holder

3.1 Introduction

Friction has been reported as the primary cause behind more than 50% of the faults in heavy industrial robots [24]. Friction in robot joints can occur due to inadequate lubrication [25], high operating temperatures [2], accumulation of wear debris [26], and heavy load [2]. Unattended friction can lead to wear and tear, which eventually results in critical damage [2], [27]. Friction effects in robotic manipulators are also widely studied for their importance in determining the overall performance in terms of accuracy and stability [25], [28], [29].

In industrial robots, wear and tear is typically avoided as part of preventive maintenance schedules. These schedules are arrived on the basis of estimated life expectancy of various robot components—the actual condition of the robot is ignored [3], a scenario that leads to high maintenance costs. Condition-based maintenance is an alternative strategy that uses the data collected from various parts of the robots to assess the condition of the robot. Data from multiple robots is then subject to conservative thresholds on measurement data such as torque, temperature and voltage, which are then used to generate different types of events such as information events, warning events, and error events. While low thresholds may be used to avoid damage of expensive equipment, these are bound to raise the rate of false alarms from error or warning events leading to consumer complacency and missed detections [30].

Fault monitoring of industrial systems is performed using a wide variety of techniques that can be broadly classified into knowledge-based, model-based, and data-driven methods [30], [31]. Among these, knowledge-based methods rely on user expertise to arrive at customized thresholds for output signals that are input to a monitoring system [32]. These are specific to robot types and the operating conditions and are therefore non-transferable across different situations [30]. Model-based methods rely on friction models to predict the occurrence of a fault [3], [33]. In [33], for example, a friction model is used to estimate physical parameters of an industrial robot which are then utilized for fault diagnosis. While modeling approaches focus on an accurate representation of friction effects [28], [34], their direct application in fault detection and prediction requires the additional step of calibrating the robot to determine robot-specific model parameters, and may therefore be difficult to implement on

industrial systems that have a diverse line up [35]. Instead, data-driven approaches rely on statistical/signal processing or machine learning methods to characterize normal and faulty situations. While data-driven methods are designed for model independence, they require high-volume data inputs to perform reliably [30], [10], and rely on the variability in the training data for an exhaustive fault detection framework [36]. Following a recently proposed approach [37], we use an approximate model to generate high-volume synthetic data of wear due to low maintenance across multiple scenarios. The resulting data is used to develop and validate a wear detection and identification approach.

We model wear as a slow exponential growth in Coulomb friction spread over a period of five days to simulate instances of oversight in maintenance [26]. Realistic values of disturbance and measurement noise are introduced [38]. To allow detection and location of wear irrespective of the joint configuration, we transform the periodic torque values into a normalized distribution, and compute Kullback-Leibler (KL) divergence from the initial distribution as the new output data. The resulting output for all axes is then subject to principal component analysis (PCA) to compare with normal operating conditions. Wear is identified as an error in the data fit beyond a fixed confidence interval and located by cross-correlating the output data between axes. The approach is tested by simulating an increase in friction in all of six robot axes, one at a time, in three different tasks simulated using the open source MATLAB robotics toolbox [39]. The contributions of this paper are: (i) we add the capability of simulating wear due to friction in industrial robots in an open source software [39]; (ii) we implement PCA and cross-correlation based approach to detect the onset of wear in robots and identify the source location axis; (iii) we validate and analyse the proposed detection and identification approach on simulated data.

This paper is organized as follows: Section 2 gives a background of manipulator dynamics, PCA, and cross-correlation as a measure of detecting the delay between two signals. Section 3 describes the approach for simulating a gradual increase in friction in the presence of disturbance and noise, and its implementation on the MATLAB robotics toolbox [39]. Section 4 presents the wear detection and source identification approach. Section 5 provides results from data generated from an exhaustive set of experiments performed on three different tasks and finally, Section 6 summarizes the findings and discusses future work.

3.2 Background

3.2.1 Manipulator Dynamics

The simulation model used in this study is based on a six degrees-of-freedom (dof) manipulator called PUMA 560 (Programmable Universal Machine for Assembly). Given the joint angles $\boldsymbol{\phi} \in \mathbb{R}^6$, velocity $\dot{\boldsymbol{\phi}} \in \mathbb{R}^6$ and acceleration $\ddot{\boldsymbol{\phi}} \in \mathbb{R}^6$, the required torque $\boldsymbol{\tau} \in \mathbb{R}^6$ is given by the dynamic model

$$\boldsymbol{\tau} = M(\boldsymbol{\phi})\ddot{\boldsymbol{\phi}} + C(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) + F(\dot{\boldsymbol{\phi}}) + G(\boldsymbol{\phi}), \tag{3.1}$$

where M, C, F and G are inertia matrix, coriolis and centrifugal force matrix, friction matrix, and the gravity torque matrix respectively.

Equation (6.1) can be represented in a discrete state-space form for a single axis i with the state variable $\mathbf{x}^i = [\phi^i, \dot{\phi}^i, \ddot{\phi}^i]^{\mathrm{T}}, \in \mathbb{R}^3$ as

$$\begin{aligned} \mathbf{x}^i_{k+1} &= A^i \mathbf{x}^i_k + B^i \tau^i_k + \omega^i_k \\ \mathbf{y}^i_k &= C^i \mathbf{x}^i_k + \nu^i_k, \end{aligned} \tag{3.2}$$

where $\tau^i_k \in \mathbb{R}^1$ is the input torque at time step k, $\mathbf{y}^i = \phi^i \in \mathbb{R}^1$ is the measured output, $\omega^i_k \in \mathbb{R}^1$ is disturbance and $\nu^i_k \in \mathbb{R}^3$ is the measurement noise. A^i, B^i, and C^i are known system matrices [40], [41].

3.2.2 Principal Component Analysis and Cross-correlation

We use PCA and cross-correlation to detect and locate onset of wear.

Principal Component Analysis

PCA is a multivariate analysis technique [30] used to classify patterns in data [21]. Briefly, PCA finds principal components (PCs) that can best approximate the variation in original data [42]. We use PCA to detect a change in the overall torque values over extended periods of time. Given n torque values from all six robot axes, a torque matrix is constructed as

$\mathcal{T} \in \mathbb{R}^{n\times 6}$. PCA eigen-decomposes the matrix $\mathcal{T}$ into eigenvectors $P_m \in \mathbb{R}^{6\times m}$, and scores matrix $T_m \in \mathbb{R}^{n\times m}$, as [43]

$$\mathcal{T} = T_m P_m^{\intercal} + E, \tag{3.3}$$

where m is selected so that the principal components account for more than seventy five percent of the variation in data [44], and E is the residual error matrix.

The accuracy with which new data can be projected onto the principal components represents the degree of data fit and can be calculated using Q statistic [16]

$$Q = \mathcal{T}(I - P_m P_m^T)\mathcal{T}^{\intercal} \tag{3.4}$$

In our case, we use PCA to detect onset of wear by setting a threshold on Q-statistic. In other words, an onset of wear is identified if the Q value rises above a data-driven threshold [45]. We use 95% confidence level to set threshold on the Q-statistic [46].

Cross-correlation

Cross-correlation compares two signals at different time lags by correlating them. Given N pairs of observations of two time series τ^i and τ^j, the cross-covariance at lag δt is [47]

$$c(\delta t)_{\tau^i,\tau^j} = \begin{cases} \sum_{t=1}^{N-\delta t}(\tau_t^i - \bar{\tau}^i)(\tau_{t+\delta t}^j - \bar{\tau}^j)/N, & \delta t \geq 0 \\ \sum_{t=1-\delta t}^{N}(\tau_t^i - \bar{\tau}^i)(\tau_{t+\delta t}^j - \bar{\tau}^j)/N, & \delta t < 0 \end{cases} \tag{3.5}$$

and the cross-correlation coefficient for the time series τ^i, τ^j, is

$$R(\delta t)_{\tau^i,\tau^j} = \frac{c(\delta t)_{\tau^i,\tau^j}}{s_{\tau^i} s_{\tau^j}}, \tag{3.6}$$

where $\bar{\tau}^i$, $\bar{\tau}^j$ are the average of series τ^i and τ^j; s_{τ^i}, s_{τ^j} are the standard deviation of τ^i and τ^j, respectively.

The time delay δt at which the two signals give the maximum cross-correlation $R(\delta t)_{\tau^i,\tau^j}$ corresponds to the lag at which the signals are best aligned. In the case of two similar signals, δt at maximum R corresponds to the delay between them. A positive value of δt indicates that τ^j is trailing τ^i; conversely, a negative value indicates that τ^j is leading τ^i.

We use the delay in cross-correlation to determine the direction of fault propagation.

3.3 Simulating wear induced due to friction

To simulate wear, we increase Coulomb friction coefficient over an extended period of time (5 days) representing oversight in scheduled maintenance. The increase is set up so that the maximum torque stays within the recommended operational limits.

Denoting the Coulomb friction coefficient for positive and negative joint velocity as μ^+ and μ^-, the change in friction with time k is modeled as

$$\begin{aligned} \mu_k^+ &= r_0^+ + \frac{c_1}{1+\exp(-(c_2 k)^4)} \\ \mu_k^- &= r_0^- - \frac{c_1}{1+\exp(-(c_2 k)^4)}, \end{aligned} \tag{3.7}$$

where r_0^+ and r_0^- are set so that at $k = 0$, the initial values of Coulomb friction, μ_0^+ and μ_0^- are equal to the default values available from the PUMA560 Robotics toolbox [39], and c_1 and c_2 are constants computed by solving (3.7) for increase in friction at the end of the five days. For each k, the updated value of Coulomb friction coefficient is substituted in equation (3.1) to arrive at Friction matrix F [39]

$$F(\dot{\phi}) = V\dot{\phi} + \mu, \tag{3.8}$$

where V denotes the coefficient of viscous friction. Figure 3-1 shows a sample change in friction torque as a result of change in Coulomb friction using (6.6) (Note that disturbance and noise are added in the same sample as per Section III B). Operational torque limit for the Puma 560 robot model is given in Table 3.1 [48].

Table 3.1: Operational torque limit for the Puma 560 robot model

Axis	**1**	**2**	**3**	**4**	**5**	**6**
τ_{max}**(Nm)**	120	200	100	22	22	21

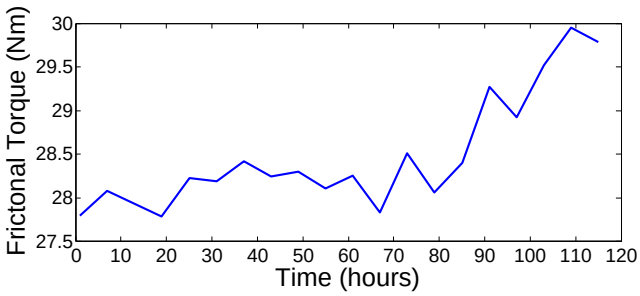


Figure 3-1: Change of frictional torque due to increase in Coulomb friction.

3.3.1 Implementation on the robotics toolbox

The MATLAB robotics toolbox is a set of open source MATLAB functions that allow ease of simulation and analysis of robotic manipulators [39]. The toolbox consists of user-defined functions that can help visualize and create a robot trajectory, compute input torque using inverse dynamics (1), and compute output torque using forward dynamics [39]. We use the robotic toolbox to simulate increase in Coulomb friction along with disturbance and measurement noise in a PUMA 560 manipulator.

3.3.2 Adding disturbance and measurement noise

In order to represent a realistic scenario where disturbance and measurement can cause high false alarm rates, we add disturbance and measurement noise to the simulations of manipulator tasks. In particular, unknown disturbances may occur during the operation of a robotic manipulator due to minor variations in load torques and change in lubricant temperatures [36]. Measurement noise may occur due to sensor degradation and high or low operating temperatures.

Disturbance ω_k in (4.1) is added in the form of a zero-mean Gaussian process with unit standard deviation scaled to one percent of the average torque [49]. Using (4.1) as reference, here, we assume $C \in \mathbb{R}^{3\times3}$ to be an identity matrix. Similarly as in [49], the measurement noise ν_k is also modeled as a zero mean Gaussian process. To obtain the noise levels in $\dot{\phi}$ and $\ddot{\phi}$, ν_k is then scaled with the inverse of the sampling time-step δt and δt^2 respectively. The modified measurement equation is

$$\mathbf{y}_k^i = C\mathbf{x}_k^i + \begin{bmatrix} 1 \\ \frac{1}{\delta t} \\ \frac{1}{\delta t^2} \end{bmatrix} \nu_k, \tag{3.9}$$

where $C \in \mathbb{R}^{3\times3}$ is an identity matrix, ν_k is 0.0001 rad, based on the average accuracy of the encoder in robotic manipulators [50] and $\delta t = 0.5$ s. Once $\mathbf{y}$ is obtained, we use inverse dynamics (6.1) to obtain the measured torque.

Table 3.2: Algorithm to simulate wear due to friction

Input: Start and end position of trajectory, duration of operation, time step between successive operation, and torque operational limits

For each time step k

1. Generate joint position,velocity, and acceleration, $\phi, \dot{\phi}, \ddot{\phi}$ using forward kinematic equation (*jtraj*)
2. Calculate torque for each time step using inverse dynamic equation (*rne*). Introduce friction increase by modifying Coulomb friction coefficient (6.6) along with disturbance as in (4.1) modeled as zero mean Gaussian process
3. Compute $\phi, \dot{\phi}, \ddot{\phi}$ for each axis.
4. Add measurement noise as per equation (3.9) to $\phi, \dot{\phi}, \ddot{\phi}$
5. Calculate measured torque using inverse dynamic equation (*rne*)

Table 3.2 lists the schematic of the algorithm for simulating wear in robots using the MATLAB robotics toolbox [39].

3.4 Detecting and locating wear

In this section, we describe the wear detection and location method. The torque in an axis is dependent on configuration and may vary drastically depending on the task to be performed. Sampling the torque at fixed time intervals of time does not guarantee measuring it at the same configuration due to changes in acceleration and velocity of robot joints. Further, sampling the torque at a fixed configuration may also lead to completely ignoring

a problematic configuration of joints and extent of wear depends on velocity, acceleration, and direction of movement of the axes. To make the comparison of torque independent of configuration, we implement the proposed wear detection approach on the KL distance [36] of the normalized torque distribution. The KL distance measures the similarity between two distributions [1, 51] The torque is sampled in every ten seconds for first ten minutes of every hour. Thus, in an hour, sixty samples are collected. This ensures that we obtain the torque values for six possible configurations in a trajectory that repeats itself once every minute. These sixty values are then normalized with respect to maximum torque and binned into twelve equal-sized intervals. We generated such torque histograms to simulate our existing data collection from robots at customer site and comparing histograms help to evaluate full trajectory within a cycle instead of a single configuration.

3.4.1 Identification of onset of wear

We use PCA based data-driven approach that does not depend on predefined thresholds to detect changes in axis torque due to increase in friction. In particular, we use the torque distribution in normal operation as training data to generate principal components using equation 3.3. Torque from all six axes are used together to create a single normal-working profile. Fig. 3-2 shows the KL distance over time across all axes for trajectory 1 when we introduce 10% wear in Axis 1. The value of KL distance was increased significantly after seventieth hour for axis 1 and KL distance for axis 6 appears to be very sensitive. However, such sensitiveness of KL distance for axis 6 is not observed consistently across all trajectories and the variation in KL distance depends on the robot trajectory and the load. In further analysis (not shown here), for 20% wear in axis 1, KL distance is significantly higher for axis 1 compared with all other axes. The number of PCs are selected as described in Section 3.2.2 and ranges between 3-4 in our case. Figure 3-3 shows a sample training and testing procedure where one-hundred and twenty hours of normal working operation is used to obtain the output data for training. Friction is introduced at the 120^{th} hour. Therefore, Q-residual shows an increase in value from 120^{th} hour and significant increase after 180^{th} hour. Anomalies are detected by checking if the data falls within the thresholds defined using the Q-statistic (3.4).

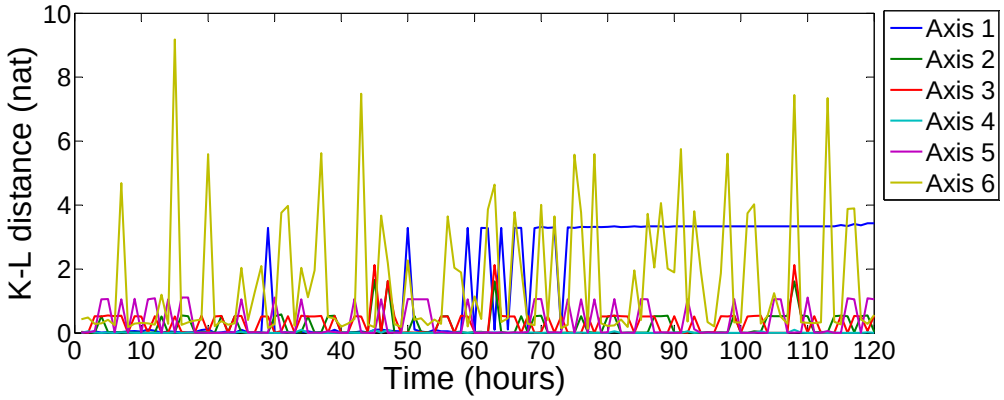


Figure 3-2: KL Distance Vs Time. Friction increase of 10% is introduced in axis 1.

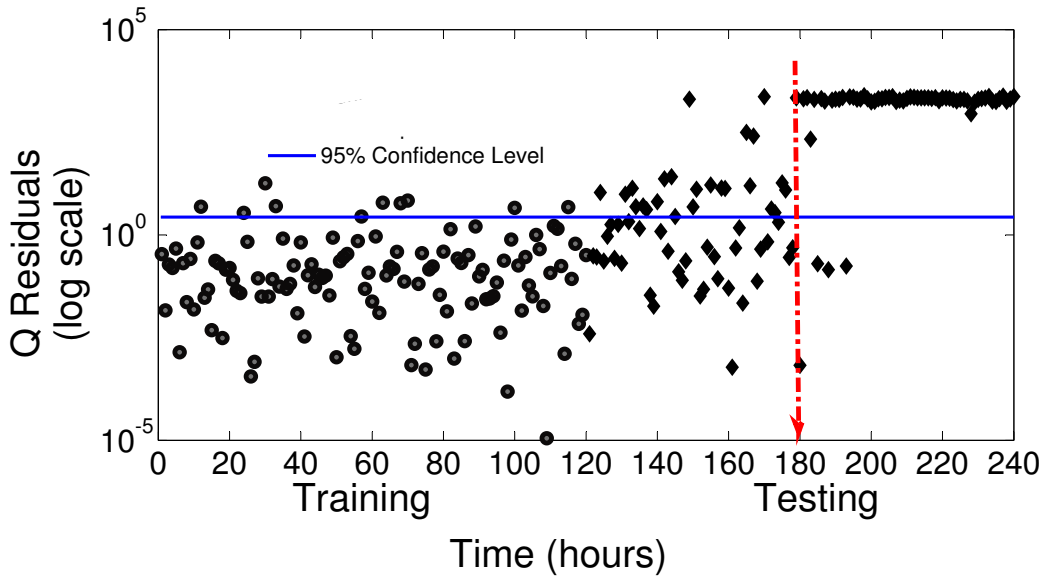


Figure 3-3: PCA analysis based on KL distance. Circles represents training or calibration data and diamond represents test data

3.4.2 Identification of location of wear

We have used the classical time lag analysis to infer whether a process is leading or following the other. We cross-correlate the output data (torque distribution for each axis across time) to locate the source axis of the wear. Specifically, we use the sign of lag at maximum cross-correlation between torque from two axes as an indicator of whether the process is leading or trailing.

Table 3.3: Lag (hours) at maximum correlation between axes when friction is introduced in axis 1. Leading signal will register negative time lag. Axis with wear is identified as one that has all negative lags.

Source Axis	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6
Axis 1	0	-3	-3	-27	0	-4
Axis 2	3	0	0	0	-38	-2
Axis 3	3	0	0	0	6	0
Axis 4	27	0	0	0	-22	5
Axis 5	0	38	-6	22	0	32
Axis 6	4	2	0	-5	-32	0

In this case, we expect that a leading signal will register a negative time lag. Table 3.3 shows the lag (in hours) at maximum cross-correlation for all pairs of axes where the friction was increased in axis 1. We identify the source axis as one that has all negative lags. Fig. 3-2 shows the KL distance of each axis from a reference distribution.

3.5 Experiments and analysis

The details of simulation experiments is given in Table 3.4. We introduce increase in friction corresponding to 10% and 20% change over five days (this resulted in the overall torque not exceeding more than 50% of the maximum allowed design torque for any axis). We simulate the friction change in all six axes, one axis at a time.

Table 3.4: Experimental design

Trajectory	% Friction Increase	No. of simulation runs
1	10	6
1	20	6
2	10	6
2	20	6
3	10	6
3	20	6

We designed three different tasks to exhaustively validate the detection and location approach described in this paper. The tasks represent pick-and-place operations and involve the robot end-effector reaching its workspace limits and rotation of all joints (Fig. 3-4).

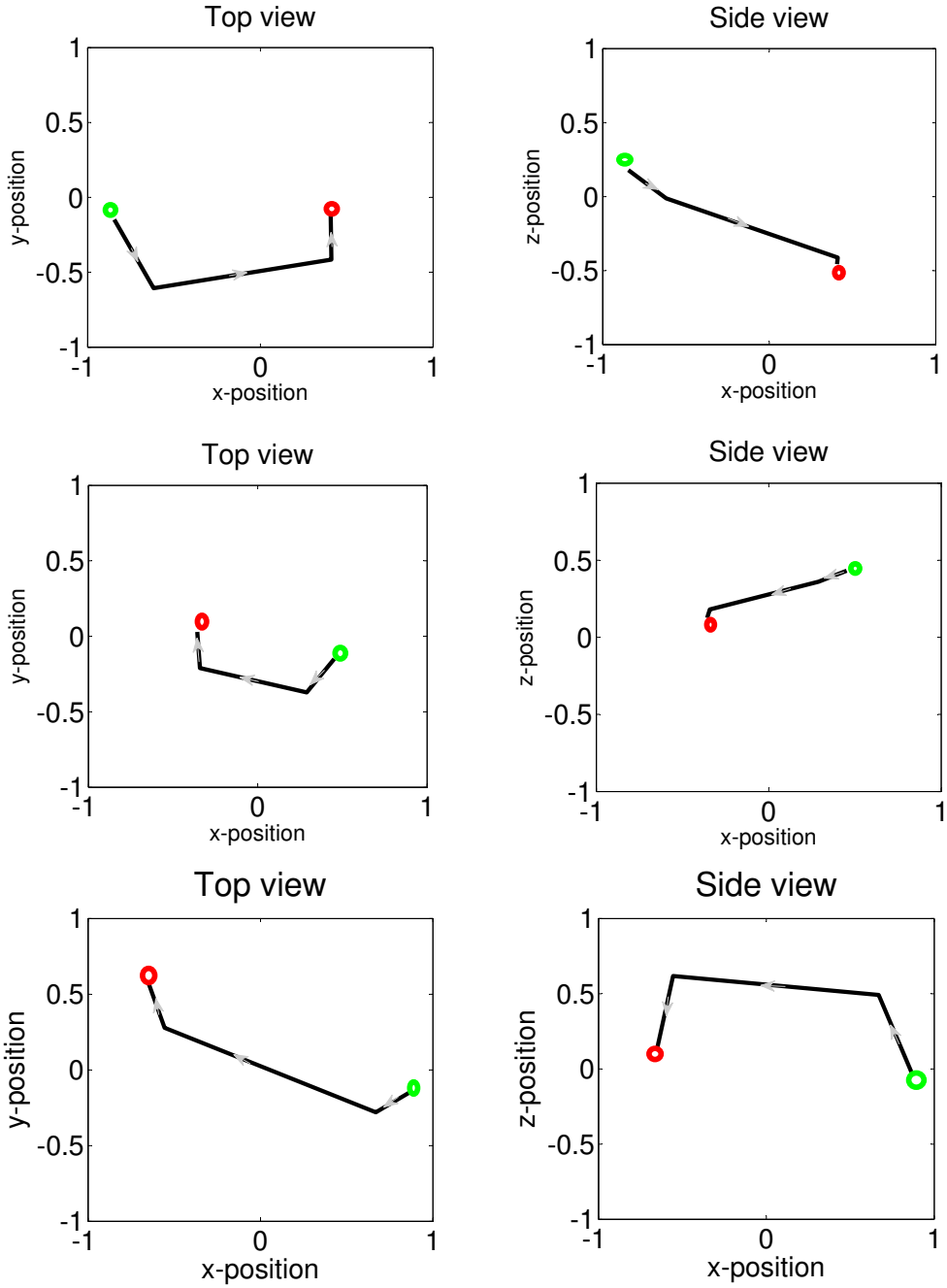


Figure 3-4: End-effector trajectories for three different manipulator tasks. Top and side view of a single run of the trajectories are shown for each task. The tasks represent simple pick-and-place operations. Green circle denotes start and red circle denotes the stop position

Duration of one cycle for each task is set to one minute repeated over five days (120 hours). The top and side view of the end-effector for each of the three trajectories are shown in Figure 3-4.

Table 3.5 shows the detection and source location performance for all thirty-six simulations. Onset of wear on the basis of a data-driven threshold using PCA is both detected and located in all axes except axis 2 for task 2, two axes for task 1, one axis for task 3 with 10 % friction increase, and four axes in task 3 with 20% friction increase. While we increased the friction coefficient by the same amount in all axes, the maximum absolute torque at the time of detection or at the end of simulation period (Table 3.6) shows that the increase does not have the same effect on all axes. Between all tasks, task 2 has the lowest loads for axes two and three of the manipulator, and has the highest load for the sixth axis. At the same time, task 2 has a low coefficient of variation (Table 3.7) for most of the axes between all tasks. This shows that the detection of wear is dependent on the torque load and variability on an axis. Between axes, 4-6 are closer to end-effector and their movement and rotation depends on the task at hand and it is likely that the disturbance and noise, also introduced into the training data, do not allow the change due to increase in friction to manifest.

Table 3.5: Wear detection and location using PCA and cross correlation- with the number of hours at the time of detection

	Friction Increase 10%			Friction Increase 20%		
Axis	T1	T2	T3	T1	T2	T3
1	✓(60)	✓ (60)	×	✓(50)	✓(50)	✓(40)
2	×	×	×	×	×	×
3	✓(100)	✓(90)	×	✓(85)	✓(78)	✓(105)
4	×	✓ (80)	×	×	✓(60)	✓(90)
5	×	✓ (115)	×	×	✓ (95)	×
6	×	✓ (90)	✓ (95)	×	✓ (90)	✓ (90)

Task 2 trajectory is different from that of tasks 1 and 3 in terms of the region of workspace it explores. The low coefficient of variations could possibly result from a simpler path that has less workspace region to cover compared to task 1 and task 3. Both tasks 2 and 3 involve the rotation of sixth axis, an effect of which is indicated in the similarity between the absolute torque values for axis 6 for these two tasks as compared to task 1.

The detection and location of the onset of wear is the same for tasks 1 and 2 for both ten and twenty percent increase in friction with the detection taking place earlier for twenty

Table 3.6: Maximum absolute torque (Nm) at the time of detection or at the end of five days if wear was not detected

	Friction Increase: 10%			Friction Increase: 20%		
Axis	T1	T2	T3	T1	T2	T3
1	28.520	29.705	29.070	29.724	29.475	28.799
2	81.842	64.176	71.769	82.533	64.665	71.735
3	26.240	9.248	25.620	26.421	9.484	25.442
4	2.292	1.381	2.226	2.292	1.409	2.249
5	2.434	3.183	2.822	2.446	3.231	2.803
6	0.000	0.912	0.834	0.000	0.928	0.873

percent than ten percent. For task 3, the wear is detected in three additional axes when the friction increase is 20 percent. This is expected since the anomalous change in torque distribution will be quicker with a larger increase in friction over the same time period. The time of detection itself indicates that disturbance and noise can overshadow an increase in friction for over three days before any change in torque can be detected.

Table 3.7: Coefficient of variation of absolute torque (Nm) at the time of detection or at the end of five days

	Friction Increase 10%			Friction Increase 20%		
Axis	T1	T2	T3	T1	T2	T3
1	0.733	0.049	0.047	0.733	0.048	0.051
2	0.210	0.227	0.230	0.225	0.233	0.233
3	0.766	0.254	0.350	0.773	0.242	0.351
4	0.625	0.192	0.824	0.625	0.189	0.798
5	0.758	0.614	0.675	0.757	0.607	0.677
6	0.731	0.409	0.445	0.692	0.405	0.422

Studies on fault detection and prediction using simulation, focus on introducing faults artificially by incorporating sudden load changes in operation [37], [52]. In [37], two types of faults: actuator fault and sensor fault, are introduced and detected immediately using statistical differences. The source of the actuator fault is identified by comparing regression coefficients between normal and faulty condition. In [52], actuator faults in two-link manipulators are detected with negligible delay in all of the nine test cases. In contrast, the

wear simulated and detected in this paper does not create sudden changes in manipulator performance but indicative of an alarm event. Accordingly, although the onset of wear is detected hours after it is introduced, the actual fault may appear still later. Finally, using our approach, the source axis was correctly identified in all cases where the onset of wear was detected.

To ensure that the algorithm can be implemented in practice, we ensured that the measurement data matches the format of the output from the remote servicing agent and the normal operating condition is characterized on the basis of sixty days of normal operation, a time period for which we expect data to be easily available. The computation of secondary measures such as KL divergence and cross-correlation lags is computationally simple and can be implemented on the remote servicing agent. Finally, open issues include identifying optimal thresholds that likely depend on robot and environment specific disturbance and measurement noise values for usable fault time prediction.

3.6 Conclusion

In this paper, we present an approach to detect and locate the onset of wear in industrial robots operating over long time periods. Wear is simulated on a PUMA 560 robot using MATLAB Robotics toolbox by increasing Coulomb friction in the presence of process and measurement noise. PCA and cross-correlation is used to create data-driven thresholds for detecting wear, and locating the source axis. Multiple simulation experiments are conducted on output data represented in the form of torque distribution data and compared during periodic cycles.

Our results show that the detection and location of wear depends on the manipulator task. Low variability in load and movement of axis that may result from different trajectories affect the change in torque as a result of adding friction. Further, given disturbance and noise in the system, the time of detection shows that the onset of wear may take more than three days to manifest.

Future studies will focus on two directions: first, on validation of this approach on real data collected from the field, and second on developing methods to characterize and identify the cause behind an increase in friction that may result from multiple sources [25], [2], [26].

THIS PAGE INTENTIONALLY LEFT BLANK

4

Training Data Selection Criteria for Detecting Failures in Industrial Robots[1]

Abstract: We study the effect of source and type of training data on detecting failures in industrial robots using Principal Component Analysis (PCA). Specifically, using field data across multiple robots performing different tasks, we compare two scenarios: first, where training data obtained from a single robot is used to evaluate multiple robots (one-to-many), and second, where each robot is evaluated on the basis of its own training data (one-to-one). We further investigate if the data preprocessing prior to running PCA affects the ability to detect and predict failures. To reduce task dependence of the raw signal, we preprocess the same by computing the absolute difference between successive measurements and compare the results with a PCA model that is built using raw signal alone and another that is built from a combined signal having both raw measurements and their absolute difference. We quantify effectiveness of detecting failures in terms of three measures: coefficient of variation of the Q-residual obtained by projecting the test data on the PCA model, number of samples above a data-driven confidence threshold, and lead time, measured as the number of days prior to failure when the residual error rises above a given threshold. Our results show that

[1]This chapter is a slightly modified version of paper published in IFAC-PapersOnLine 49, no. 1 (2016): 385-390 and has been reproduced here with the permission of the copyright holder

coefficient of variation of the Q-residual from a PCA model built using absolute difference between measurements serves as a robust descriptor for predicting and detecting failure in robots in the one-to-many training scenario. Specifically, we show that while both one-to-one and one-to-many training sources are valid for detecting failures, signal preprocessing has a significant influence. With the same signal, when using number of samples above threshold, we find that one-to-one training source is able to detect failure in robots. Finally, with lead time, we find that one-to-one training scenario with absolute difference as signal type can be used to raise warning as early as nineteen days before failure.

4.1 Introduction

In a remote service set-up for industrial robots, information such as events and measurements are sent to a remote database for hundreds of robots ([4]). This information is further processed to prepare advisory reports that assist field engineers. The value of a remote service support can be enhanced by improving the capability of predicting and detecting failures across multiple robots.

Failure detection can be improved by modeling the robot operation subject to different loads ([2]). However, given the variability in the type of tasks and loads borne by industrial robots, and the difficulty in accurately modeling a failure within a robot operation, data-driven approaches provide an alternative black-box type approach to detect failures ([53, 54]). Data-driven methods do not require prior knowledge about the process and rely entirely on collected data ([17]. Such algorithms learn the nominal behavior of the system from the actual data itself and are validated on the basis of accurate classification of new unseen data called the test data. Anomalous behavior is then detected in the form of error in a model fit ([13, 36, 55, 56]).

Industrial robots are typically instrumented with many sensors, monitoring and reporting the health of the robot in terms of multiple measurements including torque loads, temperature, arm speeds, and position. With more than a hundred such measurements reported on an hourly basis, it is not immediately clear which measurement is important for which type of failure.

Principal component analysis (PCA) is a statistical learning approach that helps to reduce the dimensionality of the input dataset under the assumption that the data points are linearly separated. It has been used in the past effectively for dimensionality reduction ([17, 30, 57]) and fault detection in equipment and processes ([14, 42, 44, 58–60]). However, PCA effectiveness is evaluated on the basis of model fit which traditionally entails using data from the same source that has to be tested ([14, 57]).

Here, we first investigate if failure detection across multiple robots is significantly affected by the source of training data, that is, a single reference robot for all tests (one-to-many) or nominal data from the same robot (one-to-one) for all tests (Hypothesis 1). Instead of using measurement data as is, which is likely a function of the specific task performed by the robot, we further investigate if signal preprocessing can make detection of failure task invariant. For example, assuming that a failure is accompanied by large changes in measurement values, measurement change over time may be used to represent normal operation across multiple robots. Accordingly, we test if the effectiveness of the data-driven approach depends on three types of signal input used to build the PCA model: raw signal, absolute difference between two successive raw signal values, and a combination of both (Hypothesis 2).

We test hypotheses 1 and 2 on the basis of the descriptors on the residual error obtained after projecting the test data on the PCA model. In particular, we use multiple descriptors such as coefficient of variation of Q-residuals, number of test samples above Q-residual threshold limit, and lead time (in days), defined as the remaining time in days left in the test data set when the first sample above threshold is detected, to quantify the effectiveness of the failure detection and prediction approach using PCA. The contributions of this paper are: (i) we collect data from industrial robots for analysis of fault-detection strategies, (ii) we describe and test alternative data selection criteria to reduce collection time and task dependence, and (iii) we perform experimental analyses to demonstrate the effectiveness of signal processing on a PCA based approach for failure detection.

This paper is organized as follows: Section 2 gives a background on different failures in industrial robots and training strategies for PCA model development. Data collection, preprocessing and experimental analysis are described in Section 3. Section 4 provides the results followed by discussion, conclusion and a summary of ongoing work in Section 5.

4.2 Background

4.2.1 Failures in Industrial robots

Failures in industrial robots can be classified as mechanical based, sensor related, and actuator system related ([61]). Different reasons have been attributed to each of these failures. For example, gearbox and brake failures occur mainly due to wear ([1]), and motor failure can be due to a short circuit ([61]). Because the pattern of change of data and the time of change can vary widely across all failures ([1]), combining all the failures into a single failure-detection framework is difficult when a single model is used. Instead, an approach that uses all reported measurements and events may be able to capture robot operation for failure detection across multiple tasks and operating environment.

4.2.2 Training data selection and preprocessing for PCA

Principal Component Analysis is a multivariate analysis technique ([30]) that helps to reduce the number of features to a small number of principal components without losing important information ([13], [21]). PCA has been widely used in other fields such as face recognition ([62]), fault detection in semiconductor processes ([60]), and anomaly detection in by monitoring sensor data in air plane systems ([63]).

Given n instances of m different measurements, the PCA algorithm eigen-decomposes the input data $\mathcal{X} \in \mathbb{R}^{n \times m}$ into the principal components, where the number of principal components are selected such that they account for maximum variation in data ([44]). The accuracy with which data can be projected onto the principal components represents the degree of model fit and can be represented using Q-residual ([16]). Q-residual is calculated as the difference between a sample and its projection into the principal components retained in the model ([60]). We use a 95% confidence level to set the threshold on the Q-residual beyond which a model is declared not fit ([46]).

The effectiveness of PCA to classify new data accurately depends on the size and extent of the training data ([62]). At the same time, a generic PCA model that is built using a large volume of training data may give rise to higher incidence of false negatives in a failure-

detection framework ([64]). PCA model is traditionally developed with training data from the same system where the test is to be performed ([14, 17]). To overcome variability and underfitting due to new data, Li et. al. ([64]) have used a recursive PCA algorithm that is periodically revised to recalculate the thresholds for the Q-residuals. This approach helps to reduce false alarms because of slow and normal changes in the industrial processes over period such as equipment aging and sensor drifting.

4.3 Data collection, pre-processing and experimental analysis

4.3.1 Data collection

Time-series data collected from industrial robots can be classified into two categories: measurements, which in turn can be mechanical (e.g. speed, torque) and non mechanical (e.g. CPU temperature, fan speed), and events, which are generated using rule-based criteria on measurements, reported or otherwise. An example of an event is the over-temperature event, generated because temperature of the main computer unit is too high. We obtained data from seven different instances of four types of failures and three instances of normal operation from ten different robots. The seven failure instances consist of two gearbox failures, two motor failures, two brake failures and one computer unit failure. All robots have six axes performing different tasks within the same application domain. For each robot, measurements and event frequency was extracted by querying the remote servicing database using a custom SQL script. Fault types, wherever a failure instance was found, was reported in the comment log of the service engineer. Robot type and application were confirmed using remote service data[2].

[2]For confidentiality reasons ABB specific information such as robot type, location, and measurements are not disclosed.

4.3.2 Preprocessing of data

Data from the three normally operating robots was further split so that six instances of normal operation were available for testing. Specifically, two datasets of 20 days each were extracted from each normal robot so that they were furthest apart in time. Each training and test dataset comprised 60 days of operation time that was further split into first 40 days of training data followed by 20 days of test data. The last day of the 60-day dataset (20th day in the test data) was the day of failure. To avoid including measurements and events very close to failure in time, the last five days of test data (including the day of failure if it occurred) was removed from the test data. This approach also ensured predictive capability of a model for up to at least five days (one work week) before failure. While there was some variability in the number of measurements and events reported by each robot, a total of 104 measurements and 13 events were found to be common across all robots and were therefore used for training and testing. The events were reported in the form of number of observations per day.

The measurements and events were further preprocessed to provide three different types of inputs as training data for PCA: i) raw signal, ii) absolute difference between two successive raw signal values and iii) combination of both (i) and (ii). Some axis-specific measurements such as position and torque were available in the form of a frequency distribution. To compare such values we used a probabilistic measure called Kullback Leibler (KL) distance. While preserving the difference in such measurements, this approach also reduced the total number of measurements. Given two probability distributions of measurement X, $p(x)$ and $q(x)$, KL distance $D(p||q)$, is defined as ([65],[51])

$$D(p||q) = \sum_{x \in \chi} p(x) \times \log \frac{p(x)}{q(x)} \tag{4.1}$$

In the case of raw signal, KL distance was computed between the current measurement and the first measurement. As an example, for a given data set with twenty days of data, first measurement indicates the measurement that is farthest away from the failure date for failed cases and farthest away from the last measurement date for normal robots (Fig. 4-1). For absolute difference, KL distance was computed between successive measurements.

Linear trends in measurements were removed prior to training by detrending the signal. The number of final measurements and events for each data set were 117 for raw and absolute difference signal, and 234 for the combined signal.

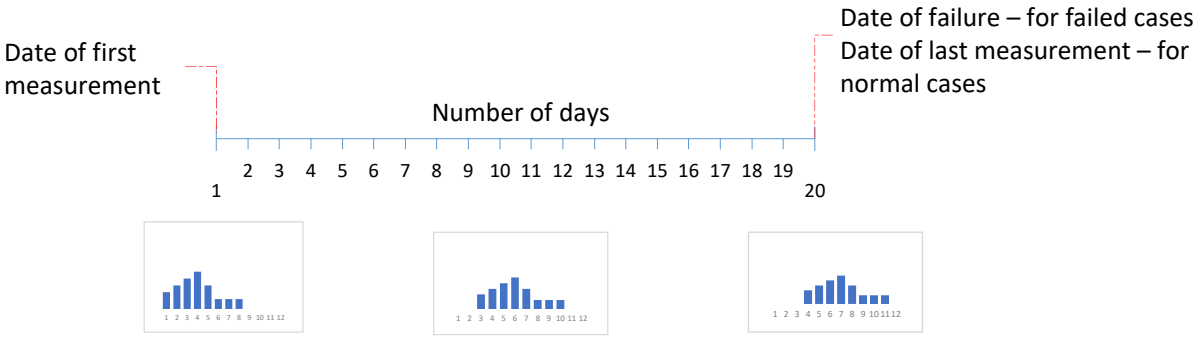


Figure 4-1: Concept of first measurement and last measurement for a given data set.

4.3.3 Test setup

A three-way analysis is designed to test the effect of training source and training signal type and robot condition on the ability to isolate failures using the defined descriptors. However, we have not considered the effect of failure type in the study due to few failure cases in each failure type. The training source was set as one-to-many (OTM), or one-to-one (OTO) (Figure 4-2); the training signal type was set to raw, absolute difference, and combined; and the robot condition was selected as failure or normal. To avoid bias due to training data from particular robot in the OTM scenario, PCA models were trained using 7 failure and 6 normal robots to create 42 test cases. Table 4.1 provides a list of test scenarios. The number of Principal Components were chosen in such a way that selected components explains 90% variability in the training data (Fig. 4-3).

In addition to testing hypotheses 1 and 2 on the 15-day test data, we varied the number of test samples to 10, 12 and, 13 days to identify a higher lead time for predicting failure.

We compared different scenarios using the following descriptors based on Q-residual: coefficient of variation ([46]), defined as the ratio of standard deviation to the mean; number of test samples above a threshold limit; and lead time defined as the number of days prior to failure when the model fit error first rises above threshold. Note that since we ignore the

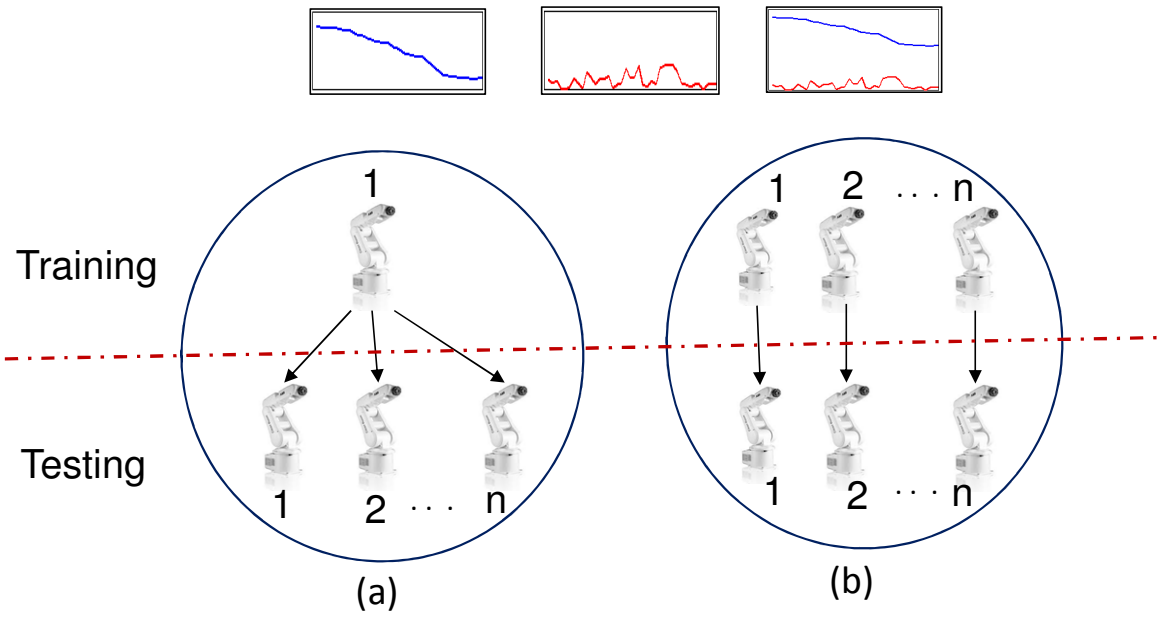


Figure 4-2: Test Conditions. (a) One-to-many scenario: PCA model built using training data from a single robot. (b) One-to-one scenario: PCA model is built for each robot using its own training data, shown as dashed line. Robot image is obtained from ABB product catalogue.

Table 4.1: Experimental Design

Training Source	Signal Type	Robot Condition	Number of test Cases
OTM	Raw	Failure	42
OTM	Raw	Normal	36
OTM	Difference	Failure	42
OTM	Difference	Normal	36
OTM	Combined	Failure	42
OTM	Combined	Normal	36
OTO	Raw	Failure	7
OTO	Raw	Normal	6
OTO	Difference	Failure	7
OTO	Difference	Normal	6
OTO	Combined	Failure	7
OTO	Combined	Normal	6

last five days of the test data the actual lead time adds five days to the value obtained.

We used analysis of variance (ANOVA) to compare the different scenarios. Given groups of experiments for each independent variable, ANOVA measures the source of variation in the data and compares the relative sizes ([66]). The sources of variation can be within-group or between-groups. The ANOVA comparison reports the probability of rejection of the null hypothesis (p-value), namely that the groups are not significantly different, and the F-statistic which is the ratio of variation between the group to within the group, we evaluate the significance. A large F value indicates that there is more difference between groups compared to within groups ([66]). If the p-value is less than 0.05, then the group means are considered to be significantly different.

We assume an interaction model for the following reasons: (i) signal type and training source will likely be impacted by the condition of the robot that is tested; (ii) it is also likely that the impact of the signal type (raw or difference) will be affected by the training source (OTO or OTM); and (iii) the absolute difference signal captures variation in data and is therefore more representative of the condition of the robot (normal or failure), where as raw signal is more representative of the task.

Significance level is set to $p < 0.05$. If an interaction effect is found, post-hoc comparisons are made using Tukey's honest significant difference (hsd) test. All statistics were performed using MATLAB (R2014a, Mathworks, Natick, MA). Generalized Linear Models (GLMs) are as secondary analysis using "Stat" package in R language [67].

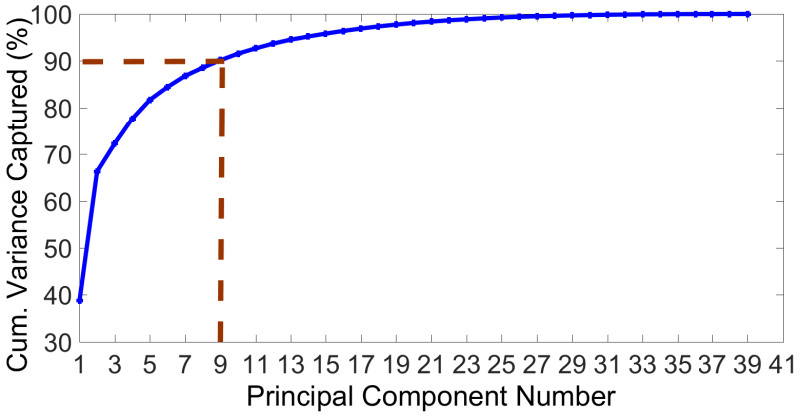


Figure 4-3: Selection of Principal Components: number of PCs selected which explains 90% variability in the data .

4.4 Results

ANOVA comparison of coefficient of variation (COV) of Q-residual across different scenarios shows a combined interaction effect between signal type and robot condition. Specifically, results show that while there was no interaction effect found between training source × signal type × robot condition ($p = 0.789$), training source × robot condition was significant ($p = 0.033$); there was no interaction between training source × signal type ($p = 0.661$) and signal type × robot condition ($p = 0.123$). Averaging across all other variables (main effects), the coefficient of variation from both signal type and robot condition was found to be significantly different (p<0.01), but not for training source (p=0.823). Post-hoc comparisons between different robot conditions across signal types show that the normal and failure robots were significantly different when absolute difference was used as the signal type and training source as OTM (Figure 4-4). With fewer days of test data, the significant difference between normal and failure robots was lost for 12 days and less.

ANOVA comparison of the number of test samples above threshold limit shows that the interaction effects as well as main effects are significant ($p < 0.01$). Post-hoc analysis for all pairs shows that number of samples above threshold is significantly more for failure robots than normal robots when absolute difference is used as signal type with OTO as the training source (Figure 4-5). This result of significant difference between normal and failure robot continued to hold even when the test-data was reduced to 10 days.

ANOVA results on lead time in days shows that the interaction effects are significant for all combinations and main effects ($p < 0.01$). Post-hoc comparisons across all pairs of conditions show that when signal type is absolute difference, the lead time is significantly more for a failure robot with OTO training source (Figure 4-6). This result of significant difference between normal and failure robot continued to hold even when the test-data was reduced to 10 days.

COV of Q residual includes the changes in Q- residual that are below threshold. However, other two descriptors (lead time and number of samples above threshold) are based on the samples above threshold; therefore, these descriptors are not capturing the changes in Q residual observed below the threshold value. As a result, COV of Q residual shows a clear

difference between evaluated factors compared to other two descriptors. ANOVA results are provided in Tables 4.2, 4.4, and 4.3.

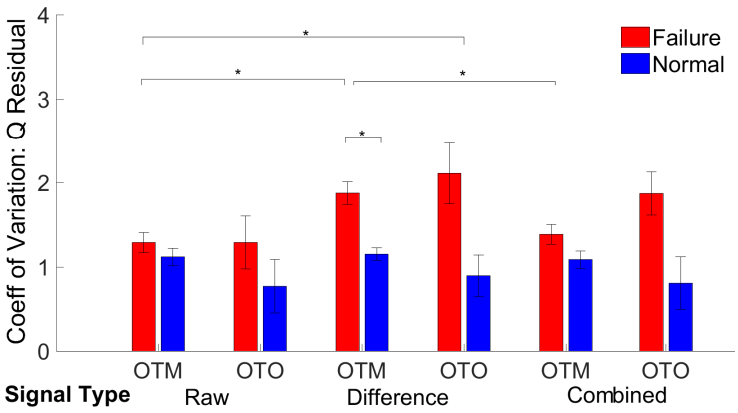


Figure 4-4: Coefficient of variation of Q-residual for each robot condition. * indicates significantly different conditions in post-hoc comparisons. Error bars denote standard error of the mean.

Table 4.2: Analysis of Variance – Coefficient of Variation (CoV) of Q Residual

Source	DF	Adj SS	Adj MS	F-Value	P-Value
TrainingSource	1	0.02	0.0197	0.02	0.88
SignalType	2	7.376	3.6878	4.32	0.014 *
RobotCondition	1	20.306	20.3064	23.78	0 ***
TrainingSource*SignalType	2	1.056	0.5282	0.62	0.539
TrainingSource*RobotCondition	1	3.073	3.0727	3.6	0.059
SignalType*RobotCondition	2	5.817	2.9086	3.41	0.035 *
TrainingSource*SignalType*RobotCondition	2	0.965	0.4823	0.56	0.569
Error	261	222.833	0.8538		
Total	272	278.307			

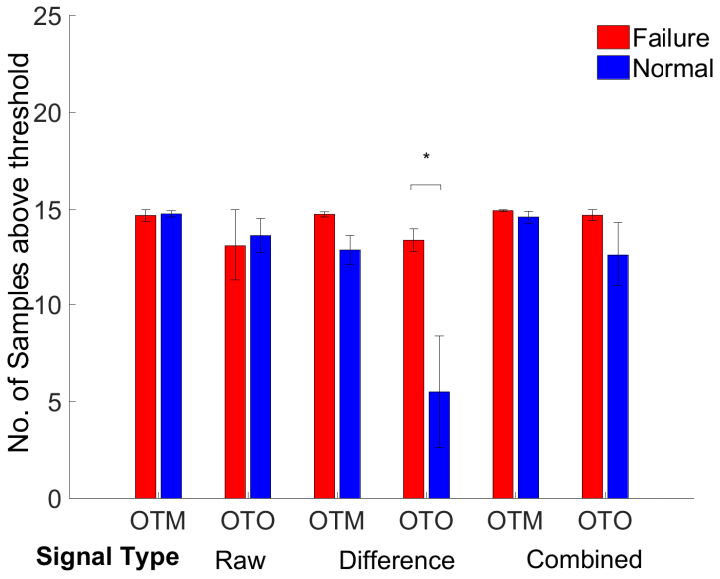


Figure 4-5: Number of samples above threshold for each robot condition. * indicates significantly different conditions in post-hoc comparisons. Error bars denote standard error of the mean.

Table 4.3: Analysis of Variance: Number of samples above threshold

Source	DF	Adj SS	Adj MS	F-Value	P-Value
TrainingSource	1	278.35	278.35	28.89	0 ***
SignalType	2	319.6	159.802	16.59	0 ***
RobotCondition	1	114.94	114.939	11.93	0.001 ***
TrainingSource*SignalType	2	160.95	80.474	8.35	0 ***
TrainingSource*RobotCondition	1	55.39	55.386	5.75	0.017 *
SignalType*RobotCondition	2	276.69	138.347	14.36	0
TrainingSource*SignalType*RobotCondition	2	134.21	67.106	6.97	0.001 ***
Error	261	2514.6	9.634		
Total	272	3445.63			

Table 4.4: Analysis of Variance: Lead time

Source	DF	Adj SS	Adj MS	F-Value	P-Value
TrainingSource	1	44.52	44.516	10.39	0.001 ***
SignalType	2	174.5	87.251	20.37	0 ***
RobotCondition	1	87.25	87.251	20.37	0 ***
TrainingSource*SignalType	2	89.03	44.516	10.39	0 ***
TrainingSource*RobotCondition	1	44.52	44.516	10.39	0.001 ***
SignalType*RobotCondition	2	174.5	87.251	20.37	0 ***
TrainingSource*SignalType*RobotCondition	2	89.03	44.516	10.39	0 ***
Error	261	1117.89	4.283		
Total	272	1567.37			

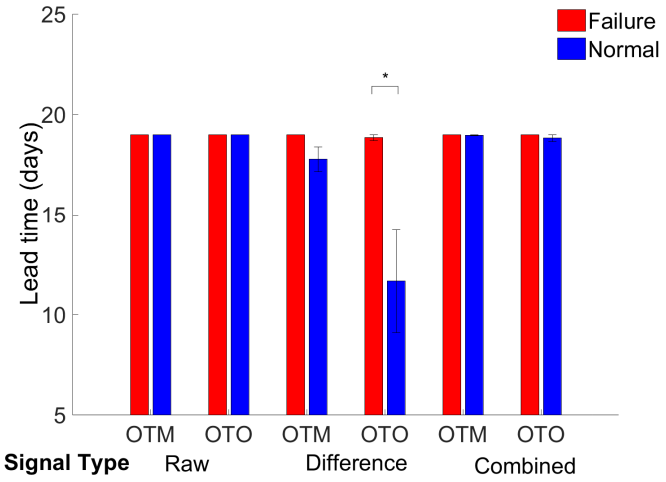


Figure 4-6: Lead time (measured as number of days prior to failure) for each robot condition. * indicates significantly different conditions in post-hoc comparisons. Error bars denote standard error of the mean.

To check for robustness against normality assumptions additional analysis was performed using generalized linear models (GLMs) [68]. GLMs can be used to find the influence of factors and their interactions on the response variable when ANOVA assumption are not met. GLMs are regression models and consist of a random component and a systematic component. The choice of probability distribution should be based on response variable. The systematic component depends on how the exploratory variables relate to the mean of the response variable after a transformation called the link function [69]. GLM models were

generated by using the following probability distributions and link functions:

- Response variable COV of Q residual (Variable type: Continuous): Gamma distribution (link "log")
- Response variable Lead time (Variable type: Count): Poisson distribution (link "log")
- Response variable Number of samples above threshold (Variable type: Count): Poisson distribution (link "log")

Comparison of GLM models for response variable COV of Q residuals, lead time and Number of samples above threshold were made (Tables 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11). It was noted that factors influencing COV of Q residual is same as that observed in ANOVA. However, the results of GLM analysis for lead time and number of samples above threshold are different from ANOVA results. The attribution for this difference associated with significant violation of normality assumptions for lead time and number of samples above threshold.

Formulation of models 1-8 considered for this analysis are given in Table 4.5. All three descriptors are considered as response variables.

Table 4.5: Models considered for GLM analysis

Model 1:	Response variable $\sim$	1
Model 2:	Response variable $\sim$	$TrainingSource$
Model 3:	Response variable $\sim$	$TrainingSource + SignalType$
Model 4:	Response variable $\sim$	$TrainingSource + SignalType + RobotCondition$
Model 5:	Response variable $\sim$	$TrainingSource + SignalType + RobotCondition + TrainingSource * SignalType$
Model 6:	Response variable $\sim$	$TrainingSource + SignalType + RobotCondition + SignalType * RobotCondition$
Model 7:	Response variable $\sim$	$TrainingSource + SignalType + RobotCondition + TrainingSource * SignalType + SignalType * RobotCondition$
Model 8:	Response variable $\sim$	$TrainingSource + SignalType + RobotCondition + TrainingSource * SignalType + SignalType * RobotCondition + TrainingSource * SignalType * RobotCondition$

Table 4.6: GLM Analysis: Coefficient of Variation of Q Residual

Model no	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	272	142.55			
2 (w.r.t 1)	271	142.52	1	0.04	0.7788
3 (w.r.t 2)	269	135.99	2	6.52	0.0009 ***
4 (w.r.t 3)	268	126.87	1	9.12	0.0000 ***
5 (w.r.t 4)	266	126.23	2	0.64	0.5039
6 (w.r.t 4)	266	121.94	2	4.93	0.0051 **
7 (w.r.t 6)	264	121.38	2	0.56	0.5504
8 (w.r.t 7)	261	119.13	3	2.25	0.1859

For Cov of Q, model 4 is found to be statistically significant with minimal residual deviance. Significant factors that contribute to model 4 are given in Table 4.7.

Table 4.7: GLM Analysis: COV of Q. Significant factors that contribute to model 4

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			272	142.55	
TrainingSource	1	0.04	271	142.52	0.7768
SignalType	2	6.53	269	135.98	0.0008 ***
RobotCondition	1	10.85	268	125.13	0.0000 ***

Table 4.8: GLM Analysis: Lead Time

Model no	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	272	159.83			
2 (w.r.t 1)	271	157.75	1	2.08	0.1496
3 (w.r.t 2)	269	153.79	2	3.96	0.1382
4 (w.r.t 3)	268	151.50	1	2.30	0.1297
5 (w.r.t 4)	266	146.88	2	4.61	0.0995
6 (w.r.t 4)	266	146.60	2	4.90	0.0865
7 (w.r.t 6)	264	141.99	2	4.61	0.0995
8 (w.r.t 7)	261	131.92	3	10.07	0.0180 *

For lead time, model 8 is found to be statistically significant with minimal residual deviance. Significant factors that contribute to model 8 are given in Table 4.9

Table 4.9: GLM Analysis: Lead Time. Significant factors that contribute to model 8

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			272	159.83	
training.type	1	2.08	271	157.75	0.1496
SignalType	2	3.96	269	153.79	0.1382
RobotCondition	1	2.30	268	151.50	0.1297
$TrainingSource * SignalType$	2	4.61	266	146.88	0.0995
$SignalType * RobotCondition$	2	4.90	264	141.99	0.0865
$TrainingSource * RobotCondition$	1	2.83	263	139.16	0.0924
$TrainingSource * SignalType * RobotCondition$	2	7.24	261	131.92	0.0268

Table 4.10: GLM Analysis: Number of samples above threshold (NSAT)

Model no	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	272	302.02			
2 (w.r.t 1)	271	287.79	1	14.23	0.0002 ***
3 (w.r.t 2)	269	280.50	2	7.29	0.0261 *
4 (w.r.t 3)	268	277.36	1	3.14	0.0765
5 (w.r.t 4)	266	267.70	2	9.66	0.0080 **
6 (w.r.t 4)	266	269.25	2	8.11	0.0174 *
7 (w.r.t 6)	264	259.59	2	9.66	0.0080 **
8 (w.r.t 7)	261	241.66	3	17.93	0.0005 ***

For number of samples above threshold (NSAT), model 8 found to be statistically significant with minimal residual deviance. Significant factors that contribute to model 8 are given in Table 4.11

Table 4.11: GLM Analysis: NSAT. Significant factors that contribute to model 8.

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			272	302.02	
$TrainingSource$	1	14.23	271	287.79	0.0002 ***
$SignalType$	2	7.29	269	280.50	0.0261 *
$RobotCondition$	1	3.14	268	277.36	0.0765
$TrainingSource * SignalType$	2	9.66	266	267.70	0.0080 **
$SignalType * RobotCondition$	2	8.11	264	259.59	0.0174 *
$TrainingSource * RobotCondition$	1	4.13	263	255.46	0.0421 *
$TrainingSource * SignalType * RobotCondition$	2	13.80	261	241.66	0.0010 **

4.5 Discussion and conclusion

Our results show that the failure in robots can be detected using coefficient of variation of Q-residual while using signal type as absolute difference and training source as OTM. Using lead time and number of samples above threshold as the descriptor, the failure can be detected if training source is OTO and signal type is absolute difference.

Contrary to our expectations in hypothesis 1, training source was not found to significantly affect COV across different conditions. However, the combination of robot condition and signal type was significantly different across other two descriptors. Further analysis showed that it is possible to use a single reference robot to generate training data for testing multiple scenarios if absolute difference is used as the signal type for training the PCA model. The OTO strategy is task and robot dependent and is therefore expected to reflect any variability present in different robots. On the other hand, in the OTM strategy, we are combining data from multiple robots to develop a single model to test and therefore likely to dilute variability within the robot and affect the performance of failure detection.

In agreement with hypothesis 2, signal type affects the ability to detect a failure in robots. In particular, all descriptors were significantly different for different signal types (main effect). For signal type as absolute difference, all descriptors were able to differentiate between a failure and a normal robot at least seven days before failure. The absolute difference is a measure of change in robot operation and is expected to be similar across different tasks as long as the robot operates normally without any sudden changes. The sampling rate of once per day possibly further reduced any anomalous changes due to task stop and start operations that occurred within a single day.

The significant difference in lead time for detecting failure as the time when the first data point is detected above threshold is an expected result for OTO. At the same time, the high variability in Q-residual itself implies that it is difficult to use a single threshold that can be used across all scenarios. In this respect, the lead time can be used as a warning to further process the data over the next few days for computing COV and number of samples above threshold. This can help in reducing the number of false positives. Between training sources, we find that while OTO finds significant difference between failure and normal robots when

using number of samples, OTM is able to register a similar change with COV. In an industrial setting, a combination of these training strategies can be adopted. Using OTM approach, effort on training model for every robot can be avoided and help to identify the robots that may require closer inspection.

Future studies will focus on collecting more data to highlight significant differences and contributing factors to different types of failures and identifying the correlation between events and measurements in time domain. In our analysis, we have considered measurements that are periodic as well as event driven. In ongoing work we seek to extend the current approach to study the effect of measurement type in prediction capability.

Part III

Archival journal articles

5

Data Preprocessing for Improving Classifier Performance in Detecting Gearbox failures in Industrial Robots[1]

Abstract: Gearbox failures cost thousands of lost production hours in plants that use industrial robots. In this context, an automated monitoring system that can warn the user of an impending failure can save precious resources. This problem has been addressed in many other domains through the use of machine learning approaches. However, standard machine learning algorithms are limited in their ability to detect gearbox failures, mainly due to task variability arises from robot-specific data. To improve detection performance of machine learning approaches, we propose techniques to curate the data prior to building a classification model. In a systematic hypothesis-driven study exploring the effect of different preprocessing techniques, we evaluate training data augmentation with estimated measurements, data differencing to suppress task dependence, inclusion of local variation, and selection of principal components on data collected from 26 industrial robots from the field. Our results show that preprocessing techniques improve the failure detection performance.

[1]This chapter is a slightly modified version of paper published in IEEE Transactions on Industrial Informatics 16, no. 1 (2019): 193-201 and has been reproduced here with the permission of the copyright holder

5.1 Introduction

Industrial robots are vital equipment to improve productivity, quality and safety in many applications (Fig 5-1). In a typical manufacturing line the loss of production from an unexpected failure of an industrial robot can cause heavy economic losses. At the same time, without a clear way to predict failure, overcautious maintenance schedules can lead to unproductive downtimes. Therefore robust and reliable failure detection is an active area of research [70, 71].

Failures in industrial robots can occur both in electrical and electronic components such as sensors, encoders, electrical drives and computer control systems [72] and mechanical components such as the mechanical structure, gearbox, actuators, and cables [73]. Approximately 45% of failures are attributed to mechanical problems [74]. Being a highly complex and coupled system detecting the onset of failure in industrial robots is a difficult problem.

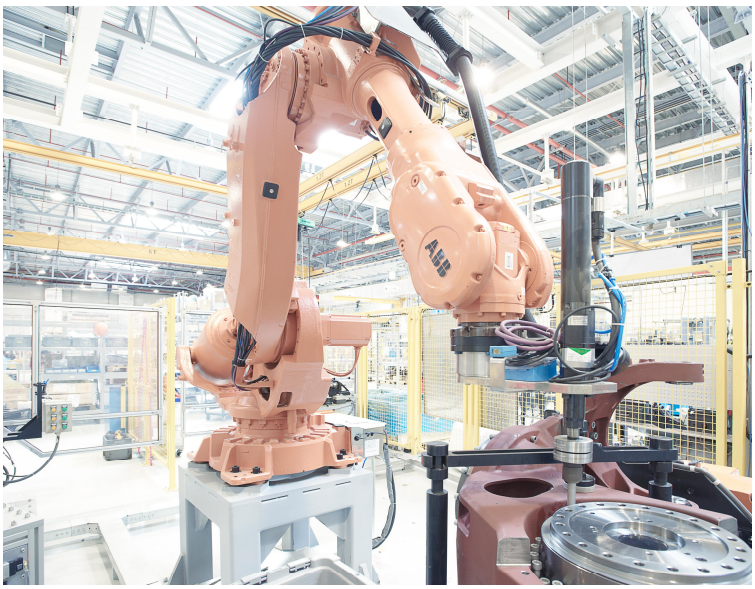


Figure 5-1: An example of six axis industrial robot similar to the one that was monitored in this study.

Failure detection in industrial robots can be broadly classified into knowledge-based, model-based, and data-driven methods [75]. While knowledge-based systems rely on the experience of expert users to detect failure, these are typically conservative leading to high

false alarm rate [76]. Model-based methods rely on a mathematical representation of the system to recreate the fault scenarios witnessed during robot operation [76]. However, there can be a number of factors responsible for a fault which may not be captured by a mathematical model, thus posing the risk of low detection rate or attributing the cause of the fault wrongly [76]. By contrast, machine learning methods use measurement data to build a detailed representation of the robot operation process, which in turn can then be used to classify anomalous situations based on deviation from a statistical model of normal operation process [71].

Machine learning provides a broad array of classification algorithms that can be used to detect faults in industrial robots. However, in their standard form, most of the classification algorithms such as logistic regression and support vector machines are designed to deal with static data, where examples are independent and identically distributed [77]. On the other hand, robot operation data is typically available in the form of time series implying time dependence. Standard classification algorithms cannot be directly used with time series data due to high feature correlation, high dimensionality, and large amount of noise [78]. Further, nearly all classification algorithms require large volumes of data to capture the wide variety of conditions in order to perform reliably [76]. In such scenarios, data preprocessing can be used to highlight the temporal and low-dimensional features and may therefore provide a means for extending the vast number of machine learning approaches to detect faults in robot operation.

In this paper, we propose four new preprocessing techniques for implementation on training data used to identify gearbox failures in industrial robots (Fig. 5-2). First, since the dataset consists of robots that perform different tasks, we investigate if taking the first order difference of the signal will improve classification accuracy across all robots, possibly due to task invariance [79]. Second, inspired from methods used to detect sudden crashes in ecosystems and financial markets, which show that change in variance in time-series can be a reliable predictor of a sudden event [80], we investigate the effect of including local variation in training data on classification performance. Third, to evaluate if augmenting the measurement data with original values can improve classification accuracy, we estimate actual torque and speed values from aggregated measurement data typically available in the

data stream from connected systems, and concatenate the same to the measurement data prior to training. Finally, use of correlated and noisy measurements can result in overfitting in classification [81]. Therefore, we investigate if replacing the original high-dimensional data with principal components improves classification performance [76]. Accordingly, we test the following hypotheses on monitoring data obtained from the field towards improved detection of gearbox faults.

H1: Signal differencing improves fault detection performance

H2: Inclusion of local variation improves fault detection performance

H3: Inclusion of estimated torque and estimate speed improve fault detection performance

H4: Selecting principal components of the data improves fault detection performance

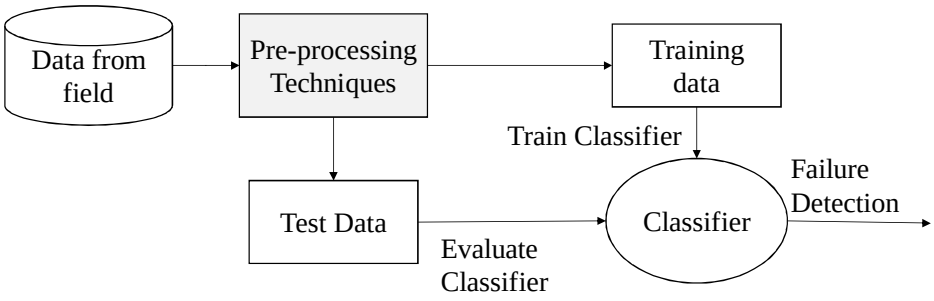


Figure 5-2: Research objective - evaluate the effect of meaningful data preprocessing techniques on failure detection in industrial robots.

We analyze the effect of these preprocessing techniques on four different classification algorithms that are selected on the basis of ease of use, popularity, and their ability to tradeoff bias and variance issues in classification from training data [81]. Specifically, we use logistic regression (LR), which has high bias and low variance and, support vector machine (SVM), which has low bias and high variance [82]. In order to obtain a classifier that balances bias and variance, we further use two type of ensemble methods: random forest (RF) and ensemble stacking (ES) [81]. Other machine learning algorithms such as deep learning and neural networks may also be used to detect gearbox failures [83], however, these algorithms

typically require large number of samples for each type of robot (normal and failed) [84] than what is available to us. Accordingly, their effectiveness on limited datasets available from the field has not been tested to the best of our knowledge.

Section 2 explains classification methods and performance evaluation of the classification models. Section 3 describes data collection, preprocessing, training, the testing strategy adopted in applying classification algorithms, and the procedure for data analysis. Section 4 provide the results of classification methods along with discussions followed by conclusion in Section 5.

5.2 Classification methods and their performance evaluation

Classification is the task of learning a target function f that maps each attribute set X to one of the predefined class labels [82]. This section discuss binary classification and the performance of such classifiers.

5.2.1 Binary classification

A binary classifier is a function $f : X \to \{0, 1\}$ that maps a space of input measurements to binary labels Y. Robots are monitored to get data, which can be used to infer the state of the the robot. The data are sampled at different time instances and are here denoted by X_k, where the $k = 1, ..., N$ denotes the sample number. Each sample contains M measurement components, $X_k = [X_{k,1}^T, X_{k,2}^T, .., X_{k,M}^T]^T \in \mathbb{R}^M$ examples of such components include angular distance, torque, and speed, in general the components can be vectors of different length.

Therefore, in the context of classifying a robot as normal or failed $Y_k \in 0, 1$, we seek the following binary relationship between the state of the robot, and the measurement vector X_k,

$$Y_k = \begin{cases} 0, g(X_k, \alpha) < \gamma \\ 1, \text{otherwise}, \end{cases} \tag{5.1}$$

where $Y_k = 0$ implies a normal robot, and $Y_k = 1$ implies a failed robot; g comprises a model of the robot performance in terms of measurements, γ is the associated threshold, and α denotes the set of classifier dependent parameters.

LR is a supervised classification method that bounds the measurement output to lie between 0 and 1. In this sense, the output can be used to build a probabilistic model of the training data that quantifies the probability of a certain event. For example, if the measurement X_k belongs to a failed data then g quantifies in some sense the probability of failure $p(Y_k = 0|X_k)$ as

$$g(X_k, \alpha) = \frac{1}{1 + \exp(-\alpha_0 + X_k^T \alpha)}, \tag{5.2}$$

where $\alpha \in \mathbb{R}^M$ is an unknown model parameter vector, α_0 is an unknown scalar term, and g is a sigmoidal function that constrains the output between 0 and 1. The parameters of a logistic regression model are estimated by the probabilistic framework called maximum likelihood estimation. Using the training data, the model fitting routine searches across different values of these two parameters to find a combination that, given the observed training data, maximizes the likelihood of the binomial distribution [82].

With SVM the principle is to find a decision boundary that maximizes the separation between two types of data, such as measurements from failed and normal robots. Depending on the number of features, the decision boundary can be a line, or a hyperplane. For example, with two features, the data can be represented on a two-dimensional plane and the decision boundary will be a line. For n features, the decision boundary will lie on a n-1 dimensional hyperplane. The output of the SVM algorithm can in turn be constrained to lie between 0 and 1 with a function [81],

$$g(X_k, \alpha) = \frac{1}{1 + \exp(-\beta_0 - \sum_{l=1}^{N} \alpha_l K(X_k, X_l))}, \tag{5.3}$$

where K is the Gaussian radial basis kernel function, X_k and X_l are two samples in the data-set, and β_0 and α are parameters to be determined. Tuning of parameters are performed using Classification and Regression Training (CARET) package in R language [67] and it helps to automatically choose the optimal values for the model tuning parameters, where

optimal is defined as values that maximize the model accuracy.

We consider two ensemble methods, namely RF and ES. RF is a classifier that aggregates predictions from multiple decision trees using majority voting. The probability of failure can be calculated by [81] $g(X_k, \alpha) = \frac{1}{B} \sum_{b=1}^{B} Tr(X_k, \alpha_b)$, where $Tr(X_k, \alpha_b)$ represents each classification tree and B is total number of trees. On the other hand, in ES, an ensemble of classifiers are first trained with the training data to create a set of base classifiers. The base classifiers are then used to generate a new dataset of, measurements and associated estimated labels, that is merged with the original training data to train a meta-classifier. The meta-classifier can be any one of the base classifiers. This approach of stacking multiple classifiers addresses the statistical, computational, and representational shortcomings of individual classifiers [85], and has been shown to improve classification performance both in theory [86] and in experiments [87]. In our implementation, we used the base classifiers of LR and SVM for their competing effects on the bias and variance of estimated values. Our meta-classifier is LR(Fig. 5-3). We used LR model as meta classifier because of its simplicity and ability to make inferential statements about model terms and providing a smooth interpretation of the predictions made by the base classifiers [82].

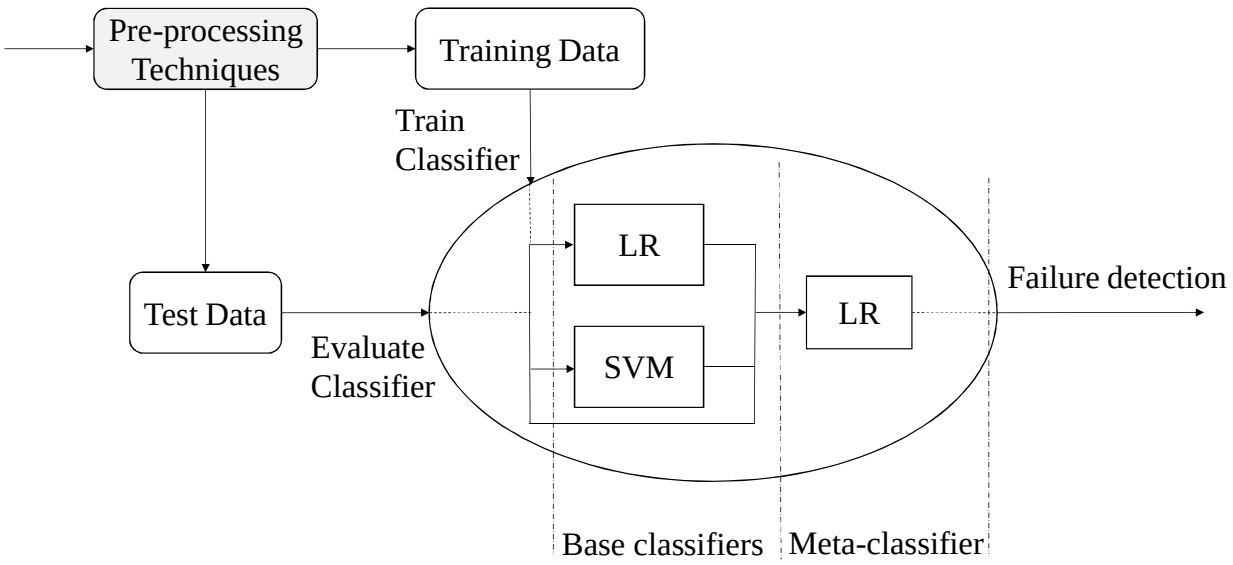


Figure 5-3: Ensemble Stack: LR and SVM are base classifiers and LR is meta- classifier.

5.2.2 Classification performance

A widely used metric for evaluating classifier performance is receiver operating characteristic (ROC) curve [81]. The ROC curve marks the true positive rates (TPR) against false positive rates (FPR) for different values of decision threshold γ. An ROC curve has operating points that corresponds to the "cutoff" value which provide optimal performance in a given sense, such as the highest true positive rate and lowest false positive rate. The operating point corresponding to higher TPR and lowest FPR on the ROC curve is located by finding the point that is closest to the upper left corner.

In an ROC, the area under curve (AUC) is used to measure the performance of classifier and is frequently applied for comparison among different classification methods. A high AUC denotes better accuracy with AUC=1 for 100% accuracy and AUC=0.5 for chance level accuracy.

5.3 Data acquisition and processing

This section introduces the data and the acquisition of the data in Section 5.3.1, Section 5.3.2 explains the preprocessing of data, Section 5.3.3 describes the selection of data for training and finally Section 5.3.4 discusses the data analysis procedure.

5.3.1 Data acquisition

The dataset consists of data from 26, six degree of freedom (DOF) industrial robots. Out of 26 robots, 13 have reported gearbox failures while 13 are normally operating robots with no reported gearbox issues. This data was acquired over more than four years and represents one of the challenges in data-driven methods developed using real data. The robots are used in different environments and in different processes such as welding, painting, and palletizing. Size and load of the robots also varies. Fig. 5-4 shows the normalized average estimated speed and torque for all 26 robots that indicates task variability across robots. Failure description and date for the robots with reported gearbox failure was received from engineers and it is based on their expert judgment to assess the condition of the robot. The

minimum number of days for which the data was available across all failed robots is 23 days.

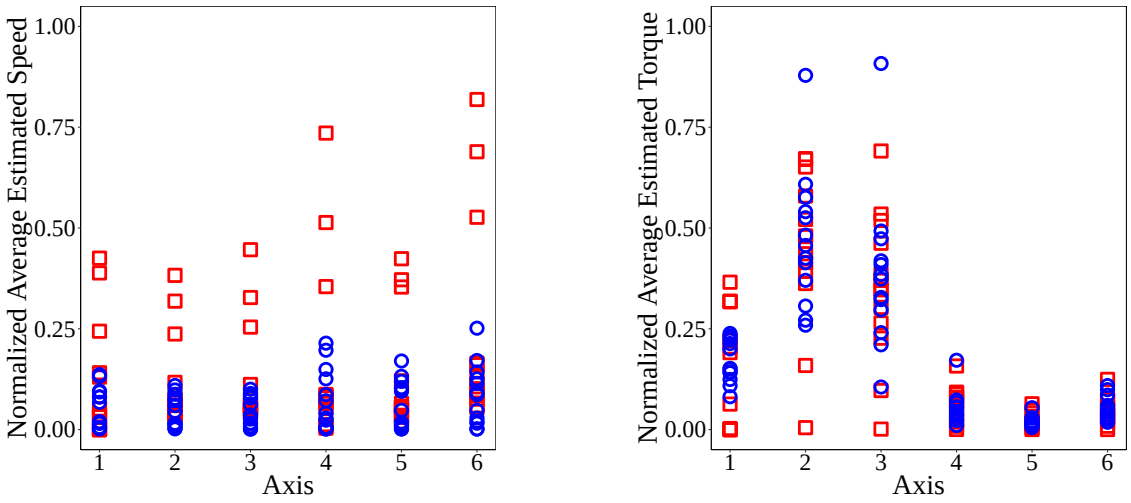


Figure 5-4: Average speed (a) and torque (b) for each of the six axes of failed (red squares) and normal (blue circles) robots. The speed and torque data are normalized with respect to maximum values.

Measurements collected from the robot systems can be classified as controller level and manipulator level. Measurements such as temperature, CPU load are examples of controller level measurements while speed and torque are examples of manipulator level measurements. For this study, the focus is on gearbox failures and for this purpose manipulator level measurements [73] are useful. The following measurements were considered here: i) speed indicator, ii) torque indicator, iii) accumulated joint angle, iv) joint speed, and v) joint torque. Speed and torque indicator values are invertible functions of actual speed, actual torque, production time, and production time difference, and are computed using expert knowledge [2]. Joint angle, speed, and torque measurements were available as histograms with 12 values per measurement, each representing the fraction of time during which robot performs in a specific fractional range of maximum design value. As a result a total of 36 values per manipulator axis was collected. Therefore, a total of 38 values (36 plus speed and torque indicators) were available per axis resulting in a robot sample $X \in \mathbb{R}^{228}$. Fig. 5-5 shows a schematic of the data collection approach and preprocessing techniques used for the study.

[2]Specific information about the functional forms are not disclosed due to company policy.

5.3.2 Data preprocessing techniques

Measurements can be collected once to four times per day by the current data collection system. For most robots only one measurement is collected per day and therefore it was decided to resample the data to have the same sampling frequency (1 measurement per day) for all robots. Specifically, if a robot had more than one sample per day, we selected the maximum value across samples to capture the highest intensities in such quantities during normal operations.

Data is collected using a schedule in the controller which means that the data is collected at the same time every day. Since the measurements represent average values it is not necessary that the robot is in a certain condition when the measurement is taken (except that it has to be switched on).

Compactly representing joint angle, speed and torque measurements

To reduce the size of the data vector, the histograms for joint angle, speed and torque for axis j at instant k, given as a subset of X_k,

$$X^j_{hist,k} = \{X^{j,1}_k, X^{j,2}_k, \ldots, X^{j,12}_k\}$$

can be transformed into one single value using the Kullback Leibler(KL) divergence [88]. The KL divergence is computed as distance from the histogram of first sample to histogram of k^{th} sample,

$$D(X^j_{hist,k}||X^j_{hist,1}) = \sum_{i=1}^{12} X^{j,i}_{hist,k} \log\left(\frac{X^{j,i}_{hist,k}}{X^{j,i}_{hist,1}}\right) \tag{5.4}$$

This representation allows us to transform the histograms into a single value per histogram type and axis, reducing the size of X for the robots from $X \in \mathbb{R}^{228}$ to $X \in \mathbb{R}^{30}$.

Data differencing

Since the test dataset consists of robots that are used in different tasks with significant variation in speeds and loads (Fig. A.1), we investigate if differencing time series can reduce task dependence. Accordingly we use the absolute difference of consecutive measurements of

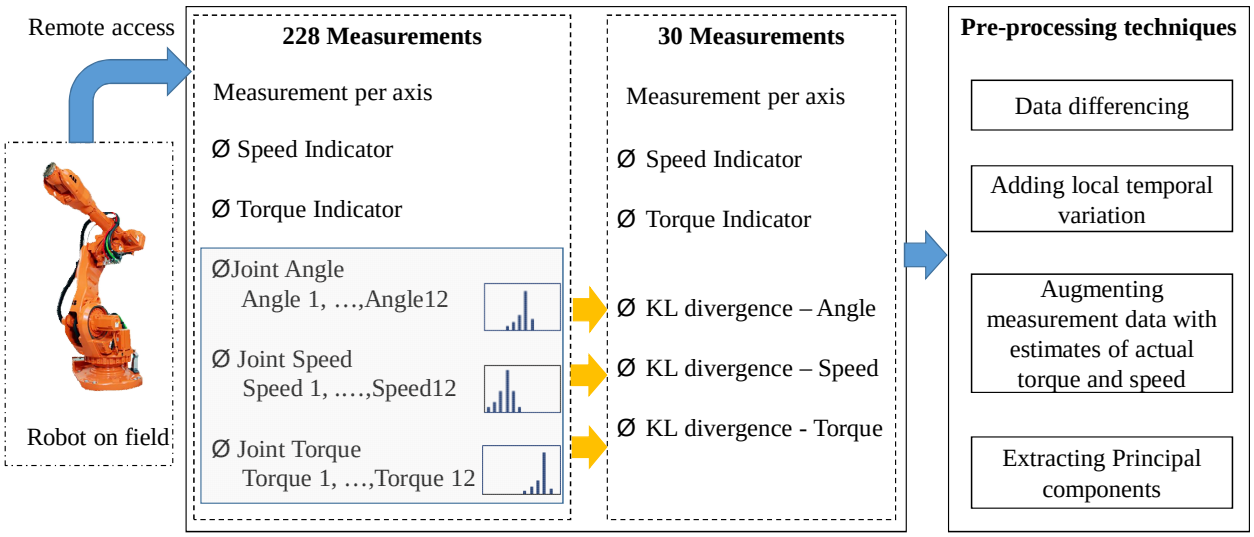


Figure 5-5: Overview of data collection approach and preprocessing techniques for the study.

the time series as $|X_k - X_{k-1}|$ instead of the raw signal for classification. Similarly, the KL divergence measure is computed between successive measurements as $D(X^j_{hist,k}||X^j_{hist,k-1})$.

Including local temporal variation

It has been shown that change in variance in time-series can be a reliable predictor of a sudden event [80], therefore we include local variation to improve detection performance. Specifically, we evaluate the influence of adding variation by considering measurements along a moving window of 3 and 5 days. In this case the vector $X_k \in \mathbb{R}^M$, is replaced by a concatenated vector $[X^{\mathrm{T}}_{k-n+1}, X^{\mathrm{T}}_{k-n+2}, \ldots, X^{\mathrm{T}}_k]^{\mathrm{T}} \in \mathbb{R}^{nM}$ for a variation of $n = 3$ and $n = 5$ days. The size of the concatenated vector increases n times the original size of X_k.

Augmenting measurement data with estimates of torque and speed

The measurement data consisted of speed and torque indicators in addition to frequency distribution of torque and speed. To investigate if augmenting these measurements with estimates of torque and speed improves failure detection performance, we invert the functional forms of speed and torque indicators to estimate instantaneous values of torque and speed and add the same to the existing measurement data for each robot. Fig. 5-6 shows plots of normalized estimated torque for a failed and a normal robot.

After above initial pre-processing, total number of measurements in the dataset is 42 (seven measurements per manipulator axis) listed in Table 5.1.

Table 5.1: List of measurements per axis after initial preprocessing

Measurement per axis	Unit	Description
Speed indicator	rad/s	Function of actual speed and production time
Torque indicator	Nm	Function of actual speed, actual torque and production time
Estimated Speed	rad/s	Derived by inverting the functional form for speed indicator
Estimated torque	Nm	Derived by inverting the functional form for torque indicator
KL Divergence - Speed	bits	
KL Divergence - Angle	bits	KL divergence between two measurement (angle, speed, torque)
KL Divergence - Torque	bits	distribution

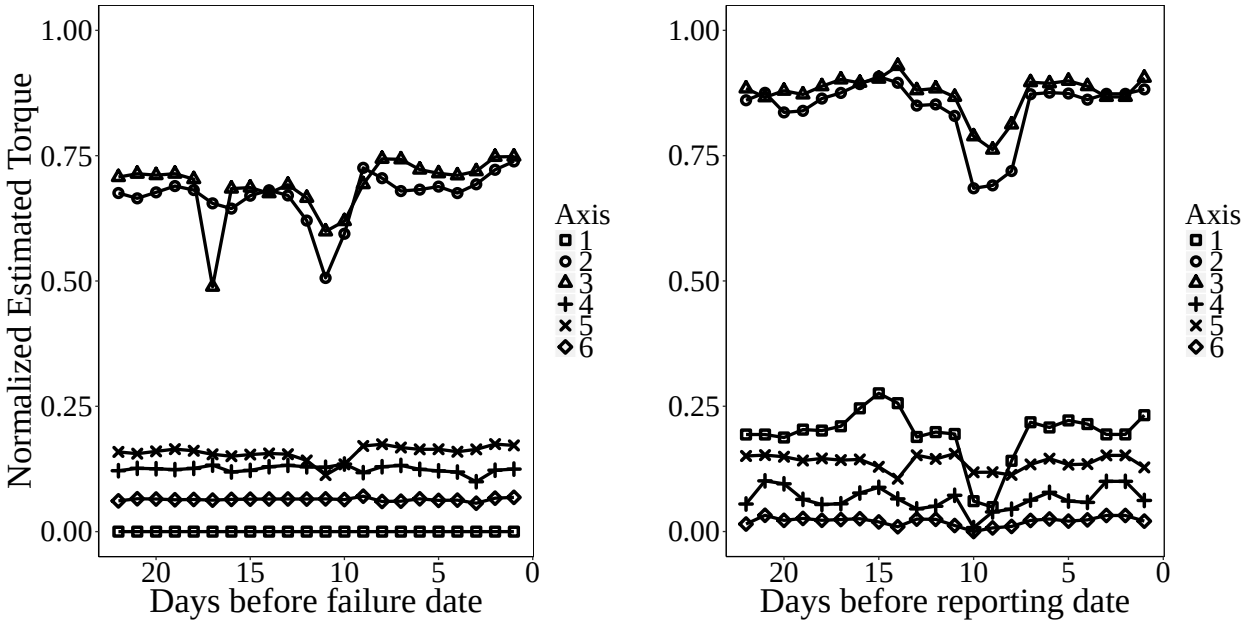


Figure 5-6: Estimated torque for each of the six axes of a normal robot (a) and a failed robot (b). Torque values are normalized with respect to the maximum values.

Extracting Principal components

Despite the reduction of the number of measurements to 42, these are still a large number of measurements for analysis. To retain the components that explain maximum variation in data we use principal components analysis (PCA). This helps create uncorrelated measurements, which improves performance in methods such as LR [76]. Given n instances of m different measurements, the PCA algorithm eigen-decomposes the input data into $u \leq m$ principal components, where u is selected to account for more than ninety percent of the variation in data [89]. With respect to our training dataset, the PCA algorithm reduces 552 instances of 42 measurements to 10 principal components.

Combinations of preprocessing techniques

There are total of sixty-six combinations of possible preprocessing techniques. Out of these we only consider the ones that are compatible with our hypotheses. For example, we exclude conditions where signal is differenced (+diff) prior to adding estimated values (+estM), because then we lose any benefits of task invariance that may have resulted due to signal differencing. Similarly, +pca should be applied after estimated values have been added and signal differencing has taken place to enable reduction of dimension after including all signal types.

The combinations of preprocessing techniques used are listed in Table 5.2: control indicates data without any preprocessing, +diff indicates absolute difference of the raw time-series data, +estM represent the inclusion of estimated torque and speed, which as noted before, are not available as part of the original dataset, +pca indicates preprocessing of data sets with PCA to reduce the dimensionality, and var# represents the inclusion of local variation of # days in the data set.

5.3.3 Training data selection

To ensure that classifier performance is not dependent on a specific subset of the data, we develop a strategy to exhaustively evaluate the influence of pre-processing methods on classifier performance from a limited dataset. Considering a failure prediction capability of 1, 3, and 5 days, we select data that is ten days prior to the prediction capability. For example the training data for a 3 day prediction capability was selected as T-13 to T-3, where T is the day of failure. The value of ten days is based on a sensitivity analysis where it was found that failed and normal robots can be distinguished for up to ten days prior to failure.

Each training sample is assigned with a class label depending on the state of the robot. Procedure 1 describes our approach to train and test each of the classifiers.

We implement the classification methods on a personal computer (Intel Core i5 3.30 GHz processor, 8 GB memory, Windows 7 OS) using Classification and Regression Training (CARET) package in R language [67] that have many modeling functions for classification

Procedure 1 Training and testing of classifiers

Input : Preprocessed samples from Failed (F) and Normal (N) robots
Output: Area under ROC curve (AUC)

1: **for** each robot i in the set F of failed robots **do**
2: **for** each robot j in the set N of normal robots **do**
3: Create a training data set $\{F \cup N\} \setminus \{i, j\}$ by combining data from all failed and normal robots except robots i and j
4: Build the classifier using the training data set
5: Create a test data set $\{i, j\}$ by combining data from failed robot i and normal robot j
6: Test the classification model with test data and return failure probability for each sample
7: Evaluate the classification performance of model using Area Under ROC curve
8: **end for**
9: **end for**

problems.

5.3.4 Data analysis

To evaluate the influence of data pre-processing techniques, we statistically compare the performance of each of the four classification methods after data preprocessing. Specifically, we classify failed and normal robots after data processing as described in Table 5.2, and use AUC as performance measure.

We compared variance across preprocessing techniques, with the Levene's test [90]. Our analysis showed that the AUC does not have equal variance across preprocessing techniques. Therefore, we use Welch's Analysis of variance (ANOVA) [91] to identify the effect of preprocessing (independent factor) on classification performance (dependent factor) for each of the four classification methods. This is performed across all of fifteen (Table 5.2) combinations of preprocessing techniques. If the Levene's test shows that variance are not significantly different, then regular ANOVA is performed. The procedure for comparing performance is summarized in Procedure 2. In all the examples of statistical tests-significance level is set to $p < 0.05$.

We use R [67] to perform these tests: ANOVA are performed with R package car [94], Levene's test is performed with R package lawstat [95] and Games Howell post-hoc test is

Procedure 2 Comparing classification performance

```
Perform Levene's test for equal variance of AUC across preprocessing techniques
if variances are different (p < 0.05) then
    perform Welch's ANOVA
    if preprocessing technique is found to be significant (p < 0.05) then
        apply Games Howell post-hoc test [92]
    end if
else
    perform ANOVA [93]
    if preprocessing techniques is found to be significant (p < 0.05) then
        apply Tukey's HSD as post-hoc test.
    end if
end if
```

performed using package userfriendlyscience [96].

5.4 Results of classification

Fig. 5-7 shows a summary of classification performance as each preprocessing method is applied on the original dataset, while Table 5.2 lists the mean AUC for each preprocessing technique across all classification methods. In particular, we see that preprocessing steps increase the values of AUC while decreasing its variance and that these changes are dependent on the type of classification method used. Briefly, while all methods show reduction in variance, only LR, SVM and ES show increased classification performance as measured by AUC.

Table 5.2: Mean AUC (with respect to standard deviation) of each classifier for select combination of preprocessing techniques. Mean AUC greater than 0.7 is highlighted in bold.

Preprocessing techniques	LR	SVM	RF	ES
control	0.47 (0.34)	0.61 (0.48)	0.66 (0.45)	0.64 (0.42)
+estM	0.45 (0.33)	0.59 (0.47)	0.69 (0.42)	0.58 (0.38)
+diff	0.51 (0.23)	0.48 (0.30)	**0.71** (0.28)	0.48 (0.30)
+pca	0.69 (0.42)	0.49 (0.50)	0.63 (0.47)	0.44 (0.49)
+var3	0.47 (0.34)	0.55 (0.49)	0.66 (0.44)	0.54 (0.49)
+estM+diff	0.54 (0.24)	0.61 (0.29)	**0.77** (0.27)	0.60 (0.30)
+estM+pca	0.50 (0.41)	0.66 (0.44)	0.61 (0.43)	0.67 (0.44)
+estM+var3	0.45 (0.33)	0.55 (0.45)	0.69 (0.43)	0.51 (0.46)
+diff+pca	0.61 (0.33)	0.60 (0.35)	0.63 (0.27)	0.61 (0.34)
+var3+pca	0.69 (0.42)	0.49 (0.50)	0.65 (0.46)	0.48 (0.48)
+estM+diff+pca	**0.72** (0.31)	0.65 (0.27)	0.69 (0.28)	0.64 (0.27)
+estM+var3+pca	0.45 (0.33)	0.55 (0.45)	0.61 (0.44)	0.51 (0.46)
+diff+pca+var3	0.67 (0.37)	0.62 (0.37)	0.68 (0.36)	0.67 (0.37)
+diff+var3+pca	0.67 (0.38)	0.61 (0.36)	0.64 (0.33)	0.62 (0.37)
+estM+diff+var3+pca	**0.75** (0.30)	**0.76** (0.27)	**0.77** (0.29)	**0.76** (0.27)

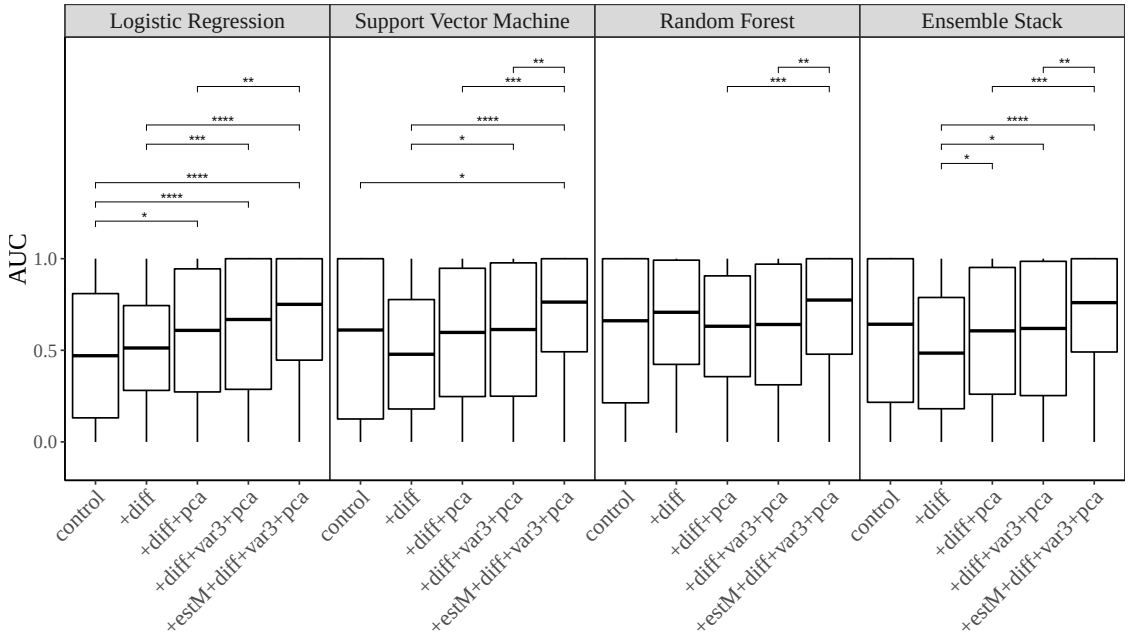


Figure 5-7: Effect of preprocessing on classification performance with training data as recent as 3 days before failure. Box plot centerline and edges denotes mean ± 1 standard deviation, and upper and lower tails denote maximum and minimum values obtained. Significant increase in mean value of AUC is denoted by *. $* : p < 0.05$, $** : p < 0.01$, $*** : p < 0.001$, $**** : p < 0.0001$.

Next, we provide results for the relevant pairs of preprocessing techniques related to the hypothesis presented in Section 5.1.

5.4.1 Signal differencing improves detection performance of gearbox faults in industrial robots (H1)

We find that signal differencing resulted in reduced variance across all classifiers individually and also in combination on a limited dataset with other preprocessing methods (Fig. 5-7 and Table 5.2, see values in Appendix section, Tables A.2 and A.3). For SVM and ES, +diff created significant reduction in variance in all combinations of preprocessing methods. Mean AUC was also significantly increased for all classifiers when used in conjunction with other preprocessing methods. Specifically, improvement in mean AUC ranges between 22% (RF) to 40% (LR) with inclusion of diff along with +estM+var3+pca.

To determine if signal differencing improved task invariance of classification methods, we correlate the mean AUC per robot (averaged over the 13 different combinations of normal and failed robots) with the average estimated speed and torque for the preprocessing techniques +estM+var3+pca and +estM+diff+var3+pca. These techniques differed with respect to the presence of signal differencing only. Our results show that while there was no correlation between mean AUC and average estimated speed or torque across all classifiers ($p > 0.78$ for estimated speed and $p > 0.17$ for estimated torque) when signal differencing was present, mean AUC was correlated with average estimated speed for LR when signal differencing was absent ($p = 0.04$, $r^2 = 0.13$) with a negative slope (0.21) indicating that the performance degraded with tasks that involved higher speeds. Therefore, we conclude that differencing shows improvement in a dataset of industrial robots that were performing different tasks, and therefore serves to induce task invariance in classification.

5.4.2 Inclusion of local variation improves detection performance of gearbox faults in industrial robots (H2)

Adding local variation did not reduce variance individually when used in conjunction with pca and augmented with estimated measures (+estM). However, it resulted in reduced

variance with LR (Fig. 5-7 and Table 5.2, see values in Appendix, Tables A.2 and A.3). In the case of mean values, for SVM and ES, +var3 significantly increased mean AUC when used in conjunction with other preprocessing methods. Specifically, improvement in mean AUC ranges between 4% (LR) to 17% (SVM) with inclusion of var3 along with +estM+diff+pca. Further, we found the effect of size of the variation window is not significant between +var3 and +var5 (+estM+diff+var3+pca vs +estM+diff+var5+pca - LR: p=0.957 , SVM:p=0.217, RF:p=0.667, ES: p=0.124).

Inclusion of local variation in conjunction with other preprocessing techniques improved classification performance. The tolerance of gearbox degradation of industrial robot might be varied across the robots depending on robot application. Therefore variation in measurement across failed robot may not be similar. When we combine data from robots with varied degradation level to develop model, inclusion of local variation alone may not be very sensitive to detect failures.

5.4.3 Inclusion of estimated torque and speed improve detection performance of gearbox faults in robots in at least one of the classification methods (H3)

Adding estimated measures by itself did not reduce variance but, used in conjunction with other pre-processing methods, +estM, resulted in reduced variance across all classifiers when added with +diff+var3+pca method (Fig. 5-7 and Table 5.2, see values in Appendix, Tables A.2 and A.3). For SVM and ES, +estM created significant reduction in variance in three out of seven combinations of preprocessing methods.

Addition of estimated measures also significantly increased mean AUC values for all classifiers except LR when used in conjunction with other preprocessing methods. Specifically, improvement in mean AUC ranges between 12% (LR) to 24% (SVM) with inclusion of estM along with +diff+var3+pca.

It is likely that although estimated measures contained the same information as in the processed data output from the remote system and therefore did not show significant improvement in performance by itself. However, when preprocessed using signal differencing

and temporal variation, features that are predictive of failures were amplified adding to a general improvement in failure detection.

5.4.4 Selecting principal components of the data improves detection performance of gearbox faults in industrial robots (H4)

PCA preprocessing resulted in increased variance for LR (Fig. 5-7 and Table 5.2, see values in Appendix, Tables A.2 and A.3). In this case +pca created significant increase in variance in five out of six combinations of preprocessing methods.

PCA preprocessing also significantly increased mean AUC values for LR when used in conjunction with other preprocessing methods. In particular, mean value of AUC for LR improved around 34 % when PCA preprocessing was included (+estM+diff+pca).

Datasets with correlated measurements can lead to lower performance for LR and PCA transform original measurements to uncorrelated to principal components. Performance of other classifier methods such as SVM and RF are not sensitive to data with correlated measurements and therefore PCA preprocessing did not improve those classifiers ability to detect faults. However, PCA preprocessing reduced overall execution time of training and testing of SVM and RF classifier methods to at least one third execution time with original features and without any significant reduction in classification performance (Appendix, Table A.1).

5.5 Conclusion

In this paper, we test hypotheses related to the effect of combining four preprocessing techniques on classification performance in detecting gearbox failures in industrial robots.

We found that preprocessing techniques improved classification performance. Specifically, differencing reduced variance in AUC across all classification methods, inclusion of local variation in conjunction with other preprocessing techniques improved classification performance, among three out four classification methods (except LR), inclusion of

estimated measures along with other preprocessing methods show significant increase in performance. In addition, classification performance of LR method improved significantly with PCA preprocessing. Maximum improvement in mean value of AUC across classification methods was between no processing and +diff+estM+pca+var3. LR was the most receptive classification method whose improvement with data preprocessing was comparable with complex classifiers such as RF and SVM while executing in a fraction of the time (Appendix, Table A.1). In the presence of high-volume data it is possible that these preprocessing techniques may aid in reducing the complexity of neural network type approaches where data redundancy is a growing concern [97].

We note that the preprocessing techniques presented involve extra steps in addition to a direct application of machine learning. At the same time, these techniques improve the accuracy of failure detection in industrial robots, are intuitive and straightforward to implement and significantly reduce the execution time.

However, we foresee some challenges in assessing condition of new robots using this approach. We assume that robots are performing repetitive operations and higher variation in measurements associated with deterioration of gearboxes in robots. But, higher variation in measurements can also be attributed to frequent changes in the operating pattern of robot.

In the absence of high-volume real-world data, the field of fault prediction stands to gain from focused efforts on enabling machine-learning approaches that operate on limited data and can isolate features associated with mechanical failures. Further it is desirable that such failures can be detected independent of the task at hand. In this context, the current study aims to address these issues through preprocessing techniques that focus on improving task invariance, highlighting temporal variations, and isolating principal components.

The results presented here set the stage for developing a reliable procedure to prepare monitoring data from robots for use in fault monitoring and prediction. Future work will be focusing on addressing the challenges and subsequent improvement of classification models. Further, we validate these models using data from new set of robots and evaluate the quality of prediction.

6

A Transfer Entropy Based Approach for Fault Isolation in Industrial Robots[1]

Abstract: In this paper, we cast the problem of fault isolation in industrial robots as that of causal analysis within coupled dynamical processes and evaluate the efficacy of the information theoretic approach of transfer entropy. To create a realistic and exhaustive dataset, we simulate wear induced failure by increasing friction coefficient on select axes within an in-house robotic simulation tool that incorporates an elastic gearbox model. The source axis of failure is identified as one which has the highest net transfer entropy across all pairs of axes. In an exhaustive simulation study, we vary the friction successively in each axis across three common industrial tasks: pick and place, spot welding, and arc welding. Our results show that transfer entropy based approach is able to detect the axis of failure more than 80 percent of the time when the friction coefficient is 5% above the nominal value and always when friction coefficient is 10% above the nominal value. The transfer entropy approach is more than twice as accurate as cross-correlation, a classical time-series analysis used to identify directional dependence among processes.

[1]This chapter is a slightly modified version of paper published in ASME Letters in Dynamic Systems and Control (2021): 1-23, and has been reproduced here with the permission of the copyright holder

6.1 Introduction

Detecting and diagnosing faults in complex equipment such as an industrial robot requires technical expertise and onsite assessment that is often time consuming and repetitive. Therefore, remote monitoring solutions serve to provide a pre-assessment of problems in industrial robots by a robotic specialist from a distant location saving time and expense. Remote monitoring, however, can only help in determining if a failure has occurred, or is about to occur [98]—identifying the source of failure is something the specialist is expected to do in person. Because fixing faults involves bringing the right parts and components to the robot location, knowing which part to carry can reduce overall maintenance costs.

Faults can happen in both the manipulator and controller of an industrial robot. A typical industrial manipulator consists of rigid links called axes. Fault detection and identification of the failure axis in industrial robots are established areas of research [99]. This involves detecting the occurrence of fault (fault detection) and recognizing the location (fault isolation) [100]. Such methods have been successfully applied to various safety-critical systems such as nuclear plants [101] and flight control systems [102].

Fault detection and isolation in industrial robots can be broadly classified into knowledge-based, model-based, and data-driven methods[75]. While knowledge-based methods rely on user expertise, model-based methods rely on a mathematical representation of the system to detect and isolate the fault during robot operation [30]. Bittencourt et.al [103] for example, have used a detailed model with a maximum likelihood estimator to isolate the changes of friction. However, this approach needs re-estimation of friction parameters to adapt to variations due to robot model and gearbox type. As industrial systems become more complex it becomes even more difficult to develop a high fidelity physical model of the mechanical system, particularly when the machine is operating in an uncertain environment with a lot of external disturbances from the process. Data-driven methods mitigate this problem provided enough data is available [104], however they have been primarily used for fault detection by statistically distinguishing between normal and faulty processes.

In industrial robots, fault isolation entails identifying the axis of failure. As the joints are dynamically and tightly coupled in an industrial manipulator, faults in one joint can influence

another joint. In this context, it becomes important to determine the joint that causes the faults and the corresponding fault propagation route. This entails deriving the causal structure of the fault propagation. Methods that infer this type of directional dependence in coupled processes are required in a wide variety of fields and include signed directed graph modeling for fault propagation [105], information theoretic approach of transfer entropy in weather forecasting, neuroscience, and animal behavior [106], directed transfer functions, cross correlation analysis and partial directed coherence in [107].

Among these approaches, transfer entropy (TE) is especially useful in situations where the coupling between processes is not linear or time invariant. Transfer entropy (TE) is a model-free approach to detect causal relationship between time series, in that it does not presume any mathematical relationship between the two processes [108]. Lungarella et. al. [109] compared methods for quantifying the causal structure of bivariate time series extracted from systems with complex dynamical behavior and suggested TE as first method of choice if the given system is complex and with unknown dynamics. In this paper, we evaluate TE as a possible method for isolating failure in industrial robots.

In an exhaustive analysis, we simulate wear in each axes on a model of ABB IRB 6700, a six-axis manipulator, for three different types of common industrial tasks: pick and place, spot welding, and arc welding. Using realistic simulations with a model similar to an extended flexible joint dynamic model [110] that allow us to recreate a wide variety of scenarios, some of which are rare in industrial setups, we assess the sensitivity of our approach to intensity of failure, type of task, and location of the failure. We model wear as slow exponential growth in friction coefficients over 100 repeated cycles. The axis of failure due to wear is hypothesized as one that most affects other axes, quantified in terms of the maximum net transfer entropy across all pairs.

The contributions of this paper are: (i) we assemble a framework based on established methods [33, 111] for creating wear-induced faults in an industrial robot for different tasks and stages of wear, (ii) we test the hypothesis that TE implemented to mitigate its need for large data, will be able to detect the location of a wear induced fault (source axis) in an industrial robot, and (iii) we perform an exhaustive analysis of the accuracy of TE for fault isolation for all of the robot axes, on a variety of robot tasks for different wear percentages.

The axis of failure due to wear is hypothesized as the one that most affects other axes, quantified in terms of the maximum net transfer entropy across all pairs.

The paper is organized as follows: Section 2 provides a background of the robot simulation model, the friction model and transfer entropy. Section 3 describes the process of simulation of wear, data collection, prepossessing, and transfer entropy based isolation of source axis with wear. Section 4 presents the analysis and results along with discussions, followed by conclusion in Section 5.

6.2 Background

6.2.1 Robot model simulation

The simulation model used in this study is based on a prototype of a large six degrees-of-freedom (DOF) manipulator, similar to ABB IRB 6700. For a six-axis manipulator, the joint angles $\boldsymbol{\phi_a} \in \mathbf{R}^6$, velocities $\dot{\boldsymbol{\phi}}_\mathbf{a} \in \mathbf{R}^6$, accelerations $\ddot{\boldsymbol{\phi}}_\mathbf{a} \in \mathbf{R}^6$, and the required torques at arm side $\boldsymbol{\tau_a} \in \mathbf{R}^6$ are given by the dynamic model [112],

$$\boldsymbol{\tau_a} = \mathrm{M}(\boldsymbol{\phi_a})\ddot{\boldsymbol{\phi}}_\mathbf{a} + \mathrm{C}(\boldsymbol{\phi_a}, \dot{\boldsymbol{\phi}}_\mathbf{a}) + \mathrm{f_a}(\dot{\boldsymbol{\phi}}_\mathbf{a}) + \mathrm{g_a}(\boldsymbol{\phi_a}), \tag{6.1}$$

where M, C, $\mathrm{f_a}$ and $\mathrm{g_a}$ represent inertia, Coriolis and centrifugal force, friction, and the gravity torque respectively.

Including flexibility

In most cases when modeling robots, a rigid multi-body model is not sufficient to describe the system in a realistic manner. This approximation is especially unrealistic for compact gearboxes; a torsional spring damper system connecting successive arms and motors is a better representation. Given that each arm is driven by a corresponding motor $m = 1, 2, \ldots, 6$, an extension to (6.1) with elasticity in the gearbox, can be described by [110],

$$\boldsymbol{\tau_a} = \mathrm{K}(\Lambda\boldsymbol{\phi_m} - \boldsymbol{\phi_a}) + \mathrm{D}(\Lambda\dot{\boldsymbol{\phi}}_\mathbf{m} - \dot{\boldsymbol{\phi}}_\mathbf{a}), \tag{6.2a}$$

$$\boldsymbol{\tau_m} = \mathrm{M_m}\ddot{\boldsymbol{\phi}}_\mathbf{m} + \mathrm{f_m}(\dot{\boldsymbol{\phi}}_\mathbf{m}) + \Lambda\boldsymbol{\tau_a}, \tag{6.2b}$$

where K and D represent the linear elasticity and damping of the gearbox, M_m the inertia of the motor and f_m the friction torque in the motor; Λ is the gear transmission ratio, τ_a is the torque generated by the electrical motor, and ϕ_m denote the motor angles. Equations (6.1) and (6.2) form a coupled system and are solved numerically to simulate a flexible industrial manipulator.

A model for the nominal behavior of friction

Wear process of mechanical systems can be attributed to friction in components such as gearboxes and motors of robot joint [113]. A common description of a direction independent friction is based on the LuGre friction model [114, 115],

$$f_m(\dot{\phi}_m) = (f_m^c + f_m^s e^{-|\frac{\dot{\phi}_m}{\dot{\phi}_m^s}|^\gamma}) \operatorname{sign}(\dot{\phi}_m) + f_m^v \dot{\phi}_m \tag{6.3}$$

where f_m^c is Coulomb friction parameter, f_m^s is static friction parameter $\dot{\phi}_m^s$ is the Stribeck velocity, f_m^v is viscous friction parameter, $\dot{\phi}_m$ is the velocity, and γ is the exponent of the Stribeck nonlinearity.

6.2.2 Transfer entropy

Transfer entropy (TE) is an information-theoretic measure which can be used to detect the directionality of the flow of information within coupled dynamical systems [108]. Being a probabilistic measure, TE does not assume any underlying model of the dynamical relation between the coupled processes [116].

To understand transfer entropy, we first consider $X_{i,t}$ and $X_{j,t}$, $t = 1, 2, 3, \ldots$ as two discrete processes, with subscripts i and j identifying the process and t denoting time.

Transfer entropy [108] from process i to j quantifies additional bits of information from the history of process X_i that help predict the future value of X_j. Assuming a history length of m for process i and a length of l for process j, TE is defined as

$$T_{i\to j}(m,l) = \sum p(X_{j,t+1}, X_{j,t}^{(l)}, X_{i,t}^{(m)}) \times \log \frac{p\left(X_{j,t+1}|X_{j,t}^{(l)}, X_{i,t}^{(m)}\right)}{p(X_{j,t+1}|X_{j,t}^{(l)})} \tag{6.4}$$

where, $p(X_{j,t+1}, X_{j,t}^{(l)}, X_{i,t}^{(m)})$ denotes the joint probability of $X_{j,t+1}, X_{j,t}^{(l)}$ and $X_{i,t}^{(m)}$ and $p(X_{j,t+1} \mid X_{j,t}^{(l)}, X_{i,t}^{(m)})$ denotes conditional probability of $X_{j,t+1}$ on $X_{j,t}^{(l)}$ and $X_{i,t}^{(m)}$, and $p(X_{j,t+1} \mid X_{j,t}^{(l)})$ denotes probability of X_j conditioned on the past l values of X_j . History lengths l and m are set as 1 and represent first order Markov processes.

Since information transfer between two processes can occur in either direction, we compute net transfer entropy to quantify the degree of asymmetry [117]

$$netT_{i\to j} = T_{i\to j} - T_{j\to i}, \tag{6.5}$$

which quantifies the preferred direction of information flow and is expected to have a positive value when X_i is the driver process and negative otherwise.

For identifying the axis of failure in industrial manipulators, we assume that net transfer entropy from axis i to j quantifies the degree of influence. In the event of a failure, this influence is dominated by the propagation of failure and therefore the total net transfer entropy for each axis i is $\sum_{j=1,j\neq i}^{6} netT_{i\to j}$, which will serve to indicate whether that axis is the failure axis or not. Note that this definition precludes derived dependence (if i affects j through k then, we will pick up that influence between j and k as well), but assumes that upon adding all values of net transfer entropy, the maximum influence on all other axes will be from the axis of failure.

The calculation of TE depends on the availability of large amounts of data. This is because TE is typically measured by estimating the joint probability densities (equation 6.4) by using binning methods [118] or kernel density estimators [108]. Binning methods or histograms come with a free parameter—the number of bins - and the estimation of entropy-based quantities varies dramatically as the number of bins is varied [116]. Further, a binning approach is sensitive to bin width, which controls the amount of smoothing, and the start point of the first bin, which affects the shape of the histogram [118]. A reasonable mitigation for this parameter dependence and data dependence is to create bins by choosing specific quantiles of the empirical distribution of the data [118] and then recoding the data in the form of its bin location. Specifically, we create n bins, whose bounds are denoted by $q_1, q_2, ..., q_n$, such that $q_1 < q_2 < ... < q_n$; using these bins we then symbolically encode the

time series $X_{j,t}$,

$$S_{j,t} = \begin{cases} 1 & for\ X_{j,t} \leq q_1 \\ 2 & for\ q_1 < X_{j,t} \leq q_2 \\ 3 & for\ q_2 < X_{j,t} \leq q_3 \\ \vdots & \\ n-1 & for\ q_{n-2} < X_{j,t} \leq q_{n-1} \\ n & for\ X_{j,t} > q_n - 1 \end{cases}$$

where in the recoded process, each value in the observed time series $X_{j,t}$ is replaced by symbols $S_{j,t}$ according to the interval specified by the lower and upper bounds q_0 to q_n. This recoded process $S_{j,t}$ is then used to calculate TE as per equation 6.4. The choice of the number of bins should be motivated by the distribution of the data. However, Behrendt et.al. [118] recommend that the number of bins be limited in order to avoid too many zero observations for a given symbol. The use of symbolic encoding instead of samples themselves in the estimation of the transfer entropy (TE) helps to reduce the sensitivity to noise contribution which often occur in real world applications [119]. It also helps to run the algorithm faster.

6.3 Methodology

In this section, we describe simulation of the wear induced torque signal, the pre-processing of torque signals and the isolation of source axis using TE.

6.3.1 Simulating different tasks with wear

The results in this paper are validated on simulations performed using an in-house implementation of the RobotStudio software package [120], which significantly extends the model described by equations ((6.1)) and ((6.2)). As an example, the implementation includes a friction model as well as several non-actuated 6-DOF spring and damper elements distributed within the structure of manipulator. These elements introduce elasticity in the form of a flexible joint dynamic model [110]. During simulation, we consider only motor

side friction (f_m in equation (6.2)) [120] and set the arm side friction (f_a in equation (6.1)) to zero because gearbox friction constitutes the low-speed portion of the drive chain, which is typically considered to be negligible compared to that of motor side [103]. Furthermore, friction at arm side is difficult to identify in practice.

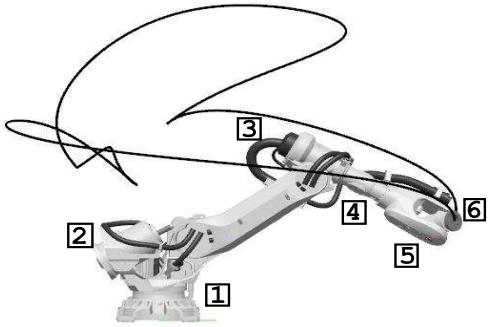
We simulated three separate tasks while introducing different degrees of wear in the axes of the robot. We designed task 1 to have maximum range of movement of each axis as per the design limits [121] and represents a pick and place operation. Task 2 represents a discontinuous process with maximum acceleration and deceleration with many stoppages similar to that of spot welding. Finally, Task 3 performs a continuous task such as arc welding or laser cutting. Geometrically, task 2 (Fig. 6-1) and task 3 are similar, but applications are different. All the three tasks represent non-contact applications where the effects of environment factors are limited. Friction coefficients include both Coulomb and static friction coefficients.

We simulate friction increase one axis at a time. The behavior of friction in a robot joint is considerably affected by variables other than wear. We ignore effects of load and temperature on friction based on earlier studies [33] that found this influence to be minimal in specific speed ranges.

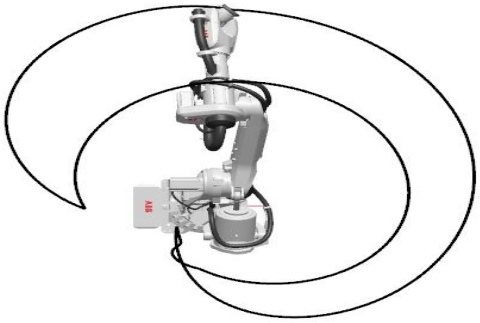
Progress of wear is typically exponential- slow in the beginning and accelerates as time progresses. We simulate wear by increasing Coulomb and static friction coefficients, f_s and f_c in equation (6.3) up to 20% over nominal values during hundred repeated cycles of a specific task. The exponent of cycle number (k) is chosen to have a steeper slope towards the end of simulation (Fig. 6-2) to reflect the exponential growth of wear. Specifically, we vary each coefficient during cycle k according to,

$$f_{m,k} = f_{m,0} \exp(c_1 k^4) \tag{6.6}$$

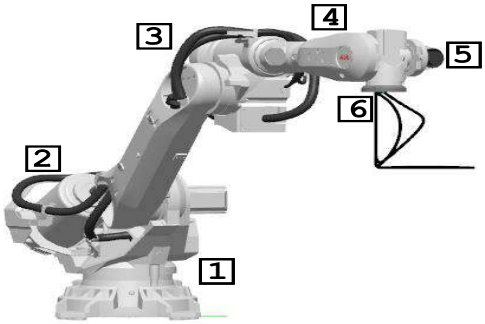
where $f_{m,0}$ is initial value of friction coefficients at cycle $k = 0$ and $c_1 = 1.785 \times 10^{-9}$ is a constant computed by solving (6.6) for desired increase in friction at the end of a hundred cycles. The form of equation 6.6 was selected on the basis of its ability to replicate an exponential rise in friction observed when manipulators are subject to wear [33]. For each k,



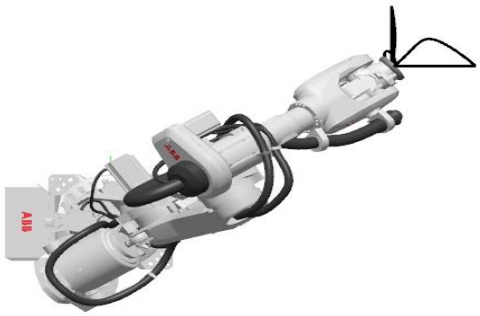
(a) Side view



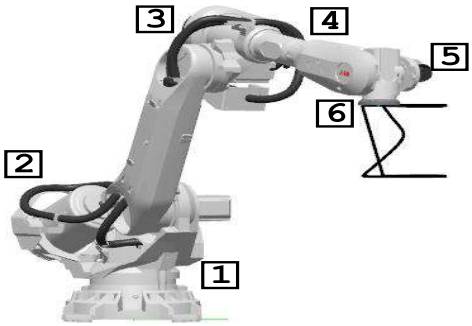
(b) Top View



(c) Side view



(d) Top View



(e) Side view

(f) Top View

Figure 6-1: End-effector trajectories for tasks: (a) & (b) for task 1 which represent pick and place operation, (c) & (d)) for task 2 which represents a discontinuous process such as spot welding, and (e) & (f) for task 3 which represents a continuous process such as arc welding or laser cutting. Top and side views of single run of the trajectories for the tasks is given. Axis numbers are shown in boxes.

the updated value of friction coefficients are substituted in equation (6.3) to arrive at friction torque f_m. Fig. 6-2 shows torque vs cycles when friction is introduced in Axis 1.

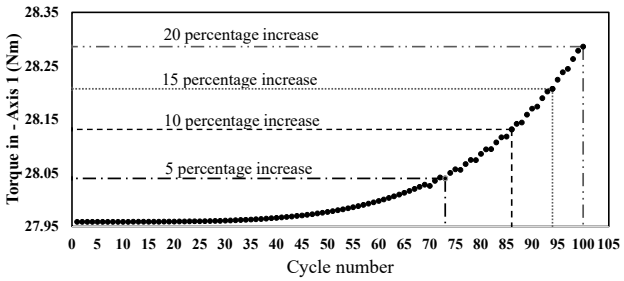


Figure 6-2: Average torque per cycle due to friction increase in Axis 1 motor of robot model when the wear increase is 20 % over the nominal value over a hundred cycles.

We performed seven simulations lasting around 60 minutes for each of the three tasks, one for no friction increase and remaining 6 runs for friction increase in each of the six axes. Fig. 6-3 shows raw torque of axis 1 from last few cycles of Task 1 with wear introduced in axis 1.

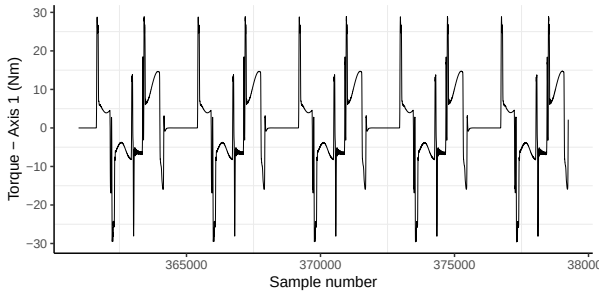


Figure 6-3: Raw torque from axis 1 for Task 1, sampled last few cycles from simulated data. Wear is introduced in axis 1.

6.3.2 Data pre-processing

The data set consists of simulated data from a six degree of freedom (DOF) industrial robot model. The simulation model is developed in-house based on theory described in section 6.2.1. The measurements included in the data are i) joint speed, and ii) joint torque, collected with sampling rate of 4 ms. We sample the torque signals associated with motor speed range between 50 rad/sec and 120 rad/sec to minimize the effects of load and temperature variation [103]. Instead of using torque signals as is, which is likely to be a function of

Table 6.1: Mean ± standard deviation of the absolute torque in Nm for each axis during the 100th sample for each simulation with 20% wear introduced in each axis, one axis at a time. Wear effects are shown for all axes for task 1, and only axis 1 for tasks 2 and 3. A '0' in Axis wear location indicates no wear.

Task	Axis / wear loc.	Torque per Axis (Nm) 1	2	3	4	5	6
1	0	**18.42**±6.85	**7.45**±4.29	**15.20**±2.39	**1.95**±2.00	**2.53**±0.97	**1.90**±1.05
1	1	**18.66**±6.38	**7.45**±4.29	**15.20**±2.39	**1.95**±2.00	**2.54**±0.97	**2.12**±1.05
1	2	**18.64**±6.38	**7.72**±4.40	**15.18**±2.36	**1.95**±2.00	**2.54**±0.97	**2.12**±1.05
1	3	**18.62**±6.36	**7.79**±4.39	**15.59**±2.50	**1.96**±2.01	**2.54**±0.97	**2.12**±1.06
1	4	**18.63**±6.37	**7.78**±4.38	**15.67**±2.51	**1.89**±2.01	**2.54**±0.97	**2.12**±1.06
1	5	**18.42**±6.85	**7.45**±4.29	**15.20**±2.39	**1.95**±1.99	**2.65**±0.96	**2.11**±1.04
1	6	**18.42**±6.85	**7.45**±4.29	**15.20**±2.40	**1.95**±1.99	**2.67**±0.96	**2.02**±1.05
2	0	**3.22**±5.10	**7.67**±7.72	**10.61**±7.12	**1.14**±1.34	**1.45**±1.34	**1.07**±1.04
2	1	**3.36**±5.20	**7.67**±7.72	**10.61**±7.12	**1.14**±1.34	**1.45**±1.34	**1.07**±1.04
3	0	**6.23**±2.29	**10.30**±4.16	**13.99**±5.60	**1.93**±1.72	**2.45**±1.01	**2.27**±0.68
3	1	**6.68**±2.34	**10.30**±4.16	**13.99**±5.60	**1.93**±1.72	**2.45**±1.01	**2.27**±0.68

different configurations within a cycle, we transform the signal to the average of absolute values per cycle resulting in 100 data points for each simulation. This also mirrors the type of data that is typically available as part of a remote monitoring solution for such robots. These values are then symbolically encoded according to the approach described in Section 6.2 and subsequently used to compute TE between pairs of robot axes.

Table 6.1 shows that if we induce 20% wear over a hundred cycles in axis 2 or axis 3, most of the axes show small changes in torque values compared with torque values associated with no wear. We attribute this phenomena to the coupling among axes, where raw measurements make it difficult to detect changes in wear. We use quantile approach discussed in Section 6.2.2 to create the bins and vary the number of bins between 5 to 20 (Fig. 6-5) for Task 1 to compute TE using RTransferentropy package [118] in R language [67]. Although the torque values are output from a deterministic process, the variability in these values (Fig. 6-3) leads to finite entropy. As a fault is introduced in the form of increased friction in one of the axes, torque, and in turn entropy, for that axis is expected to change. This change in entropy is then transferred through the coupled dynamical system of equations to other axes.

We observe that smaller number of bins are better able to detect the smaller wear changes (e.g. 1% of change in friction coefficients over a hundred cycles), where as more bins are needed for higher wear changes of more than 1%. Further, the wear percentage that leads to degradation in robot performance likely depends on the task at hand. For example, a

pressing operation may be able to tolerate higher wear, but paint application, which requires precision will have lower tolerance for wear. To assess these task dependencies we evaluate fault isolation using TE with both 5 and 10 bins. We select the number of bins to be 10, which gives us the highest accuracy in terms of fault isolation in task 1 overall.

6.4 Analysis and Results

We compute net transfer entropy between pairs of axes as per equation (6.5). We identify the source axis as the axis that has the maximum net transfer entropy. In other words, we assume that fault information encoded in the form of changes in torque is transmitted from the source axis to all other axes in the robot. Note that this inference ignores relationships that may be due to indirect couplings. For example if axis 1 is faulty then information transfer measured between axis 2 and 3 may be partly due to that coming from axis 1.

However, we assume that the information due to a fault or wear overcomes any other information and will emphasize the source axis.

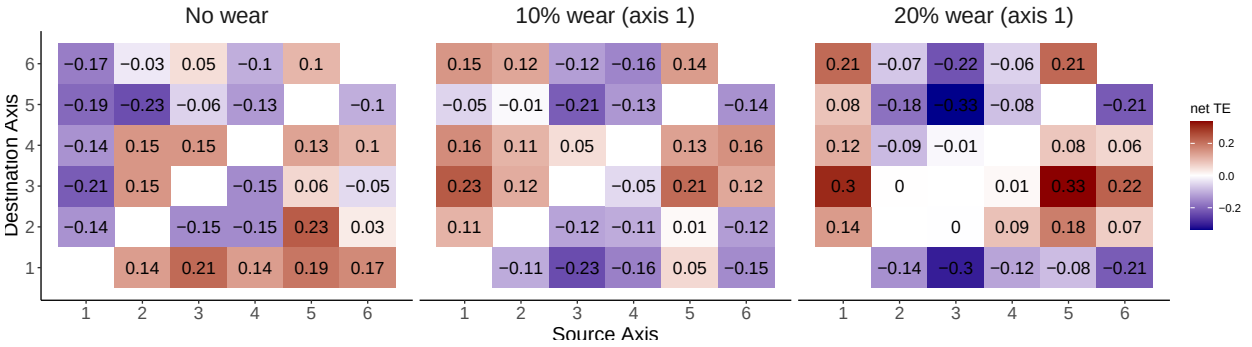


Figure 6-4: Net transfer entropy with 10 bins between pairs of axes for normal and wear induced operations in axis 1 performing task 1. Positive sign indicates information flow from source axis to destination axis. In the case where wear was induced in axis 1, the same can be inferred on the basis of all positive net transfer entropy.

We systematically evaluate the isolation rate of source axis of wear for Task 1 with different number of bins and wear amount above the nominal value. The wear amount (10% or 20%) corresponds to processing data during the 86th cycle or the 100th cycle. In this context, isolating source axis at a lower wear percentage corresponds to an early isolation of the fault. Our results indicate (Fig. 6-5) number of bins has significant influence on

isolation performance. With number of bins 20 and friction increase of $\geq 10\%$, we detect source axis of wear for all the cases. For smaller percentage increase in friction, isolation percentage is higher with lower number of bins. We find bin sizing with quantile approach as an effective tuning parameter to set the wear threshold for isolation. Fig. 6-4 shows the net transfer entropy flow between pairs of axes without friction change and with friction increase introduced in axis 1. The net transfer entropy is highest for axis 1 when we introduce 10% and 20% increase in friction in axis 1.

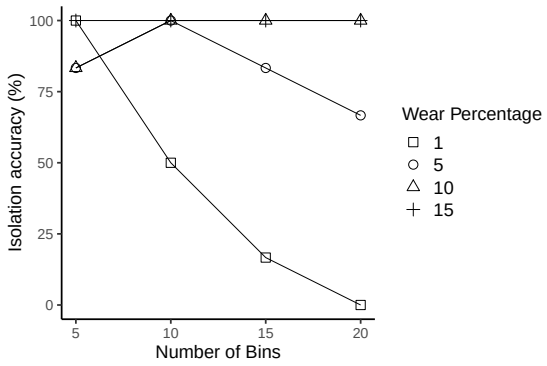


Figure 6-5: Accuracy of isolating source axis as a function of number of bins with different wear percentages in Task 1.

Table 6.2 shows the isolation performance for Tasks 1, 2, and 3 where we see that the percentage of times the source axis of wear is isolated correctly improves with increase in wear percentage. Specifically, we introduce wear in each of the six axes for the three different tasks with bin sizes 5 and 10. This resulted in six instances of failure for each task. We then calculated net transfer entropy for each axis at different times representing the different levels of wear percentages. The accuracy of isolation for a given wear percentage is calculated as the number of axes out of six identified correctly as the fault axes.

To compare these results with an existing standard approach [122], we cross-correlate the torque signal for each axis across time to identify the source axis of the wear. Specifically, we use the sign of lag at maximum cross-correlation between torque from two axes as an indicator of whether the process is leading or trailing. In this case, we expect that a leading signal will register a negative time lag. We identify the source axis as one that has all negative lags. Compared with cross-correlation, the average source axis isolation across the three tasks with TE is twice as effective at friction coefficient 10% above the nominal value

for when the number of bins is 10 and 1.4 times when number of bins is 5.

The number of bins used in calculating TE play an important role in fault isolation using this approach, with lesser bins showing better isolation rates at lower wear rate for tasks that do not require high precision (e.g. Task 1). This is likely because of high variance in data from such tasks over the same time. The approach of simulating a task for different wear percentages in RobotStudio and then using such percentages to determine the number of bins could also constitute a calibration tool for real world scenarios. Furthermore, cross correlation results vary significantly across tasks compared with isolation rate using the TE method.

Table 6.2: Comparing accuracy of isolation of source axis with wear using transfer entropy (TE) with 5 and 10 bins and cross correlation (CC). Source axis isolation markers are shown for 10 bins only. Tick mark indicates successful isolation of axis with wear and cross indicates failure in isolation.

		Source Axis of Wear								
Task	Wear Pct	1	2	3	4	5	6	Isolation accuracy (TE, 10 bins)	Isolation accuracy (TE, 5 bins)	Isolation accuracy (CC)
T1	1	✓	×	✓	×	×	✓	50	100	16.7
T1	5	✓	✓	✓	✓	✓	✓	100	83.3	33.3
T1	10	✓	✓	✓	✓	✓	✓	100	83.3	33.3
T1	15	✓	✓	✓	✓	✓	✓	100	100	33.3
T2	1	✓	✓	×	✓	✓	✓	83.3	66.7	16.7
T2	5	✓	✓	✓	✓	×	✓	83.3	66.7	0
T2	10	✓	✓	✓	✓	✓	✓	100	66.7	33.3
T2	15	✓	✓	✓	✓	✓	✓	100	50	33.3
T3	1	✓	✓	✓	✓	✓	✓	100	83.3	50
T3	5	✓	✓	✓	✓	✓	✓	100	66.7	66.7
T3	10	✓	✓	✓	✓	✓	✓	100	66.7	83.3
T3	15	✓	✓	✓	✓	✓	✓	100	66.7	100

6.5 Conclusion

The problem of identifying causal relationships within multiple processes is prevalent across natural and engineered systems. In this paper, we evaluate an information-theoretic approach of transfer entropy to identify the source axis of wear in industrial robots. Complex systems such as industrial robots, where signals from each axis interact in a highly nonlinear fashion make it impractical to use methods such as cross-correlation that assume a linear relationship between processes. Here, we sought to test TE as an approach to

isolate faults in industrial robots because of its demonstrated effectiveness in identifying directional relationships in highly non-linear interactive subsystems [106].

We provide evidence of TE's effectiveness as an approach for fault isolation due to wear in Industrial robots. Wear is simulated on a prototype robot model, similar to the ABB IRB 6700, by increasing friction coefficients in the model using an ABB simulator. We simulated data for three common industrial tasks: pick and place, spot welding, and arc welding. The axis responsible for wear was always detected when the wear was 10% above nominal value irrespective of the location of wear on the robot. Compared to cross correlation a TE based approach is able to detect source axis of wear more accurately and consistently in our setup. When used in conjunction with fault detection methods, the TE approach presented here has the potential to be used to develop a full fault detection and isolation solution. In particular, once a fault is detected using model or data-driven methods, the same data can then be further processed to isolate the location of the fault.

Future studies will focus on online estimation of wear towards an adaptive binning approach, and evaluating this approach on real data from the field as well as on faults from other causes including electrical failures of motors.

Part IV

Conclusion

Conclusion 7

Conclusion

In this chapter, results presented in this thesis are summarized and possible directions for future research are suggested.

7.1 Summary

Good design is essential for high reliability of industrial robots. However, no amount of design effort can prevent deterioration over time. The development of failures and wear is dependent on the accumulated usage of the robot. Unexpected breakdown of robots can lead to loss of production time and result in considerable financial losses. Therefore, implementing a condition monitoring system can ensure a high level of equipment availability, and lower the spares inventory cost. Model based methods are used for condition monitoring of industrial robots. However, perfect knowledge of the manipulator model is rarely a reasonable assumption, especially progression of wear is difficult to model. In such contexts, data-driven methods are widely used to detect faults in industrial equipment which are subjected to disturbance and uncertain environment. Although wear induced faults develop slowly and are tolerable initially, it can cause very serious problems at later stages. This study is mainly motivated to detect, isolate such faults as early as possible, and take planned actions to maintain the reliability of a system. In this dissertation, the use of data-driven methods is investigated to detect and isolate wear induced faults in industrial robots using data typically collected through remote monitoring solution.

Simulated data and real data are used to develop fault detection models. Both data and algorithms influence the performance of data-driven methods. The factors that influence the robot data and associated information content include robot application, tasks and environment. Algorithms differ in terms of the extent to which they rely on an underlying model, and how they weight bias versus variance within data to isolate anomalous situations. With a remote monitoring solution, knowledge about all the factors may not be available and thus create some of amount of uncertainty about the data generation process. Therefore, a two-pronged approach is adopted for the study: a) evaluate data-driven methods on simulated data, b) apply the algorithms that worked well with simulated data on real data enhanced with preprocessing methods. In addition, simulation studies are helpful in conducting controlled experiments where one can vary key parameters of interest while keeping other factors constant. Again, studying wear induced faults with physical experiments will take more resources in terms of cost and time.

First research question (Q1), "What is the effectiveness of Principal Component Analysis (PCA) in terms of detecting the onset of wear induced failures in industrial robots?" is addressed by applying PCA algorithm on real data sets and simulated data sets. PCA is applied on simulated as well as read data to evaluate its effectiveness in detecting faults(C1, C2 and C3). The second question (Q2), "Can the effectiveness of failure identification using standard classification methods be improved by curating the data using pre-processing methods?" is addressed by curating the data with different pre-processing methods before developing data-driven models (C3 and Journal paper J1). Initial research work (C1, C2 and C3) is based on limited data sets. Further the effectiveness of pre-processing is evaluated by analyzing the data from more failures cases (J1). The third question (Q3), "Can correlation and information theoretic methods be used to isolate the source axis of the failure?", is addressed by evaluating the effectiveness of cross correlation (C2) and information theoretic approach (J2) on isolating the fault in industrial robots.

Data such as event logs and measurements from robot controllers are collected using a remote data collection solution. The investigation was started by analyzing event logs from robots to detect wear related failures. Initially, Principal Component Analysis (PCA) based approach was adapted on event logs to detect a gearbox fault. The proposed method reduces

the dimensionality of the original data which consist of correlated events while retaining the variation present in the data. An abnormal pattern is observed within 30 days before failure by using multivariate statistics: Hotelling T^2, Q Residuals and Q contribution chart. However, the approach could not be generalized using event logs as there exist variations in configured events across robots. During the initial phase of the study, there were not many field failure cases available for the investigation. Therefore, a simulation based approach was explored in parallel. A method is proposed to detect the onset of wear and subsequently locate the source axis of wear using the simulated data. Wear was simulated by increasing friction coefficients of different axes in a PUMA 560 robot model available in MATLAB Robotics toolbox on three separate tasks. Load disturbances and measurement noise are added in the simulation to evaluate the robustness of the wear detection. Wear is detected using principal component analysis, and the source axis is located by cross-correlating the output data from all axes through time. The investigation further focused on the multiple failure cases observed in the field. The effect of source and type of training data on detecting field failures in industrial robots are studied with Principal Component Analysis (PCA). Specifically, using field data across multiple robots performing different tasks, two scenarios are compared: first, where training data obtained from a single robot is used to evaluate multiple robots (one-to-many), and second, where each robot is evaluated on the basis of its own training data (one-to-one). The effects of data preprocessing prior to running PCA on ability to detect and predict failures are investigated. The preprocessing of signal has found to be significant influence on detecting failures.

As data were gathered for many gearbox failures cases, the research was narrowed down on fault detection and isolation (FDI) of gearbox faults. First, the occurrence of fault was detected using supervised learning algorithms in a systematic hypothesis-driven study. In this work, preprocessing methods are found useful in improving the performance of simple classification methods such as Logistic regression (LR). Four common preprocessing techniques are used on data collected from twenty-six industrial robots. Hypotheses related to the effect of combining four preprocessing techniques are tested on classification performance. Combinations of preprocessing techniques include absolute difference of the raw timeseries data (+diff), inclusion of estimated torque and

speed (+estM), preprocessing of data sets with Principal Component Analysis (+pca) to reduce the dimensionality, and inclusion of local variation of # days (var#) in the data set. Specifically, differencing reduced variance in area under curve (AUC) across all classification methods, inclusion of local variation in conjunction with other preprocessing techniques improved classification performance, among three out of four classification methods (except Logistics Regression(LR)), inclusion of estimated measures along with other preprocessing methods showed significant increase in performance. In addition, classification performance of LR method improved significantly with PCA preprocessing. Maximum improvement in mean value of AUC across classification methods is between no processing and +diff+estM+pca+var3. LR is the most receptive classification method whose improvement with data preprocessing is comparable with complex classifiers such as Random Forest (RF) and Support Vector Machine (SVM) while executing in a fraction of the time.

Second, the problem of isolating the axis of fault is investigated. This problem is addressed by inferring pairwise causal relations between all axes using an information theoretic approach called Transfer Entropy (TE). In industrial robots, the axes are dynamically and tightly coupled. The degree of influence between axes are varying across axis. To isolate the faulty axis from tightly coupled axes, data is simulated using ABB in-house simulations software that include the industrial control system and realistic manipulator simulation models and transfer entropy approach that helps to isolate the fault is evaluated. Simulation data is used instead of real robot data to avoid confounding effect of other factors on performance of the approach. The axis of fault is located by finding the axis which has the highest net transfer entropy across all pairs of axes. To generate representative data sets, three common application tasks are simulated: pick and place, spot welding and arc welding. Wear is simulated by increasing Coulomb and static friction coefficients in the simulation model. The axis responsible for wear is always detected when the wear is 10% above nominal value irrespective of the location on the robot. The results of TE are compared to that of cross correlation and the performance of TE in detecting source axis is found to be higher and consistent among the tasks.

Analysis of data collected using a state of the art remote monitoring system at the time

of its first release, provided very interesting results in detecting wear induced failures of mechanical arms of industrial robots. However, more real-time data are required to improve the fault detection performance further. New developments in the area of IoT based solutions are happening with higher pace than ever. In this context, our methods and results can be considered as input for the design of next generation condition monitoring systems.

7.2 Future work

In this section, future research directions are identified and grouped into methods and algorithms, data collection, and faults other than those from wear.

7.2.1 Methods and algorithms

The methods could be fine tuned to detect the right amount of wear that can impair the functionalities of the gearbox and give enough time to repair or replace the parts. Another direction is to investigate the association between the length of the data before failure and the detection lag. In addition, improvement in fault detection performance can be evaluated by adopting methods to segregate influence of external disturbances -e.g. contact applications where the effect of disturbances exceed the effect of friction change. It is observed that while using transfer entropy approach for fault isolation, number of bins is the major influencing factor. Therefore, correlation between number of bins and fault isolation properties can be further investigated.

ML algorithms have great potential in detecting faults in industrial robots and can be of great help to service engineers. However, the decisions/predictions made by these algorithms usually do not explain the reasoning which is barrier in adopting such technology in performing service activities. Therefore, explainable ML algorithms for fault detection can be investigated.

Further, possibility to deploy the algorithms in the control system and use of statistical control charts for near real time fault detection can be explored.

7.2.2 Data collection

With respect to data collection, the approaches developed here could be validated on larger data sets collected from the field. This will help us to address all of the nuances that occur in real production environment, such as disturbances and unexpected signals from external devices. In current monitoring system, collection of measurements related to health condition/state of the robot is more robust than collection of service information on reported problem which includes causes and resolution on target robot. Therefore, the improved collection of service information can enhance the information content of the data to detect faults.

Further, analyzing data from component level sensors and correlating with failures will be a natural step in the next generation of FDI scheme. The new generation robots are equipped with additional sensors and we need to continuously evaluate the approaches using the newly acquired data. Wear and tear of robots can lead to significant vibration of physical structure of robot. Therefore, possibility of using vibration signal for condition monitoring of robots can be explored. In addition, faults in multiple axes can be simulated and resulting data can be used for fault detection and isolation.

7.2.3 Faults other than those from wear

The investigation can be extended on robot faults other than wear induced gearbox faults: motor faults, faults related to cabling, and drive faults. Faults are structurally different from one another can be modeled and combined models can provide a better fault detection performance. Therefore, the use of multiple-model FDI (MM–FDI) approach can be investigated.

Customers are interested in the reliability of the robot application process along with the reliability of the robots and detection of faults in robot application can further investigated. Application processes that are controlled by robots include, but are not limited to welding, gluing, painting, and picking.

In addition, the effectiveness of the methods in detecting faults in robot software system can be investigated, for example, detect memory leakages where resources are consumed in

a way that lead to a system failure.

Finally, simulation models can be further improved by adding new parameters to simulate wear in components within gearbox such as bearings. In addition, backlash fault in gearbox can be simulated by changing stiffness of axes in models such as those within the existing ABB in-house simulation platform.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Performance evaluation of preprocessing techniques

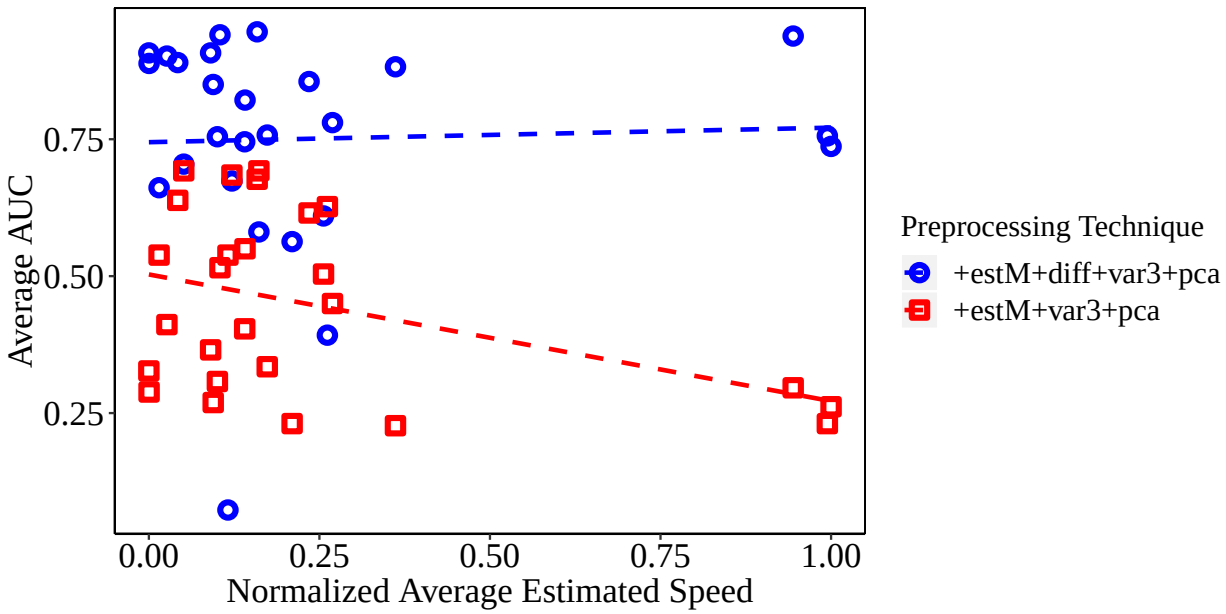


Figure A.1: Classification performance of LR as mean AUC (with and without differencing) vs normalized average estimated speed.

Table A.1: Execution time in minutes. Since ES is a meta classifier that uses the output of LR and SVM, the execution time for ES is simply the sum of LR and SVM. The execution time comprises of the time required for executing the procedure described in Procedure I. Execution time reduces significantly across all classifiers with +pca.

Preprocessing techniques	LR	SVM	RF
control	13	92	120
+diff	12	90	120
+diff+pca	3	49	70
+estM+diff+var3	17	143	213
+estM+diff+var3+pca	4	76	78

Table A.2: Effect of preprocessing technique on variance of AUC: Value in bold represent the significant reduction in variance ($p < 0.05$).

Without +diff	**With +diff**	**LR**	**SVM**	**RF**	**ES**
Control	+diff	**0.00**	**0.00**	**0.01**	**0.00**
+pca	+diff+pca	0.44	**0.00**	**0.00**	**0.00**
+estM	+estM+diff	**0.02**	**0.00**	**0.01**	**0.00**
+var3+pca	+diff+var3+pca	0.97	**0.00**	0.10	**0.00**
+estM+pca	+estM+diff+pca	**0.00**	**0.00**	**0.00**	**0.00**
+estM+var3+pca	+estM+diff+var3+pca	0.83	**0.00**	**0.00**	**0.00**
Without +var3	**With +var3**	**LR**	**SVM**	**RF**	**ES**
control	+var3	1.00	0.26	0.96	0.05
+pca	+var3+pca	1.00	1.00	0.71	0.71
+estM	+estM+var3	1.00	0.87	0.93	0.00
+diff+pca	+diff+var3+pca	0.33	0.42	0.00	0.27
+estM+pca	+estM+var3+pca	**0.00**	0.03	0.87	0.00
+estM+diff+pca	+estM+diff+var3+pca	0.59	0.38	0.65	0.44
Without +var3	**With +var3**	**LR**	**SVM**	**RF**	**ES**
control	+var3	1.00	0.26	0.96	0.05
+pca	+var3+pca	1.00	1.00	0.71	0.71
+estM	+estM+var3	1.00	0.87	0.93	0.00
+diff+pca	+diff+var3+pca	0.33	0.42	0.00	0.27
+estM+pca	+estM+var3+pca	**0.00**	0.03	0.87	0.00
+estM+diff+pca	+estM+diff+var3+pca	0.59	0.38	0.65	0.44
Without +pca	**With +pca**	**LR**	**SVM**	**RF**	**ES**
control	+pca	0.03	0.05	0.58	0.10
+diff	+diff+pca	0.00	0.00	0.62	0.04
+var3	+var3+pca	0.03	0.42	0.89	1.00
+estM	+estM+pca	0.00	0.15	0.07	0.72
+estM+diff	+estM+diff+pca	0.00	0.49	0.36	0.15
+estM+var3	+estM+var3+pca	1.00	1.00	0.076	1.00

Table A.3: Effect of preprocessing technique on mean value of AUC: Value in bold represent increase in mean value ($p < 0.05$).

Without +diff	**With +diff**	**LR**	**SVM**	**RF**	**ES**
Control	+diff	0.99	0.15	0.99	0.01
+pca	+diff+pca	0.14	0.06	0.44	**0.05**
+estM	+estM+diff	0.25	1.00	0.69	1.00
+var3+pca	+diff+var3+pca	1.00	0.42	1.00	0.17
+estM+pca	+estM+diff+pca	**0.00**	1.00	0.852	1.00
+estM+var3+pca	+estM+diff+var3+pca	**0.00**	**0.00**	**0.00**	**0.00**
control	+var3	1.00	1.00	0.99	0.25
+pca	+var3+pca	1.00	1.00	1.00	1.00
+estM	+estM+var3	1.00	1.00	1.00	0.95
+diff+pca	+diff+var3+pca	0.97	1.00	1.00	1.00
+estM+pca	+estM+var3+pca	0.99	0.61	1.00	0.05
+estM+diff+pca	+estM+diff+var3+pca	1.00	**0.01**	0.24	**0.00**
Without +estM	**With +estM**	**LR**	**SVM**	**RF**	**ES**
control	+estM	1.00	0.15	1.00	0.82
+diff	+estM+diff	1.00	**0.00**	0.67	**0.03**
+pca	+estM+pca	0.00	0.08	1.00	**0.00**
+var3	+estM+var3	1.00	1.00	1.00	1.00
+diff+pca	+estM+diff+pca	0.08	0.97	0.88	0.99
+var3+pca	+estM+var3+pca	0.00	0.99	1.00	1.00
+diff+var3+pca	+estM+diff+var3+pca	0.66	**0.00**	**0.01**	**0.01**
Without +pca	**With +pca**	**LR**	**SVM**	**RF**	**ES**
control	+pca	**0.00**	0.99	0.26	1.00
+diff	+diff+pca	0.14	0.06	0.44	**0.05**
+var3	+var3+pca	**0.00**	0.99	1.00	0.99
+estM	+estM+pca	0.99	0.98	0.93	0.75
+estM+diff	+estM+diff+pca	**0.00**	0.99	0.19	0.99
+estM+var3	+estM+var3+pca	1.00	1.00	0.926	1.00

Bibliography

[1] André Carvalho Bittencourt, Kari Saarinen, Shiva Sander Tavallaey, et al. A data-driven method for monitoring systems that operate repetitively–applications to wear monitoring in an industrial robot joint. In *Fault Detection, Supervision and Safety of Technical Processes*, volume 8, pages 198–203, 2012. 3, 33, 44

[2] André Carvalho Bittencourt. *On Modeling and Diagnosis of Friction and Wear in Industrial Robots*. PhD thesis, Linkoping University, Linkoping, January 2012. 4, 26, 39, 42

[3] André Carvalho Bittencourt and Patrik Axelsson. Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data. *IEEE/ASME Transactions on Mechatronics*, 19(5):1694–1706, 2013. 4, 26

[4] Dominique Blanc. Service from afar ABB's remote service concept is revolutionizing the robotics industry. In *ABB Review 1*, 2009. 4, 42

[5] Mark W Spong and Mathukumalli Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, 2008. 5

[6] Attila L Bencsik. Twin diagnostic method for industrial robot maintenance. In *IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, pages 87–89. IEEE, 2012. 5, 6

[7] A Yamada and S Takata. Reliability improvement of industrial robots by optimizing

operation plans based on deterioration evaluation. *CIRP Annals-Manufacturing Technology*, 51(1):319–322, 2002. 6

[8] Eigenvector Research. *T-Squared Q residuals and Contributions*. http://wiki.eigenvector.com/index.php?title=T-Squared_Q_residuals_and_Contributions. Accessed: 2014-06-25. 9, 21

[9] Chee Khiang Pang, Frank L Lewis, Tong Heng Lee, and Zhao Yang Dong. *Intelligent diagnosis and prognosis of industrial networked systems*, volume 44. CRC Press, 2011. 14, 15

[10] George Vachtsevanos, Frank Lewis, Michael Roemer, Andrew Hess, and Biqing Wu. Fault diagnosis and prognosis performance metrics. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, pages 355–399. 14, 27

[11] Andrew KS Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510, 2006. 14

[12] Michael J Roemer, Carl S Byington, Gregory J Kacprzynski, and George Vachtsevanos. An overview of selected prognostic technologies with application to engine health management. In *Turbo Expo: Power for Land, Sea, and Air*, volume 42371, pages 707–715, 2006. 14

[13] Lin Luhui and Ma Jie. Fault prediction based on data-driven technique. In *Proceedings of the 2010 International Conference on Electrical and Control Engineering*, pages 997–1001. IEEE Computer Society, 2010. 15, 42, 44

[14] M Ahmed, M Baqqar, F Gu, and AD Ball. Fault detection and diagnosis using principal component analysis of vibration data from a reciprocating compressor. In *Proceedings of 2012 International Conference on Control (CONTROL), UKACC*, pages 461–466, 2012. 15, 18, 43, 45

[15] Shen Yin, Steven X Ding, Adel Haghani, Haiyang Hao, and Ping Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the

benchmark tennessee eastman process. *Journal of process control*, 22(9):1567–1581, 2012. 15, 18, 19

[16] Q Chen, U Kruger, M Meronk, and AYT Leung. Synthesis of T^2 and Q statistics for process monitoring. *Control Engineering Practice*, 12(6):745–755, 2004. 15, 18, 29, 44

[17] Luhui Lin, Jie Ma, Xiulan Ye, and Xiaoli Xu. Mechanical fault prediction based on principal component analysis. In *Proceedings of the IEEE International Conference on Information and Automation (ICIA)*, pages 2258–2262, 2010. 15, 18, 42, 43, 45

[18] Jinyu Guo, Yuan Li, Guozhu Wang, and Jing Zeng. Batch process monitoring based on multilinear principal component analysis. In *2010 International Conference on Intelligent System Design and Engineering Application*, volume 1, pages 413–416. IEEE, 2010. 15

[19] Dan Lawesson, Ulf Nilsson, and Inger Klein. Fault isolation in discrete event systems by observational abstraction. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 5, pages 5118–5123. IEEE, 2003. 15

[20] ABB Robotics. Operating manual - trouble shooting, irc5, robot controller,irc5. *Västerås, Sweden*, 2004. 16

[21] Michael Steinbach Pan-Ning Tan, Vipin Kumar. *Introduction to Data Mining*. Pearson, 2013. ISBN 978-81-317-1472-0. 17, 28, 44

[22] Manish Misra, H Henry Yue, S Joe Qin, and Cheng Ling. Multivariate process monitoring and fault diagnosis by multi-scale pca. *Computers & Chemical Engineering*, 26(9):1281–1293, 2002. 17, 19

[23] Jiayu Chen, Dong Zhou, Chuan Lyu, and Chen Lu. Feature reconstruction based on t-sne: an approach for fault diagnosis of rotating machinery. *Journal of Vibroengineering*, 19(7):5047–5060, 2017. 23

[24] Mehrdad R Kermani, Mathew Wong, Rajnikant V Patel, Mehrdad Moallem, and Mile Ostojic. Friction compensation in low and high-reversal-velocity manipulators.

In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4320–4325. IEEE, 2004. 26

[25] Basilio Bona and Marina Indri. Friction compensation in robotics: an overview. In *Proceeding of the 44th IEEE Conference on Decision and Control, and European Control Conference*, pages 4360–4367, 2005. 26, 39

[26] André Carvalho Bittencourt. Friction change detection in industrial robot arms. *Master's Degree Project, KTH electrical engineering, Stockholm*, 2007. 26, 27, 39

[27] Koji Kato. Wear in relation to friction—a review. *Wear*, 241(2):151–157, 2000. 26

[28] André Carvalho Bittencourt, Erik Wernholt, Shiva Sander-Tavallaey, and Torgny Brogardh. An extended friction model to capture load and temperature effects in robot joints. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6161–6167, 2010. 26

[29] Pauline Hamon, Maxime Gautier, and Philippe Garrec. Dynamic identification of robots with a dry friction model depending on load and velocity. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6187–6193, 2010. 26

[30] Ola Pettersson. Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, 53(2):73–88, November 2005. ISSN 09218890. doi: 10.1016/j.robot.2005.09.004. 26, 27, 28, 43, 44, 82

[31] Leo H Chiang, Richard D Braatz, and Evan L Russell. *Fault detection and diagnosis in industrial systems*. Springer, 2001. 26

[32] Michael L McIntyre, Warren E Dixon, Darren M Dawson, and Ian D Walker. Fault detection and identification for robot manipulators. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4981–4986. IEEE, 2004. 26

[33] André Carvalho Bittencourt and Svante Gunnarsson. Static friction in a robot joint—modeling and identification of load and temperature effects. *Journal of Dynamic Systems, Measurement, and Control*, 134(5):051013, 2012. 26, 83, 88

[34] Lorinc Márton. Energetic approach to deal with faults in robot actuators. In *Proceeding of the 20th Mediterranean Conference on & Automation (MED)*, pages 85–90. IEEE, 2012. 26

[35] ABB Robotics. Product range: Improving productivity, quality and workplace safety. *ABB Robotics*, 2010. 27

[36] André Carvalho Bittencourt, Kari Saarinen, Shiva Sander-Tavallaey, Svante Gunnarsson, and Mikael Norrlöf. A data-driven approach to diagnostics of repetitive processes in the distribution domain–applications to gearbox diagnostics in industrial robots and rotating machines. *Mechatronics*, 24(8):1032–1041, 2014. 27, 31, 33, 42

[37] Riccardo Muradore and Paolo Fiorini. A pls-based statistical approach for fault detection and isolation of robotic manipulators. *IEEE Transactions on Industrial Electronics*, 59(8):3167–3175, 2012. 27, 38

[38] Yier Wu, Alexandr Klimchik, Stéphane Caro, Benoit Furet, and Anatol Pashkevich. Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments. *Robotics and Computer-Integrated Manufacturing*, 35:151–168, 2015. 27

[39] Peter I Corke and Oussama Khatib. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011. 27, 30, 31, 32

[40] Junho Choi, Sungchul Kang, et al. External force estimation using joint torque sensors for a robot manipulator. In *2012 IEEE International Conference on Robotics and Automation*, pages 4507–4512. IEEE, 2012. 28

[41] Rolf Johansson, Anders Robertsson, Klas Nilsson, and Michel Verhaegen. State-space system identification of robot manipulator dynamics. *Mechatronics*, 10(3):403–418, 2000. 28

[42] Deng Pan, Zhiliang Liu, Longlong Zhang, Yinjiang Liu, and Ming J Zuo. A study on applications of principal component analysis and kernel principal component analysis for gearbox fault diagnosis. In *Proceedings of the International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, pages 1917–1922. IEEE, 2013. 28, 43

[43] JF MacGregor and Th Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995. 29

[44] Rosani ML Penha and J Wesley Hines. Using principal component analysis modeling to monitor temperature sensors in a nuclear research reactor. In *Proceedings of the maintenance and reliability conference (MARCON)*, pages 6–9. Citeseer, 2001. 29, 43, 44

[45] V Sathish, S D Sudarsan, and Ramaswamy Srini. Event based robot prognostics using principal component analysis. In *Proceedings of IEEE International Symposium on Software Reliability Engineering (ISSRE)*, pages 14–17. IEEE, 2014. 29

[46] Jay Devore. *Probability and Statistics for Engineering and the Sciences*. Thomson Learning, 2002. 29, 44, 47

[47] Chris Chatfield. *The Analysis of Time Series, An Introduction*. Chapman and Hall, 2012. ISBN 1-58488-317-0. 29

[48] Peter I Corke et al. *Visual Control of Robots: high-performance visual servoing*. Research Studies Press Taunton, UK, 1996. 30

[49] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003. 31

[50] Benjamin W Mooring, Zvi S Roth, and Morris R Driels. *Fundamentals of manipulator calibration*. John Wiley & Sons, NY, 1991. 32

[51] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. An information-theoretic approach to detecting changes in multi-dimensional data

streams. In *Proceedings on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2006. 33, 46

[52] Warren E Dixon, Ian D Walker, Darren M Dawson, and John P Hartranft. Fault detection for robot manipulators with parametric uncertainty: a prediction-error-based approach. *IEEE Transactions on Robotics and Automation*, 16(6):689–699, 2000. 38

[53] M Samhouri, A Al-Ghandoor, S Alhaj Ali, I Hinti, and W Massad. An intelligent machine condition monitoring system using time-based analysis: neuro-fuzzy versus neural network. *Jordan Journal of Mechanical and Industrial Engineering*, 3(4):294–305, 2009. 42

[54] Aveek Datta, Constantinos Mavroidis, Jay Krishnasamy, and Martin Hosek. Neural network based fault diagnostics of industrial robots using wavelet multi-resolution analysis. In *Proceedings of the IEEE American Control Conference*, pages 1858–1863, 2007. 42

[55] Mark Schwabacher and Kai Goebel. A survey of artificial intelligence for prognostics. In *AAAI fall symposium*, pages 107–114, 2007. 42

[56] Satnam Singh, Clifton Pinion, and Halasya Siva Subramania. Data-driven framework for detecting anomalies in field failure data. In *Aerospace Conference*, pages 1–14. IEEE, 2011. 42

[57] Tom Howley, Michael G Madden, Marie-Louise O'Connell, and Alan G Ryder. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge-Based Systems*, 19(5):363–370, 2006. 43

[58] B Mnassri, EM El Adel, and M Ouladsine. Fault localization using principal component analysis based on a new contribution to the squared prediction error. In *Proceedings of IEEE Mediterranean Conference on Control and Automation,*, pages 65–70, 2008. 43

[59] Barry M Wise and Neal B Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

[60] Jianbo Yu. Fault detection using principal components-based gaussian mixture model for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 24(3):432–444, 2011. 43, 44

[61] Cesare Fantuzzi, Cristian Secchi, and Antonio Visioli. On the fault detection and isolation of industrial robot manipulators. In *7th International IFAC Symposium on Robot Control*, 2003. 44

[62] Geof Givens, J Ross Beveridge, Bruce A Draper, and David Bolme. A statistical assessment of subject factors in the PCA recognition of human faces. In *Proceedings of Computer Vision and Pattern Recognition Workshop (CVPRW)*, volume 8, pages 96–96. IEEE, 2003. 44

[63] James Schimert. Data-driven fault detection based on process monitoring using dimension reduction techniques. In *Aerospace Conference*, pages 1–12. IEEE, 2008. 44

[64] Weihua Li, H Henry Yue, Sergio Valle-Cervantes, and S Joe Qin. Recursive PCA for adaptive process monitoring. *Journal of process control*, 10(5):471–486, 2000. 45

[65] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 46

[66] Douglas C Montgomery and George C Runger. *Applied statistics and probability for engineers*. John Wiley & Sons, 2010. 49

[67] R Core Team. R: A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria. 2013, 2014. 49, 66, 73, 74, 91

[68] Joseph F Hair. Multivariate data analysis. 2014. 53

[69] Peter K Dunn and Gordon K Smyth. *Generalized linear models with examples in R*. Springer, 2018. 53

[70] Mohd Amin Abd Majid and Fauzi Fudzin. Study on robots failures in automotive painting line. *Asian Research Publishing Network (ARPN)*, 12(1):62–67, 2017. 62

[71] Tawfik Borgi, Adel Hidri, Benjamin Neef, and Mohamed Saber Naceur. Data analytics for predictive maintenance of industrial robots. In *International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2017*, pages 412–417. IEEE, 2017. 62, 63

[72] Cesare Fantuzzi, Cristian Secchi, and Antonio Visioli. On the fault detection and isolation of industrial robot manipulators. In *7th International IFAC Symposium on Robot Control*, pages 399–404, 2003:. 62

[73] Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel. Prognostics and health management design for rotary machinery systems: Reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1-2):314–334, 2014. 62, 69

[74] A G Starr, R J Wynne, and I Kennedy. Failure analysis of mature robots in automated production. *Proceedings of the Institution of Mechanical Engineers,Part B: Journal of Engineering Manufacture*, 213(8):813–824, 1999. 62

[75] Zhiwei Gao, Carlo Cecati, and Steven X Ding. A survey of fault diagnosis and fault-tolerant techniques —Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767, 2015. 62, 82

[76] Ola Pettersson. Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, 53(2):73–88, November 2005. 63, 64, 72

[77] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1-3):191–202, January 2002. 63

[78] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48, 2010. 63

[79] V Sathish, Srini Ramaswamy, and Sachit Butail. Training data selection criteria for detecting failures in industrial robots. In *IFAC International Conference on*

Advances in Control and Optimization Of Dynamical Systems, volume 49, pages 385–390, January 2016. 63

[80] Marten Scheffer, Stephen R Carpenter, Timothy M Lenton, Jordi Bascompte, William Brock, Vasilis Dakos, Johan van de Koppel, Ingrid A van de Leemput, Simon A Levin, Egbert H van Nes, Mercedes Pascual, and John Vandermeer. Anticipating critical transitions. *Science*, 338(6105):344–348, October 2012. 63, 71

[81] J Friedman, T Hastie, and R Tibshirani. *The elements of statistical learning*. Springer Series in Statistics. Springer New York Inc., 2009. 64, 66, 67, 68

[82] M Kuhn and K Johnson. *Applied predictive modeling*. Springer, 2013. 64, 65, 66, 67

[83] Han Liu, Jianzhong Zhou, Yang Zheng, Wei Jiang, and Yuncheng Zhang. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Transaction*, 2018. doi: 10.1016/j.isatra.2018.04.005. 64

[84] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning (adaptive computation and machine learning series). *Adaptive Computation and Machine Learning series*, page 800, 2016. 65

[85] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. 67

[86] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007. 67

[87] M Paz Sesmero, Agapito I Ledezma, and Araceli Sanchis. Generating ensembles of heterogeneous classifiers using stacked generalization. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 5(1):21–34, 2015. 67

[88] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, November 2012. 70

[89] R Penha and J W Hines. Using principal component analysis modeling to monitor temperature sensors in a nuclear research reactor. In *Proceedings of the 2001*

Maintenance and Reliability Conference (MARCON 2001), Knoxville, TN. Citeseer, 2001. 72

[90] Thavatchai Vorapongsathorn, Sineenart Taejaroenkul, and Chukiat Viwatwongkasem. A comparison of type i error and power of bartlett's test, levene's test and cochran's test under violation of assumptions. *Songklanakarin J. Sci. Technol*, 26(4):537–547, 2004. 74

[91] K Moder. Alternatives to F-test in one way ANOVA in case of heterogeneity of variances (a simulation study). *Psychological Test and Assessment Modeling*, 52(4): 343–353, 2010. 74

[92] Graeme D. Ruxton and Guy Beauchamp. Time for some a priori thinking about post hoc testing. *Behavioral Ecology*, 19(3):690–693, 2008. 75

[93] Douglas C Montgomery and George C Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, March 2010. 75

[94] John Fox and Sanford Weisberg. *An R Companion to Applied Regression*. SAGE, 2011. 74

[95] Wallace Hui, Yulia Gel, and Joseph Gastwirth. lawstat: An R package for law, public policy and biostatistics. *Journal of Statistical Software, Articles*, 28(3):1–26, 2008. ISSN 1548-7660. 74

[96] Gjalt-Jorn Ygram Peters. userfriendlyscience: Quantitative analysis made accessible. 2017. R package version 0.7.0. 75

[97] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 80

[98] Sathish Vallachira, Michal Orkisz, Mikael Norrlöf, and Sachit Butail. Data-driven gearbox failure detection in industrial robots. *IEEE Transactions on Industrial Informatics*, 16(1):193–201, 2020. 82

[99] Ahmad H Sabry, Farah Hani Nordin, Ameer H Sabry, and Mohd Zainal Abidin Ab Kadir. Fault detection and diagnosis of industrial robot based on power consumption modeling. *IEEE Transactions on Industrial Electronics*, 67(9):7929–7940, 2019. 82

[100] F Caccavale, P Cilibrizzi, F Pierri, and L Villani. Actuators fault diagnosis for robot manipulators with uncertain model. *Control Engineering Practice*, 17(1):146–157, 2009. 82

[101] Qing Li and John A Bernard. Design and evaluation of an observer for nuclear reactor fault detection. *IEEE Transactions on Nuclear Science*, 49(3):1304–1308, 2002. 82

[102] Halim Alwi and Christopher Edwards. Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme. *IEEE Transactions on Control Systems Technology*, 16(3):499–510, 2008. 82

[103] A C Bittencourt and P Axelsson. Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data. *IEEE/ASME Transactions on Mechatronics*, 19(5):1694–1706, October 2014. 82, 88, 90

[104] Xuewu Dai and Zhiwei Gao. From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis. *IEEE Transactions on Industrial Informatics*, 9(4):2226–2238, 2013. 82

[105] Fan Yang, Sirish Shah, and Deyun Xiao. Signed directed graph based modeling and its validation from process knowledge and process data. *International Journal of Applied Mathematics and Computer Science*, 22(1):41–53, 2012. 83

[106] T Bossomaier, L Barnett, M Harré, and JT Lizier. *An Introduction to Transfer Entropy: Information Flow in Complex Systems*. Springer International Publishing, Cham, Switzerland, 2016. 83, 95

[107] Luiz A Baccalá and Koichi Sameshima. Partial directed coherence: a new concept in neural structure determination. *Biological cybernetics*, 84(6):463–474, 2001. 83

[108] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000. 83, 85, 86

[109] Max Lungarella, Katsuhiko Ishiguro, Yasuo Kuniyoshi, and Nobuyuki Otsu. Methods for quantifying the causal structure of bivariate time series. *International journal of bifurcation and chaos*, 17(03):903–921, 2007. 83

[110] Stig Moberg. *Modeling and control of flexible manipulators*. PhD thesis, Linköping University Electronic Press, 2010. 83, 84, 87

[111] André Carvalho Bittencourt. *Modeling and Diagnosis of Friction and Wear in Industrial Robots*. PhD thesis, Linköping University Electronic Press, 2014. 83

[112] John J Craig. *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009. 84

[113] Peter J Blau. Embedding wear models into friction models. *Tribology Letters*, 34(1): 75–79, 2009. 85

[114] Basilio Bona and Marina Indri. Friction compensation in robotics: an overview. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4360–4367. IEEE, 2005. 85

[115] Karl Johanastrom and Carlos Canudas-De-Wit. Revisiting the lugre friction model. *IEEE Control Systems Magazine*, 28(6):101–114, 2008. 85

[116] Deniz Gencaga, Kevin H Knuth, and William B Rossow. A recipe for the estimation of information flow in a dynamical system. *Entropy*, 17(1):438–470, 2015. 85, 86

[117] Sachit Butail, Violet Mwaffo, and Maurizio Porfiri. Model-free information-theoretic approach to infer leadership in pairs of zebrafish. *Physical Review E*, 93(4):042411, 2016. 86

[118] Simon Behrendt, Thomas Dimpfl, Franziska J Peter, and David J Zimmermann. Rtransferentropy—quantifying information flow between different time series using effective transfer entropy. *SoftwareX*, 10:100265, 2019. 86, 87, 91

[119] Matthäus Staniek and Klaus Lehnertz. Symbolic transfer entropy. *Physical review letters*, 100(15):158101, 2008. 87

[120] ABB. Robotstudio. URL https://new.abb.com/products/robotics/robotstudio. 87, 88

[121] ABB Robotics. Product specification irb 6700. *ABB Robotics*, 3. 88

[122] Jakob Runge, Vladimir Petoukhov, and Jürgen Kurths. Quantifying the strength and delay of climatic interactions: The ambiguities of cross correlation and a novel measure based on graphical models. *Journal of Climate*, 27(2):720–739, 2014. 93